EPFL

Thèse n° 9840

Point, Plane, Set, Match! Winning the Echo Grand SLAM

Présentée le 17 janvier 2020

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

pour l'obtention du grade de Docteur ès Sciences

par

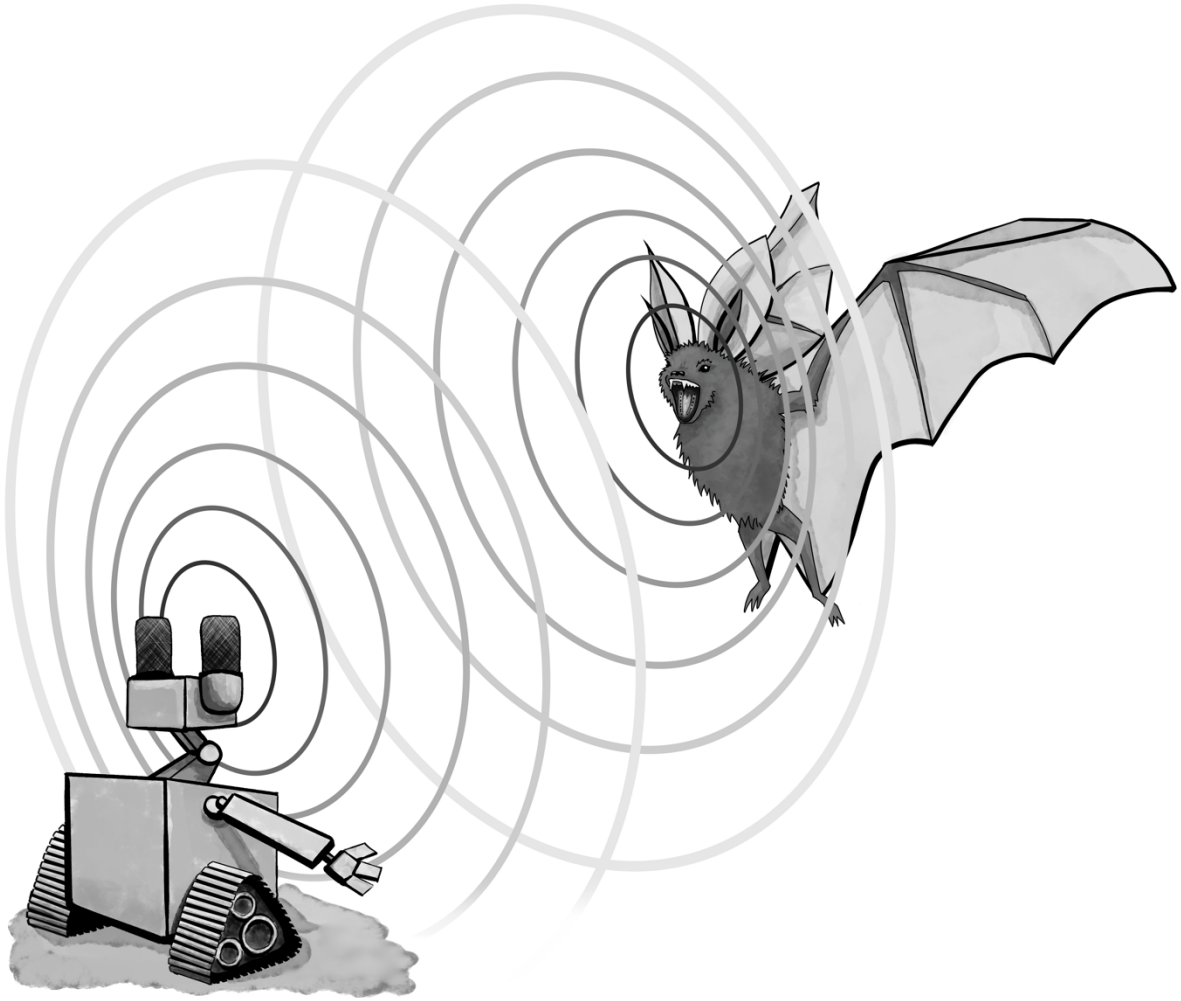
Miranda KREKOVIC

Acceptée sur proposition du jury

Dr O. Lévêque, président du jury
Prof. M. Vetterli, Dr I. Dokmanic, directeurs de thèse
Dr M. Slaney, rapporteur
Prof. K. Witrisal, rapporteur
Prof. M. Gastpar, rapporteur

École
polytechnique
fédérale
de Lausanne

2020



Credit: Michalina Pacholska

To my brother and my best friend, Gogo-bat

Acknowledgments

C'est le temps que tu as perdu pour ta rose qui fait ta rose si importante.[1]

The Little Prince
Antoine de Saint-Exupery

The journey that lead to this thesis would not have been so inspiring and enjoyable without the numerous people I have met on the way. They leveled steep roads of the path and straightened sharp turns, and I want to sincerely thank them for that.

I would like to start by expressing my deepest gratitude to my two supervisors, Prof. Martin Vetterli and Prof. Ivan Dokmanić. I was still a master student when I discovered their research on seeing rooms with echoes. To me, it sounded like magic. However, an even more magical moment happened soon after, when I was contacted by the magician itself: Ivan wrote to me about a possibility to work with them as a summer research student. Ivan, thank you reaching out and changing my life forever. Thank you for believing in me and for all the enthusiasm and courage you shared with me even before I arrived at EPFL. When I started my PhD, you helped me with my first research steps, you showed me how to ask the right questions and attack problems. During my first year when we had joint meetings with Martin, I often felt that I am in the audience of a tennis match, where instead of a tennis ball, the ideas were bouncing back and forth between Martin and you. It was inspiring, captivating, (and often hard to follow), but I was hoping that one day I will be on the field instead of in the audience. Martin, thank you for being a constant inspiration and a role model. I appreciate that you have given me all the freedom to pursue my research, but also strong directions when it was needed. Thank you for your trust and believing in my work. Thank you for always being a source of positive energy and friendly advice, and for caring far beyond mere academic achievements. One can only dream of having such a supervisor.

I would also like to thank my committee members, Prof. Michael Gastpar, Dr. Malcolm Slaney and Prof. Klaus Witrisal for taking the time to read my thesis and providing valuable feedback. I would like to specifically thank to Dr. Slaney for showing interest in my research after my first internship at Google. Seeing that my study could be practically useful and relevant was very important to me.

[1] It is the time you have wasted for your rose that makes your rose so important.

There are no words that can express my gratitude to Mihailo for his unnamed role in this journey. The cheeriest days in the lab were when you were around, sporadically passing by my office to tell jokes and talk about some random topics just to distract me from my worries. It worked each time, even when I was hearing the same joke for several times! Thank you for always offering your help and having time to listen. Thank you for proofreading parts of my thesis, brainstorming on the title, helping with practical experiments and bringing in and out desks from BC 329 countless times. I was warned that I should not work closely with you as both of us being typical east-European pessimists and self-demanding, we would shut down the best ideas even before starting them. But I am glad that I did not listen because from your side I felt nothing but support, reassurance and encouragement. Thank you for never leaving the lab without sharing a smile or a quick wave from your bike, and saying: *nemoj dugo*. Beskrajno hvala na svemu!

I also owe many thanks to Sepand. Even though you are the freshest PhD student in the lab, your advice never failed me. Thank you for your genuine care and never refusing to help. Your motivational poems brought me smiles and laughter. With your intelligence, hard work and patience, I have no doubts that you will get far. Just do not forget the R equation!

I was very lucky to be surrounded by an amazing group of friends and colleagues at LCAV. Paolo, thank you for sharing your creativity and coming up with the most awesome title for my thesis. Dalia, thank you for being a great office mate. You are one of the kindest and the most peaceful people I have ever met. I enjoyed our discussions about research very much, but it made me even happier to learn that we share a passion for dancing! Frederike, thank you for collaborations and coauthorships. Thank you for your infinite amount of understanding and everlasting patience. With you, meetings always had a calm and amiable tone. Michalina, thank you for accepting to illustrate the cover page of my thesis. Your drawing skills and imagination are impressive, as well as your mathematical expertise. Karen, thank you for always being in a cheerful mood and for caring about the well-being of PhD students. Yassine, you brought to my thesis knowledge and results that I could never have myself. It was not only your strong background in prototyping, product design and robotics that made our project successful, but it was your passion, determination and maturity. You never ceased to impress me! Thank you for believing in my theoretical results and for choosing my research to demonstrate its practical value. Thank you for constructing our robot Euclid and thanks to Euclid for not failing on us (most of the time)! Thank you Adam, Adrien, Arnaud, Eric, Golnoosh, Hanjie, Lionel, Luc, Matthieu, Niranjan and Robin for the great time in the lab, conferences, teaching sessions and hackathons. Thank you Jacqueline, Heidi and Nicoletta for your excellent organizational skills and all the help.

I was also very lucky to get to know some former LCAV students. My favorite collaboration happened with Juri and Gilles, and I would like to thank you both for taking me on board. Gilles, I believe that our teamwork, harmony and experience of pair-programming and pair-writing was unique; we were each others advisers, code reviewers, proofreaders, biggest supporters and best friends. On a less academic note, Francisco, thank you teaching me the basics of tango and inviting me to be your assistant in the tango course. Dancing was always my favorite side activity and thank you for reminding me of that! Marta, the words cannot express my gratification when I think of GirlsCoding. You fearlessly jumped on the project as soon as the first ideas were born, and I will always admire you for that. It was not always easy; in numerous occasions we had to get far out of our comfort zone, learn, improvise and start over again. Thank you for always believing in us! Not only that we reached and inspired hundreds of girls through our workshops, but we created a community of amazing volunteers who helped in organizing, advertising and mentoring the workshops. Thank you Pavle, Gilles, Mia, Iris, Sonia, Eric, Lena, Sourabh, Kristina (3x), Jelena, Nemanja, Milena, Stevan, Nikola, Ana (2x), Frederike, Evann,

Sepand, Sahand, Isinsu, Tiffany, Elena, Utku, Mahsa... Without you, this project would never have become reality!

I am also very grateful to my friends outside of LCAV who were on their own quests of attaining a PhD. I could not imagine more compassionate supporters and backers. Amer, Elena, Elie, Helena, Irene, Marguerite, Mia, Vaggelis, you were my safeguard and thank you for that! Thank you also for the ski weekends, trips, surprises, movie nights and parties. Evann, thank you for all your invaluable advice. Your innocence and kindness are rare and I feel very fortunate to have met you. Stefano, thank you for the runs and for being a French tandem partner. If I learned French with an Italian accent, now everyone knows whom to blame! Nina, Irma and Anton, our friendship dates back to high school where we sat on the same school-bench, challenged each other in mathematics, and *virili smo u daljine, puni nade*. Knowing that you were in Zurich, only a short train ride away, made me feel safer and closer to home.

During the last five years, my most beautiful memories were created with Radmila, Soumya and Hermina. You were my angels in my times of need and you enriched my life beyond measure. Hermina, thank you for accepting me under your roof and bringing randomness and joy to my life when I needed it the most. Your sense of humor, optimism and spontaneity always managed to cheer me up. Thank you for a beautiful and comforting friendship and roommateship that started more than 15 years ago; I am curious to see when our paths will cross again! Rado, the year when we were roommates was undeniably the most fantastic and the least homesick time for me in Switzerland. You are one of the bravest and the most thoughtful people I know. Thank you for many sleepless nights of dancing, chatting, laughing and crying. You gave me a lot of strength throughout my PhD and saved my life countless times! Soumya, thank you for your wonderful and honest smile. You became my older brother from day one. Thank you for all surprises, long walks and impressive stories. Even though we have been living in different countries for the last 4 years, a single day has not passed without receiving a message from you. You are incredible!

Without doubt, the best thing that happened during my PhD was meeting Gilles. Gilles, you give meaning to everything I do. I could answer the hardest problems that I pose to myself or break all personal records in running, but none of that would matter if I could not share it with you—I have learned it the hard way, but now I cannot be more certain about it! Thank you for being my best collaborator, patient listener and adviser, sport partner and the most amusing and compatible travel buddy. You make everything in my life more balanced, brighter and hearty!

Dear parents, it is impossible to put in words my appreciation for everything you have done for me and my brother. We were always your priority number one, and the older I grow, the more I am aware of that. Thank you for your infinite energy and patience, and never being tired to fulfill our craziest ideas. Thank you for all the night trains that you took to come to Zurich to see me, even when the train ride itself was twice as long as the time we spent together. You always motivated me to do my best every day and showed me that there is someone who works even harder!

Gogo, this thesis is for you. You have been my role model and my best friend since we were kids. Thank you for all the laughter and fun, for answering even my silliest questions, indulging my curiosity, challenging me with mathematical and logical questions, introducing me to coding, and, above all, always believing in me. Thank you for paving a way through many stages of my life. It was not always easy to follow it though; your imagination, talent, humility, commitment to work and results have always been beyond any measure, and you set a high bar for me. Thank you for helping me to keep jumping over it and never letting me down!

Abstract

Some of the most important and challenging problems in science and engineering are *inverse* problems. They allow us to understand phenomena that cannot be measured directly. Inverse problems might not always have a unique or stable solution, or might not have any solution at all; in these cases, they are called *ill-posed*. An example of an inverse problem in room acoustics is simultaneous localization and mapping (SLAM) from sound. It is the central theme of this thesis, and we address it from both a theoretical and practical viewpoint.

From a theoretical viewpoint, we show that our SLAM setup is ill-posed since the uniqueness condition is not satisfied. From a practical viewpoint, we propose methods to constrain and identify a solution set, and we use real experiments to confirm that such a constrained problem is stable. The acoustic SLAM consists of jointly reconstructing the geometry of a room and self-localizing. We show that it can be reformulated as the reconstruction of a set of points and planes from their pairwise distances; to solve the problem, we introduce and study *point-to-plane distance matrices* (PPDMs).

Our motivation for PPDMs comes from the need for a robust localization system in indoor environments. Location-based services are integral parts of everyday life, yet to this date no single technology can provide reliable indoor localization. Inspired by echolocation in animals, we approach the problem with the following analogy: Imagine that a bat loses the directivity in its sensing and becomes what we call an *omnidirectional bat*. It explores the room by moving randomly and listening to the echoes of its chirps that are emitted in all directions, without knowing their direction. Can it still map the room and localize itself without any prior knowledge of the room geometry and its own trajectory? Our research shows that the answer is *yes*, but not in all rooms. We emulate such a monaural and omnidirectional bat using a device equipped with a collocated speaker and microphone. At different locations inside the room, the speaker emits a pulse, and the microphone registers the room impulse response. The propagation times of the first-order echoes directly reveal the (point-to-plane) distances between the device and the walls. The problem is then to recover the locations of the device and the walls of the room from their pairwise distances; in the PPDM framework, its solution corresponds to a particular factorization of a PPDM.

Though perhaps the most popular, SLAM is not the only application of PPDMs. In fact, our abstraction can be adapted and reused to solve other problems in acoustics, computer vision and signal processing which appear quite different at first glance, but all rely on the factorization of a low-rank matrix. One famous example of such a problem is *structure from motion* (SFM), which aims at recovering a scene geometry and camera motion from images. A similar example in acoustics is called *structure from sound* (SFS), and concerns the joint localization of sensors and acoustic events. In both SFM and SFS, the idea is to factor a low-rank measurement matrix into a product of a projection and a coordinate matrix, with problem-dependent constraints on the projection matrix.

Finally, there are problems in which the projection matrix from SLAM, SFM or SFS is known,

and the goal is to recover the coordinate matrix only. To solve these problems, we introduce another low-rank matrix named *coordinate difference matrix* (CDM). Possible applications include phase retrieval of sparse signals, where the core problem can be stated as the recovery of points from their unlabeled and noisy vector differences, and optimal tournament design, where we rely on CDMs to devise an active learning algorithm from pairwise comparisons.

In a nutshell, the proposed thesis revolves around the theory, algorithms and applications of point-to-plane distance matrices and coordinate difference matrices.

Keywords: point-to-plane distance matrices, simultaneous localization and mapping, echolocation, sonar, inverse problems in Euclidean space, room geometry reconstruction, uniqueness of reconstruction, echo labeling, room impulse response, low-rank matrix factorization, measurement uncertainty, incomplete data, coordinate difference matrices, Euclidean distance matrices, vector geometry problems, structure from sound, multimodal sensor localization, ranking, phase retrieval for sparse signals.

Résumé

Certains des problèmes les plus importants et difficiles en sciences et en ingénierie sont les problèmes *inverses*. Ils nous permettent de comprendre des phénomènes qui ne peuvent être mesurés directement. Les problèmes inverses n'ont pas toujours de solution unique ou stable, ou parfois il peuvent ne pas avoir de solution du tout ; dans ces cas, ils sont appelés *mal-posés*. Un exemple de problème inverse en acoustique est *la localisation et la cartographie simultanées* (SLAM, pour *simultaneous localization and mapping*) à partir de sons. C'est le thème central de cette thèse, et nous l'abordons à la fois d'un point de vue théorique et pratique.

D'un point de vue théorique, nous montrons que la formulation SLAM est mal-posée car la condition d'unicité n'est pas remplie. D'un point de vue pratique, nous proposons des méthodes pour contraindre et identifier un ensemble de solutions, et nous utilisons des expériences réelles pour confirmer la stabilité de ces problèmes contraints. Le problème du SLAM acoustique consiste à conjointement reconstruire la géométrie d'une pièce et s'auto-localiser. Nous montrons qu'il peut être reformulé comme la reconstruction d'un ensemble de points et de plans à partir de leurs distances mutuelles ; pour résoudre le problème, nous introduisons et étudions des *matrices de distance point-à-plan* (PPDMs, pour *point-to-plane distance matrix*).

Notre motivation pour les PPDMs trouve sa source dans la nécessité d'un système de localisation robuste pour les environnements intérieurs. En effet, les services de localisation font partie intégrante de la vie quotidienne, mais à ce jour aucune technologie ne peut fournir une localisation fiable en intérieur. Inspiré par les capacités d'écholocation de certains animaux, nous abordons le problème avec l'analogie suivante : imaginons qu'une chauve-souris perde la directivité de ses sens et devient ce que nous appelons une *chauve-souris omnidirectionnelle*. Elle explore la pièce en se déplaçant au hasard et en écoutant les échos de ses sifflements qui sont émis dans toutes les directions, sans connaître la direction de leurs échos. Peut-elle toujours cartographier la pièce et se localiser sans aucune connaissance préalable de la géométrie de la pièce et de sa propre trajectoire ? Nos recherches montrent que la réponse est *oui*, mais pas dans toutes les pièces. Nous simulons une telle chauve-souris monaurale et omnidirectionnelle à l'aide d'un robot équipé d'un haut-parleur et d'un microphone co-localisés. À différents endroits de la pièce, le haut-parleur émet une impulsion et le microphone enregistre la réponse impulsionnelle de la pièce. Les temps de propagation des échos du premier ordre révèlent directement les distances (point à plan) entre l'appareil et les murs. Le problème est alors de retrouver les positions de l'appareil et les murs d'une pièce à partir de leurs distances mutuelles ; dans le cadre des PPDMs, la solution correspond à la factorisation d'une PPDM.

Bien que peut-être la plus populaire, SLAM n'est pas la seule application des PPDMs. En fait, notre abstraction peut être adaptée et réutilisée pour résoudre d'autres problèmes d'acoustique, de vision par ordinateur et de traitement du signal qui paraissent assez différents au premier abord, mais qui reposent tous sur la factorisation d'une matrice de rang faible. Un exemple connu d'un tel problème est la *structure à partir de mouvement* (SFM, pour *structure from motion*), qui vise à reconstruire la géométrie d'une scène et les mouvements d'une caméra à partir d'une

série d'images. Un exemple similaire en acoustique est appelé *structure à partir du son* (SFS, pour *structure from sound*), et concerne la localisation conjointe de capteurs et d'événements acoustiques. Dans les algorithmes SFM et SFS, l'idée est de factoriser une matrice de mesure de rang faible en un produit d'une matrice de projection et une matrice de coordonnées, avec des contraintes dépendantes du problème appliquées à la matrice de projection.

Enfin, nous pouvons envisager des problèmes pour lesquels la matrice de projection de SLAM, SFM ou SFS est connue, l'objectif étant de retrouver uniquement la matrice de coordonnées. Pour résoudre ces problèmes, nous introduisons une autre matrice de rang faible appelée *matrice de différence de coordonnées* (CDM, pour *coordinate difference matrix*). Les applications possibles des CDMs incluent l'extraction de phase de signaux parcimonieux, où le problème principal peut être reformulé comme la reconstruction de points à partir de leurs différences mutuelles, ainsi que la conception optimale de tournois sportifs, où nous utilisons les CDMs pour concevoir un algorithme d'apprentissage actif à partir de comparaisons mutuelles.

En résumé, la thèse proposée s'articule autour de la théorie, des algorithmes et des applications des matrices de distance point-à-plan et des matrices de différence de coordonnées.

Mots-Clés : matrices de distance point-à-plan, localisation et cartographie simultanées, écholocalisation, sonar, problèmes inverses dans l'espace euclidien, reconstruction de la géométrie d'une pièce, unicité de la reconstruction, marquage d'écho, réponse impulsionnelle d'une pièce, factorisation matricielle de rang faible, incertitude dans les mesures, données incomplètes, matrices de différence de coordonnées, matrices de distance euclidienne, problèmes de géométrie vectorielle, structure à partir de sons, localisation de capteurs multimodaux, classement, extraction de phase pour signaux parcimonieux.

Contents

Abbreviations and Notation

Abbreviations

Abbreviation	Description
aVGP	Assigned vector geometry problem
ACF	Autocorrelation function
AO	Alternating optimization
AOA	Angle of arrival
CDM	Coordinate difference matrix
DGP	Distance geometry problem
EDM	Euclidean distance matrix
EKF	Extended Kalman filter
ERTF	Echolocation-related transfer function
E-MDS	Edge-multidimensional scaling algorithm
FRI	Finite rate of innovation
FT	Fourier transform
GESPAR	Greedy sparse phase retrieval
GNSS	Global navigation satellite system
IPS	Indoor positioning system
MDS	Multidimensional scaling
MEMS	Microelectromechanical systems
MIMO	Multiple-input-multiple-output
MSE	Mean squared error
MUSIC	Multiple signal classification
NBA	National basketball association
NETRG	Net rating
NMR	Nuclear magnetic resonance
PPDM	Point-to-plane distance matrix
PR	Phase retrieval
RIR	Room impulse response
RMSE	Root mean squared error
RTT	Round-trip time
SFM	Structure from motion

SFS	Structure from sound
SLAM	Simultaneous localization and mapping
SNR	Signal-to-noise ratio
SVD	Singular value decomposition
TDOA	Time difference of arrival
TOA	Time of arrival
TSPR	Two-stage sparse phase retrieval
uVGP	Unassigned vector geometry problem
VGP	Vector geometry problem

List of symbols

Description	Symbol	Formula/details
discrete indexing	$[\cdot]$	or alternatively, subscript
continuous indexing	$(\cdot)$	
scalars	$a, b, c, \ldots$	
constant scalars	$A, B, C, \ldots$	
vectors	$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \ldots$	
matrices	$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \ldots$	
sets	$\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$	
Dirac delta function	$\delta(t)$	$\int_{-\infty}^{\infty} x(t)\delta(t)dt = x(0)$ (for x continuous at 0)
ℓ^2 norm	$\|\cdot\|$	$(\|x_0\|^2 + \cdots + \|x_{N-1}\|^2)^{1/2}$
ℓ^p norm	$\|\cdot\|_p$	$(\|x_0\|^p + \cdots + \|x_{N-1}\|^p)^{1/p}$
set of natural numbers	$\mathbb{N}$	$0, 1, 2, \ldots$
set of integers	$\mathbb{Z}$	$\ldots, -1, 0, 1, \ldots$
set of real numbers	$\mathbb{R}$	$(-\infty, \infty)$
set of complex numbers	$\mathbb{C}$	$a + jb$ with $a, b \in \mathbb{R}$
sign function	$\operatorname{sgn}(x)$	
diagonal of a matrix	$\operatorname{diag}(\boldsymbol{X})$	$\boldsymbol{x} = [X_{11}, X_{22}, \ldots, X_{NN}]^\top$
trace of a matrix	$\operatorname{tr}(\cdot)$	
matrix transpose	$.^\top$	
Moore-Penrose pseudoinverse	$.^\dagger$	
Hadamard product	$\circ$	
absolute value or magnitude	$\|\cdot\|$	
inner product	$\langle \cdot, \cdot \rangle$	$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^* \boldsymbol{y}$
convolution	$*$	
expected value	$\mathbb{E}[\cdot]$	
big O notation	$\mathcal{O}(\cdot)$	
Fourier transform	$\mathcal{F}\{x\}(\omega)$	$\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$
inverse Fourier transform	$\mathcal{F}^{-1}\{X\}(t)$	$\frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{-j\omega t}d\omega$

Unified notation across the chapters	Symbol
$\widetilde{\cdot}$	variable corrupted by noise
$\hat{\cdot}$	estimated variable
c_s	speed of sound
q_k	distance of a wall k from the origin
$\tan\theta_k$	slope of a wall k
D	dimension of the space
$\boldsymbol{n}_k = [\cos\varphi_k, \sin\varphi_k]^\top$	unit normal of a wall in 2D
$\boldsymbol{n}_k = [\sin\theta_k\cos\varphi_k, \sin\theta_k\sin\varphi_k, \cos\theta_k]^\top$	unit normal of a wall in 3D
$\boldsymbol{z}$	noise vector; entries are independent noise realizations
$\boldsymbol{D}$	point-to-plane distance matrix
$\boldsymbol{N}$	column-unitary matrix
$\boldsymbol{W}$	weight matrix; hollow non-negative matrix
$\boldsymbol{Z}$	noise matrix; entries are independent noise realizations
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and covariance Σ
$\mathcal{P}_k$	wall k (line in 2D, plane in 3D)
$\mathcal{R}$	room-trajectory configuration

Chapter 1

Introduction

> Un tas de pierres cesse d'être un tas de pierres dès lors qu'un seul homme le contemple avec, en lui, l'image d'une cathédrale.[1]
>
> *The Little Prince*
> Antoine de Saint-Exupery

Let us motivate the research presented in this thesis by the problem of simultaneous localization and mapping in indoor environments. We consider a specific setup which imitates the echolocation behavior of bats. Its simplicity allows us to make a connection with several seemingly different problems in signal processing. In this chapter we formally describe these problems and bring them together into a common framework with a little help from matrix algebra.

1.1 Bat-like simultaneous localization and mapping

The industrialization of the world changed the way people live and work; ever since then, the majority of people spend most of their time *indoors*. As modern day humans in the era of science and technology, we have only strengthened this trend: we expose ourselves to *new* and *unknown* indoor environments more often than ever before. As academics, we often travel to conferences that are held in large convention centers or universities. As creative and curious individuals, we frequently visit museums, concert halls and libraries. As victims of capitalism, we spend our time in shopping malls looking for a specific gadget or a piece of clothing. New environments are often accompanied by an extra level of stress and uncertainty. For instance, consider a hospital in which one needs to quickly find an emergency or a delivery room. In such circumstances, the ability to efficiently navigate in indoor environments becomes increasingly important, and a reliable and accurate indoor positioning system (IPS) enhanced by location-based services would be of great use.

The problem of outdoor positioning and navigation has largely been solved by the use of global navigation satellite systems (GNSS), considered the de facto standard for accurate outdoor navigation. However, signals from GNSS satellites are too weak to penetrate most buildings,

[1] A rock pile ceases to be a rock pile the moment a single man contemplates it, bearing within him the image of a cathedral.

making such systems ineffective for indoor localization. Many schemes have therefore been envisioned for indoor localization, but as of today, no system is in widespread use. The diversity of available technologies is large, and current systems require tailored solutions, dedicated local infrastructure and customized units [122].

One of the reasons indoor localization remains a challenge despite numerous attractive applications is that many of the proposed approaches require considerable infrastructure. Even if they take advantage of existing wireless systems to reduce the costs of equipment and installations, the solutions face the challenging problem of modeling radio propagation in indoor environments. Difficulties arise from severe multipath from walls and furniture, low probability of line-of-sight path availability, fast temporal changes, moving objects, and numerous obstacles [116]. Supported by these arguments, we anticipate that the prevalent solution to indoor localization shall neither rely on fixed infrastructure inside buildings, nor on any prior knowledge of the floor plans. We reinforce this belief by a personal experience recounted below, where having an infrastructure-free indoor localization system would have been of crucial importance.

Last winter I spent a few nights in small wooden hotel in the Swiss Alps. My room was on the first floor, while the kitchen and the dining room were in the basement. One night, a fire alarm woke me up and I quickly found my way out of the hotel. With several other guests, we stood at the entrance of the hotel at the ground level, while the smoke was coming up the stairs that connect the entrance with the kitchen. It was relieving to see that the firefighters arrived quickly, but watching them walk downstairs to the hazy basement was frightening to all of us.

This got me thinking: What if there was a localization device, which the firefighter could carry alongside a fire extinguisher? A device that could learn and display a map of the environment, continuously share its location with other firefighters, detect and localize moving people, and lead them to the exit. Clearly, such a device could not depend on the existing infrastructure which would most likely be inoperative or burned in the fire. Also, the fire-induced smoke will make vision-based solutions ineffective... One could maybe use an acoustic sonar system! The presence of small water droplets does not significantly affect the speed of sound [23], while the temperature effect can be easily taken into account in the modeling. By the time I concluded my thoughts, the fire was extinguished and we could safely return to our rooms.[2]

Motivated by the above use case, we decided to approach the problem of infrastructure-free indoor localization based on echolocation. The most well-known example is the bat: A bat emits an ultrasonic vocalization through its mouth or nose. It then listens as the signal bounces off the surrounding objects and reflects back to its ears. By comparing the times of arrival or loudness levels of echoes received at both ears, the bat accurately perceives distance and direction of objects in its vicinity. The bat can create a mental picture with amazing precision, locate, identify and capture tiny insects, and update this information with newly received echoes. Remarkably, humans have also shown the ability to echolocate, in particular when having lost their visual acuity. Some blind individuals produce palatal clicks and are able to learn about their surroundings from the echoes.

Inspired by these exemplary cases of echolocation, we used the following analogy. Imagine that a bat loses directivity in its sensing and becomes an *omnidirectional* bat. It explores the room randomly and listens to echoes of its chirps that are emitted in all directions, without knowing the direction of their echoes. Can it create the map of a room and localize itself without any a priori knowledge of the room geometry? The research presented in this thesis demonstrates that in most cases, the answer is yes, and the proposed algorithms show how it

[2]No one was hurt, except the overcooked beef and empty stomachs of a few hungry hotel guests.

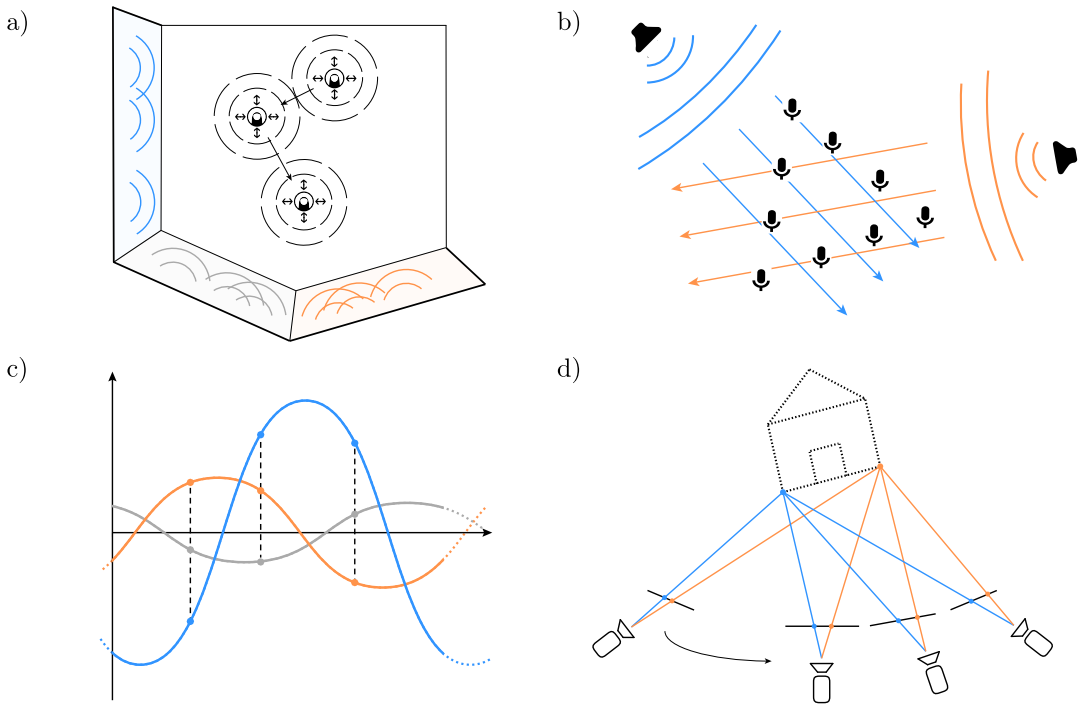


Figure 1.1: Overview of inverse problems in which measurements correspond to point-to-plane or point-to-line distances. a) Indoor simultaneous localization and mapping; b) Structure from sound; c) Sampling sinusoids at unknown locations; d) Structure from motion in 2D.

can be done. We emulate such a bat with a mobile device equipped with an acoustic sonar; it consists of a single omnidirectional source and a collocated omnidirectional receiver. The source emits acoustic pulses and the receiver registers echoes. We do not assume any prior knowledge of the room geometry, nor rely on any infrastructure inside the room. The times of flight of the first-order echoes recorded by the device correspond to point-to-line distances in 2D or point-to-plane distances in 3D. The question analogous to that of the omnidirectional monaural bat is: Can we use the point-to-plane distances to jointly localize the device and positions of the walls?

We thus consider indoor localization based on multipath with minimal equipment: one source and one receiver. Such a model has two antagonistic effects on the problem. On the one hand, it discards all information about the angles or time difference of arrivals, typically available with sensor arrays, which makes the problem harder to solve. On the other hand, it enables a mathematically simple and crisp problem statement, which appears in many other seemingly different applications, as discussed below. Hence, our analyses, uniqueness results and reconstruction algorithms find applications well beyond indoor localization in unknown indoor environments.

1.2 Unified framework for a class of problems in distance geometry

There are several practical distance geometry problems that appear very different at first glance, but can be recast into a common framework. In each, the goal is to reconstruct a set of points and planes from their pairwise distances. Perhaps surprisingly, the problems illustrated in Fig. 1.1 are (almost) equivalent: simultaneous localization and mapping with an omnidirectional sonar, structure from sound in the far field, sampling sinusoids at unknown locations, and structure from motion with orthographic projection.[3]

Measurements in these four problems range from discrete samples and images to propagation time of sound waves and echoes. In the previous section, we motivated the study of point-to-plane distances with simultaneous localization and mapping (SLAM); below, we introduce the other three equivalent problems, describe their setups, and make a connection between their original measurements and point-to-plane distances.

Structure from sound. The term *structure from sound* (SFS) was first introduced and studied by Thrun in 2006 [167]; previously, it was known as *passive localization* [21]. SFS addresses the inverse problem of simultaneously localizing a set of microphones and a set of acoustic events whose emission times are unknown. The sound sources are assumed to be in the far field, which implies that the sound propagates as a small number of plane waves and that the incident angle of each acoustic signal is the same for all microphones; this is illustrated in Fig. 1.1b. Microphones are synchronized and register absolute times of acoustic events. As the absolute emission times are unknown and different for every source, Thrun proposed to cancel them out by computing relative propagation times with respect to the microphone closest to the source. These relative propagation times are then arranged into a low-rank measurement matrix. The factorization of the measurement matrix into a product of two matrices yields microphone locations and directions of arrival of sound signals.

The solution to the SFS problem can be used for sensor array localization. Ad hoc arrays have been employed in monitoring and measurements for decades, and the importance of accurate calibration of their relative positions is evident in numerous applications. For example, in signal processing, sensor arrays are often employed to measure physical phenomena. This includes wireless sensor networks measuring weather conditions [78], ultrasonic sensors detecting breast cancer in ultrasound tomography [54], and room geometry estimation from a microphone array [49]. In acoustics, accurate source localization [118], direction-of-arrival estimation [135, 153], source separation and noise reduction [62] rely on microphone arrays with precisely calibrated microphone locations [14].

The connection between the measurement matrix and point-to-plane distances emerges from the fact that the relative times of arrival are proportional to distances between microphones and sources. With far field sources, these distances are indeed point-to-plane distances.

Sampling sinusoids at unknown locations. A set of sinusoids with the same period but different unknown amplitudes and phases are sampled at unknown locations, as illustrated in Fig. 1.1c. The inverse problem is to jointly reconstruct the sinusoids and sampling locations from a set of

[3] To be precise, structure from motion and sampling sinusoids at unknown locations are equivalent to other problems only in 2D. Moreover, sampling sinusoids is more restricted; it has one less unknown, which in terms of points and lines implies that all lines pass through the origin.

samples. This is a specific case of the more general problem of irregular sampling at unknown locations for bandlimited signals [120, 134]. Probably less obvious than for SFS, the measured samples correspond to point-to-line distances in 2D. This will become clearer in Section 1.4.1.

Structure from motion. *Structure from motion* (SFM) is a widely-used technique in computer vision. It is the inverse problem of simultaneous recovery of a scene geometry and camera motions from a sequence of images. The scenes typically consist of objects and their images are captured from different views, as illustrated in Fig. 1.1d. Every image in the sequence contains orthographic projections of the objects. An elegant solution that attracted significant attention was provided by Tomasi and Kanade in 1992 [169]. The authors proposed to arrange the stream of images into a low-rank measurement matrix. Similarly to Thrun's solution to SFS, the joint recovery of camera rotations and objects in the scene is then achieved by a factorization of the measurement matrix into a product of two matrices.

A large number of improvements to the original method have been published, making an impact in a wide range of applications, including autonomous robot navigation and augmented reality: some of them propose to handle missing entries [74, 79], and some focus on different camera matrices or dynamic scenes consisting of multiple motions [31, 149, 170, 175, 180].

We can draw a connection between the measurement matrix and point-to-plane distances in 2D only. In that case, the original objects are 2D, while *images* are represented by 2D vectors. This resembles a setup from the SFS problem. By projecting the 2D coordinates of objects onto a vector, we obtain distances between the origin of that vector and the projected points. If we consider the projection vector to be a normal vector of a line, then the measured distances indeed correspond to point-to-line distances.

Known projection matrix. We live in a 3D world where even ordinary smartphones are nowadays equipped with 3D depth cameras [89], so admittedly—SFM in 2D with 1D "cameras" is not a great achievement. On the other hand, it is easy to envision cases for which we know the camera model and attempt to recover the coordinates of unknown objects.

An analogous case in the equivalent problem of indoor SLAM is the one for which we know the room geometry and we want to localize a mobile device equipped with an omnidirectional sonar. Such a problem is of practical interest as the floor plans of apartments, hospitals, schools, and other relevant buildings are often available.

A related case in SFS occurs when the directions of arrival of sound signals are known: imagine that we are in an anechoic chamber and we want to calibrate a geometry of a microphone array using a fixed speaker. We can place the array on a turntable and rotate it multiple times to emulate a setup with one fixed microphone array and multiple speakers in the far field. As we control the rotation of the turntable, we have precise information about the incident angles of the sound waves.

The information that is assumed to be known in the three described practical situations—the camera rotations, the orientations of walls, and the sound directions—can be presented in a matrix with unit-norm columns. We call it a *projection matrix*, and refer to the corresponding columns as *projection vectors*. These examples inspired us to additionally investigate setups in which projection vectors are known, and to consider measurements different from point-to-plane distances. We realized that if we simply compute pairwise differences of point-to-plane distances associated to the same projection vector, we obtain coordinate differences between pairs of projected points. As a result, we uncovered a whole new set of inverse problems whose goal

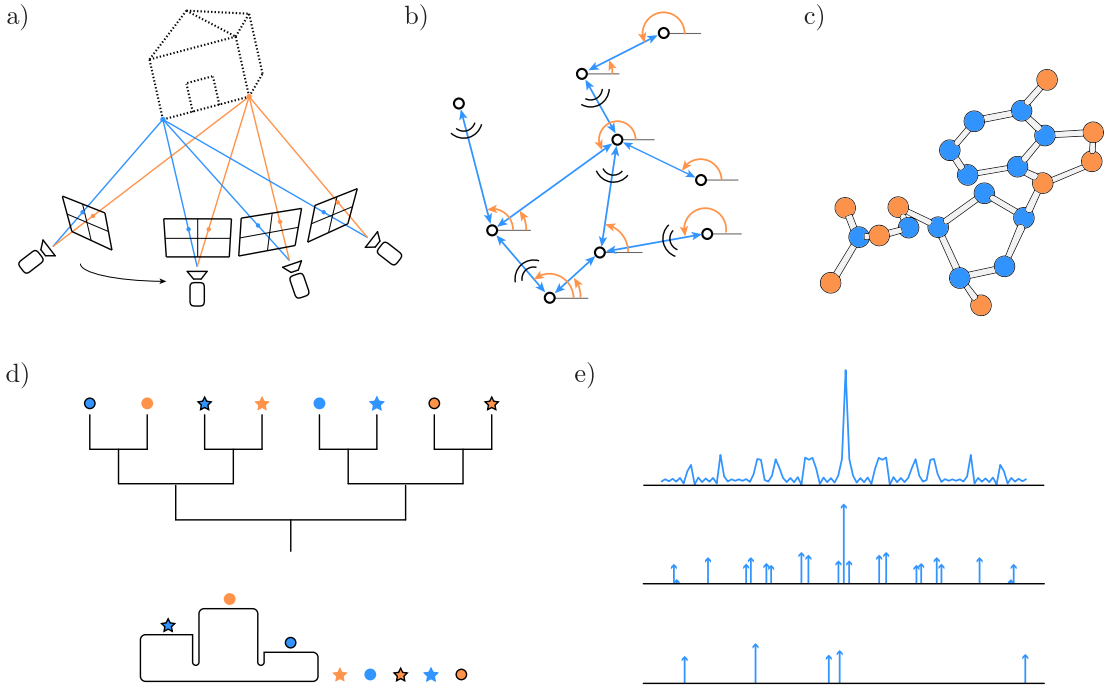


Figure 1.2: Overview of inverse problems in which measurements correspond to coordinate differences. a) Structure from motion in 3D; b) Multimodal sensor localization; c) Molecular conformation reconstruction; d) Rankings in sports; e) Phase retrieval for sparse signals.

is to recover original points from their coordinate differences, which belong to vector geometry problems (VGPs). In this thesis, we present a common framework to address these problems.

1.3 Unified framework for a class of problems in vector geometry

The term *vector geometry* was coined by Billinge [17] as an extension of distance geometry to vector measurements. Unlike distance geometry problems (DGPs) where the dissimilarity between points is given only by their pairwise distances, in VGPs the dissimilarity between points is given both by their distance and the orientation of the difference vector between them. As the distances and orientations uniquely determine vectors of the same dimension as the embedding space, the dissimilarities between the points in VGPs are nothing but their coordinate differences.

To cover a wider spectrum of applications, we extend this formulation by considering coordinate differences in higher dimension than of the embedding space. This is encountered for instance in SLAM and SFM as the number of walls and the number of images are typically much larger than three in 3D. Moreover, we study both assigned VGPs (aVGPs) and unassigned VGPs (uVGPs). In aVGPs, the indices of the pairs of points from which the measured differences originate are known, so we say that the differences are labeled, while in uVGPs one needs to jointly infer the labeling with the points.

In Fig. 1.2 we illustrate problems where measurements correspond to coordinate differences and show that the field of applications extends beyond SLAM, SFS and SFM with known projection matrices. In the following, we introduce four other equivalent problems, describe their setups, and make a connection between their original measurements and coordinate differences.

Multimodal sensor localization. We have shown that one can localize a sensor array by solving the problem of SFS. In that case, the sensor locations are determined from the measurements of several far field sources. However, as the external measurement equipment cannot always be available, a number of solutions was proposed for self-localization of nodes in wireless sensor networks. Some of them focus on the case when the nodes can only measure their pairwise distances [38, 46, 63, 132], while some methods consider only relative angles between the nodes [29, 105, 141].

Setups leveraging both measurement modalities, illustrated in Fig. 1.2b, did not attract as much attention. One example is an adaptation of a well-known multidimensional scaling (MDS) algorithm, originally devised for distances between points, to work for vectors between points [117]. We also showed that one can constructively combine distance and angle measurements into coordinate differences [91]: by recasting multimodal sensor localization as the problem of recovering points from their pairwise coordinate differences, we improve upon existing methods.

Molecular conformation reconstruction. Determining the three-dimensional structure of a protein is one of the most important and challenging problems in biology [3]. A common experimental method for finding the parameters pertaining to molecular structure is nuclear magnetic resonance (NMR) spectroscopy. NMR spectroscopy provides information about the distances between the nuclei of two bonded atoms (*covalent bond distances*), as well as the shortest distances between two non-bonded atoms (*van der Waals distances*). Moreover, from the NMR data one can extract values of the torsion angles [59]. A torsion angle is defined as the angle between planes through two sets of three atoms having two atoms in common. In addition to the distances and torsion angles, one can assume that the angles between three consecutive bonded atoms (*bond angles*) are accurately known.

Many existing methods use Euclidean distance matrices consisting of bond lengths and van der Waals distances to estimate molecular structure [27, 28, 66, 115]. Other approaches combine distance measurements with information about torsion angles and cast the problem in torsion–angle space to estimate the geometry [69, 70]. Similarly, there exist methods that rely on the torsion angles to sequentially find the coordinates of the atoms through simple geometrical angular relations, assuming a constant bond length and ignoring van der Waals distances [25, 138, 163].

We showed that one can constructively combine these different distance and angle measurements into 3D coordinate differences and cast the molecular conformation problem as point reconstruction from pairwise coordinate differences [92].

Rankings. Ranking teams based on their performance refers to arranging teams in a list, such that *better* appear higher. One of the most widely used algorithms was proposed by Massey [121]. The connection of the proposed method with coordinate differences in 1D is the following: arrange the strengths of teams as points on a line, with greater strength corresponding to the higher likelihood for a team to win a game. Then, the net scores of games can be interpreted as coordinate differences of points. Hence, by recovering points from their pairwise differences, one can reconstruct rankings from net scores.

Phase retrieval for sparse signals. Phase retrieval (PR) is the problem of recovering the phase information of a signal from the magnitude of its Fourier transform to enable the reconstruction of the original signal. PR for sparse signals is equivalent to recovering a signal from its auto-correlation function, as illustrated in Fig. 1.2e. Observe that the auto-correlation function of points corresponds to their coordinate differences. PR is more difficult than molecular conformation and ranking problems because the labels of coordinate differences are not known. In fact, the problem belongs to uVGPs, where the task of inferring the proper labeling for the differences is NP-hard [35]. Nevertheless, there exist efficient approximation algorithms. We have proposed a greedy algorithm that retrieves the labeling by iteratively selecting the point that is the most likely to generate a subset of the measured differences, so it jointly recovers the points and labels their coordinate differences [93].

1.4 The matrix tool box

Insights into the connections between these seemingly different problems would not be possible without matrix algebra. In what follows, we formally introduce the matrices that link these problems together.

1.4.1 Point-to-plane distance matrices

We define a *point-to-plane distance matrix* (PPDM),

$$\boldsymbol{D} = \mathbf{1}\boldsymbol{q}^\top - \boldsymbol{R}^\top \boldsymbol{N}, \tag{1.1}$$

as the matrix that contains pairwise distances between points and planes in 3D, or points and lines in 2D. The matrix $\boldsymbol{N} \in \mathbb{R}^{D\times K}$ is the *projection* matrix with unit-norm columns, $\boldsymbol{R} \in \mathbb{R}^{D\times N}$ is the *coordinate* matrix whose columns are the coordinates in space, and $\boldsymbol{q} \in \mathbb{R}^K$ is an unknown *offset* vector, which contains the signed distances of the planes from the origin. We use D to denote the dimensionality of the space. The matrix $\boldsymbol{N}$ is uniquely determined by K angles in 2D, and $2K$ angles in 3D.

The problems outlined in Section 1.2 can all be formulated as the following inverse problem:

Problem 1.1 (Inverse problem for PPDMs) Given $\boldsymbol{D}$, recover the coordinates $\boldsymbol{R}$, the projections $\boldsymbol{N}$ and the absolute offset $\boldsymbol{q}$, such that (1.1) holds.

A related question that naturally arises is also of importance:

Problem 1.2 (Uniqueness of PPDMs) Given $\boldsymbol{D}$, can one uniquely recover $\boldsymbol{R}$, $\boldsymbol{N}$ and $\boldsymbol{q}$, such that (1.1) holds?

As $\boldsymbol{D}$ can possibly be incomplete and noisy, we are also interested in a broader problem in which the goal is to find $\boldsymbol{R}$, $\boldsymbol{N}$ and $\boldsymbol{q}$ that minimize some error between $\mathbf{1}\boldsymbol{q}^\top - \boldsymbol{R}^\top\boldsymbol{N}$ and the

Variable	SLAM	SFS	Sampling sinusoids	SFM
D	2 or 3	2 or 3	2	2
N	number of measurement locations	number of microphones	number of sinusoids	number of features in the scene
K	number of walls	number of acoustic sources	number of sampling locations	number of images
$\boldsymbol{R}$	coordinates of measurement locations	coordinates of microphones	amplitudes a_n and phases α_n, $a_n = \sqrt{r_{n,x}^2 + r_{n,y}^2}$ and $\alpha_n = \arctan \frac{r_{n,x}}{r_{n,y}}$	coordinates of features
$\boldsymbol{N}$	unit normals of walls	directions of plane waves	sampling locations	directions of image frames
$\boldsymbol{q}$	distances of walls from the origin	absolute emission times	0	translations of images

Table 1.1: Interpretation of PPDMs in different problems.

noisy and incomplete $\boldsymbol{D}$. Moreover, measurements of point-to-plane distances can sometimes be unassigned. For instance, in indoor SLAM we record room impulse responses (RIRs) whose early reflections contain echoes and can be modeled as streams of Dirac delta pulses. By examining a single RIR, one cannot tell which pulse belongs to which echo or wall, so the measurements are unassigned. A corresponding problem can be stated as:

Problem 1.3 (Unassigned problem for PPDMs) Given permuted entries in every row of $\boldsymbol{D}$, recover $\boldsymbol{R}$, $\boldsymbol{N}$ and $\boldsymbol{q}$, such that (1.1) holds.

The interpretation of these matrices in the context of the four problems from Section 1.2 is presented in Table 1.1, while Fig. 1.3 illustrates their setups and the relevant notation.

1.4.2 Coordinate difference matrices

We define a *coordinate difference matrix* (CDM),

$$\boldsymbol{C} = \boldsymbol{x}\boldsymbol{1}^\top - \boldsymbol{1}\boldsymbol{x}^\top, \tag{1.2}$$

as the matrix that contains pairwise coordinate differences of one-dimensional points arranged in the vector $\boldsymbol{x} \in \mathbb{R}^N$. For multidimensional setups, we extend points to D dimensions and consider $K \geq D$ frame vectors $\{\boldsymbol{\varphi}_k\}_{k=1}^K$.

The problems outlined in Section 1.3 can all be formulated as one of the following two inverse problems:

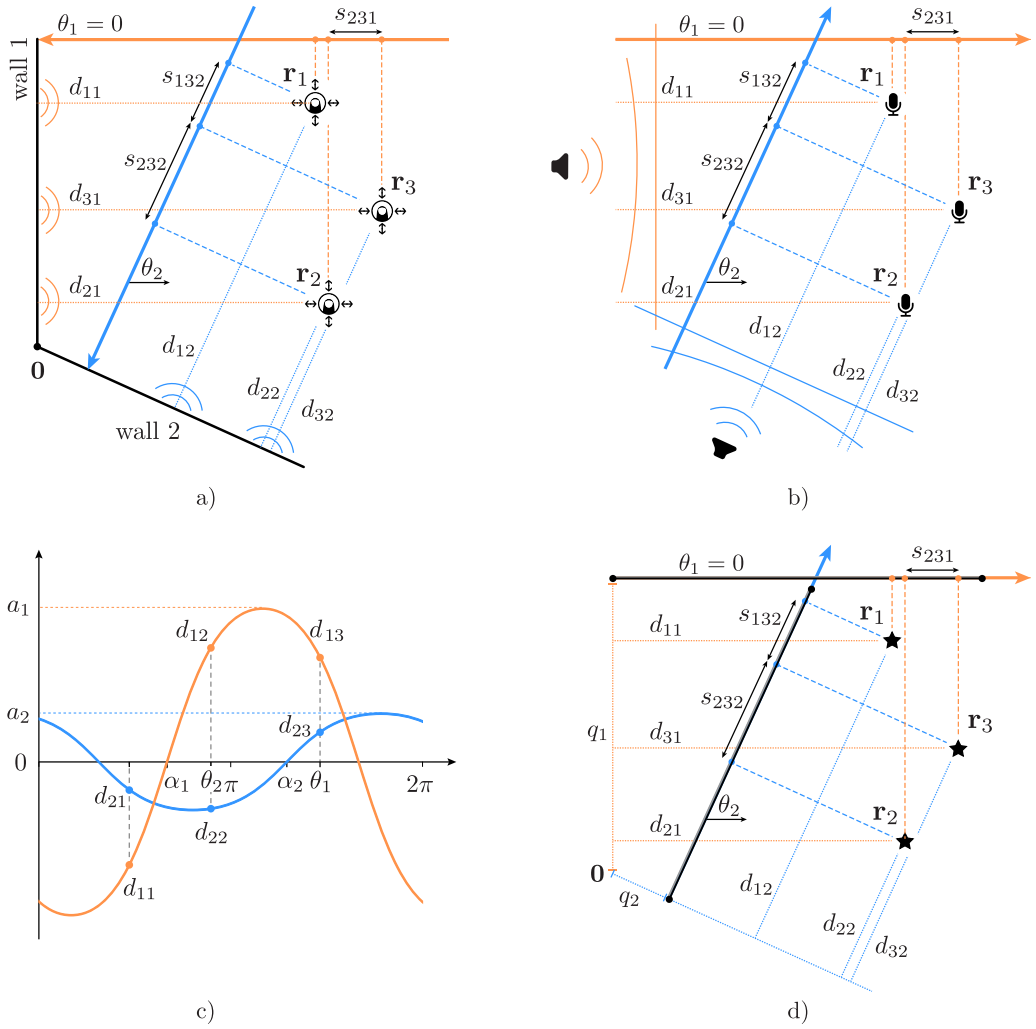


Figure 1.3: Equivalence of problems. a) Simultaneous localization and mapping: An omnidirectional speaker and microphone are mounted on a mobile device. The speaker emits a sound at unknown locations $\boldsymbol{r}_n$ and the microphone records its echoes from unknown walls k. The directions of wall normals are denoted by θ_k, while the distances of the walls from the origin in the illustrated case are 0. We measure propagation times of the first-order echoes and convert them to distances between points and planes, d_{nk}. b) Structure from sound: Microphones are placed at unknown locations $\boldsymbol{r}_n$. Speakers emit a sound at unknown times with unknown incident angles θ_k. We measure the absolute occurrence times of the sound events and convert them to relative distances between the microphones n and m projected onto incoming plane waves k, $s_{nmk} = d_{nk} - d_{mk}$. c) Sampling sinusoids: Sinusoids with the same period 2π, but different amplitudes a_n and phases α_n are sampled at unknown locations θ_k. Samples d_{nk} correspond to point-to-plane distances from a). d) Structure from motion: Different features $\boldsymbol{r}_n$ in the 2D scene are captured into images (vectors) of different orientation θ_k and translation q_k from the origin. In the local coordinate system of every image, we can measure coordinates of the projected features. These coordinates correspond to point-to-plane distances d_{nk} from a).

Problem 1.4 (Inverse problem for 1D CDMs) Given $\boldsymbol{C}$, recover the points $\boldsymbol{x}$ such that (1.2) holds.

Problem 1.5 (Multidimensional CDMs) Given D-dimensional coordinate-wise differences observed in the frame $\{\boldsymbol{\varphi}_k\}_{k=1}^K$, recover the set of points that generated them.

As was the case for PPDMs, in practice we can often measure only a noisy subset of differences from which we want to find the points that are the most likely to give rise to such measurements. Additionally, we might also need to solve an assignment problem, which in the context of CDMs is stated as:

Problem 1.6 (Unassigned CDMs) Given D-dimensional coordinate-wise differences observed in the frame $\{\boldsymbol{\varphi}_k\}_{k=1}^K$ without knowing the indices of points from which the differences originate, recover the set of points that generated them.

Even though the answer is much simpler than for PPDMs, the uniqueness question is also of practical interest:

Problem 1.7 (Uniqueness for CDMs) Given D-dimensional coordinate-wise differences observed in the frame $\{\boldsymbol{\varphi}_k\}_{k=1}^K$, can one uniquely recover the set of points that generated them?

We summarize the applications of CDMs in Table 1.2.

Variable	Sensor localization	Molecular conformation	Ranking	Phase retrieval
d	2 or 3	3	1	$\mathbb{N}$
N	number of sensors	number of atoms	number of teams	dimensionality of a signal
$\boldsymbol{x}$	coordinates of sensors	coordinates of atoms	ratings of teams	coordinates of the support points

Table 1.2: Interpretation of CDMs in different problems.

1.5 Thesis outline and main contributions

This thesis revolves around the theory, algorithms and applications of PPDMs and CDMs. The first two chapters are devoted to PPDMs: in Chapter 2 we answer the question of uniqueness, while in Chapter 3 we propose an algorithm to factor a noisy and incomplete PPDM. The practicality of PPDMs is evident in Chapter 3, where we apply the proposed algorithm to real room impulse response measurements and solve a variant of indoor SLAM. In Chapter 4 we adopt a probabilistic approach to indoor SLAM: we devise a robust algorithm which requires only one entry per row of a PPDM and describe a robot that we built to execute the algorithm in real environments. As Chapter 2 shows that some ambiguities can arise in localization from PPDMs, in Chapter 5 we find a minimal number of receivers mounted on a sonar that guarantees uniqueness; PPDMs are used as a key tool in this analysis. Finally, the last two chapters are devoted to CDMs: in Chapter 6 we study various aspects of CDMs and propose reconstruction algorithms, while in Chapter 7 we put the theory into practice and demonstrate four different applications of CDMs.

Chapter 2—Point-to-Plane Distance Matrices: Uniqueness Question. We study the problem of localizing a configuration of points and planes from the collection of their pairwise distances, which models important practical problems, such as simultaneous localization and mapping from acoustic echoes, and structure from sound. To fully understand and address localization from point-to-plane distances, we need to answer two questions: 1) is the solution unique, and 2) how to find it.

In this chapter we thoroughly study the question of uniqueness. Unlike in the case of localization from point-to-point distances, where with sufficiently many points the only possible ambiguities are those of translation, rotation, and reflection, we show that localization from PPDMs suffers from various ambiguities beyond the usual rigid body motions. These correspond to certain continuous deformations of the points–planes system. We provide a complete characterization of uniqueness, enumerate equivalence classes of configurations which lead to the same distance measurements, and algebraically characterize the related transformations in both 2D and 3D.

Summary of Contributions in Chapter 2

- We introduce a new algebraic tool: point-to-plane distance matrices.
- We derive necessary and sufficient conditions for unique reconstruction of point–plane configurations from their pairwise distances.
- We exhaustively identify the geometries of points and planes that cannot be distinguished given their PPDMs.
- In terms of SLAM, we refer to points–planes configurations as rooms and trajectories, and, therefore, our theoretical results provide a fundamental understanding and constraints under which rooms and trajectories can be uniquely reconstructed from only first-order echoes.

Chapter 3—Point-to-Plane Distance Matrices in Practice: Room Reconstruction. In this chapter, we answer the second part of the localization question, that is, *how* to estimate points and planes from their pairwise distances. Even though this brings us a step closer to having a practical solution for indoor SLAM, it is not yet enough; while PPDMs provide a good basic model for SLAM from echoes with a collocated source and receiver, the full SLAM problem presents a number of additional challenges. Therefore, in this chapter we also study the problem of associating echoes to walls, dealing with missing echoes, and distinguishing first-order from higher-order echoes.

We propose an algorithm to detect and label first-order echoes in RIRs. It is based on having a noisy estimate of a trajectory to avoid a combinatorial approach to echo labeling. We then convert the propagation times of the detected first-order echoes to distances between the measurement locations (points) and walls of a room (planes). By arranging these distances in a matrix we obtain a noisy and incomplete PPDM. We formulate the problem of reconstructing the measurement locations and walls from such a PPDM as a constrained low-rank matrix factorization with an unknown column offset. Our focus is on handling missing entries, particularly when the PPDM does not contain a single complete column. This case has not received attention in the literature and is not handled by existing algorithms, however it is prevalent in practice. We showcase the application of our algorithm in practical experiments.

Summary of Contributions in Chapter 3

- We present an end-to-end pipeline for room reconstruction and indoor localization from RIRs recorded by a mobile device at few unknown locations in an unknown room.
- In the first part, we devise an algorithm to detect the first-order echoes in RIRs and match them with the walls from which they reflect.
- In the second part, we propose a novel factorization algorithm and show its advantages over existing methods both in terms of accuracy and versatility.
- We demonstrate that the proposed approach is robust to noise using real measurements, and that it can accurately reconstruct rooms and localize devices in practice.

Chapter 4—EchoSLAM: Simultaneous Localization and Mapping with Acoustic Echoes. In this chapter, we again consider the problem of simultaneous localization and mapping from RIRs. There are two main distinctions between the algorithm proposed in this chapter and the one presented in Chapter 3. In Chapter 3, the solution is based on the assumption that we can measure the first-order echoes from all walls in a room at all measurement locations. This assumption can sometimes be violated for various reasons; for example, echoes from different walls can arrive at the same time, so their peaks in a RIR collide, or the robot can be too far from the wall, so the amplitude of the echo falls below the noise threshold. To avoid such situations and design a robust and practical solution to acoustic SLAM, in this chapter we propose to use only the collection of the first-order echoes from the *nearest* walls.

Another important distinction from the method presented in Chapter 3 concerns the estimates of trajectories. In Chapter 3, we use this information only in the first step of the algorithm; namely, to discriminate the first-order echoes from other peaks in a RIR and to label them with the correct walls. In the new approach, a noisy knowledge of measurement locations is fully

integrated in the algorithm: we rely on the robot's control to predict its next location and use it along with the measured echoes to jointly update the belief about its trajectory and the positions of the walls. This continuous joint improvement of the estimates reduces error accumulation and results in precise reconstruction, as confirmed by real experiments.

Summary of Contributions in Chapter 4

- We approach the problem from a Bayesian point of view and propose a particle-filter based algorithm to address acoustic SLAM.
- Our sensing equipment and measurement model are minimal: one omnidirectional microphone and one omnidirectional speaker mounted on a robot collect echoes with the shortest path at each location.
- We efficiently solve the echo labeling problem by building and traversing the tree of possible labels in depth-first order.
- We design and build a real three-wheeled robot equipped with a nearly collocated microphone and speaker, and use it to confirm the high accuracy of the proposed algorithm with measured RIRs.

Chapter 5—A Binaural Bat Without a Sense of Direction Can Hear the Shape of Rooms. In Chapter 2 we show that given the propagation times obtained by the collocated source and receiver, there exist practically relevant configurations of rooms and trajectories that lead to ambiguities. The question that arises naturally, and is addressed in this chapter, is to find the smallest number of receivers for which these uncertainties are resolved. To obtain a complete "if and only if" characterization, we find the constraints under which a unique solution to the SLAM problem is guaranteed, and identify the ambiguities that affect setups with fewer receivers. This has important consequences for the design of SLAM systems that do not rely on fixed beacons or odometry information.

Summary of Contributions in Chapter 5

- We show that the geometry of convex polyhedral rooms can be uniquely determined by a collection of first-order echoes of signals emitted by a source and recorded by an array of receivers mounted on a mobile robot.
- We establish conditions on the number of receivers, room geometry and robot trajectories under which uniqueness is guaranteed.
- Our theoretical findings establish a fundamental understanding of the constraints under which one can achieve a unique solution to the SLAM problem without any preinstalled beacons inside a room.

Chapter 6—Coordinate Difference Matrices: Theory. We address the inverse problem of localizing points from their pairwise coordinate differences, which is tightly related but better constrained than the localization from PPDMs. We introduce coordinate difference matrices—simple objects at the center of an efficient optimization framework for point reconstruction,

which also enable a thorough statistical analysis of the estimation errors. We first establish a reconstruction scheme in 1D and show that our algorithm is optimal in the least-squares sense. To cover a wider range of applications in which multidimensional difference vectors are available, we extend our algorithm and propose two alternative approaches to reconstruct the point configurations in higher dimensions. The first method is optimal even in the presence of noisy and partial measurements, while the main advantage of the second approach lies in its efficiency. Additionally, we also describe cases where the second method achieves an optimal solution.

Summary of Contributions in Chapter 6

- We introduce a new algebraic tool: coordinate difference matrices.
- We propose two algorithms to recover points in $\mathbb{R}^D$ from their pairwise differences (either the differences of their orthogonal coordinates or the differences of their frame coefficients).
- We provide necessary and sufficient conditions on the number and structure of difference measurements needed for a successful reconstruction.
- We present a statistical analysis of the reconstruction errors.

Chapter 7—Coordinate Difference Matrices: Applications. We demonstrate the value and versatility of CDMs in a variety of real-world problems, interesting from both theoretical and practical perspectives. More precisely, we apply the knowledge and algorithms devised in Chapter 6 to four practical problems: ranking of sports teams, sensor array calibration, source localization, and phase retrieval.

In sports, how to rank teams based on their performance has been a long-standing question with a number of proposed solutions, which differ in the modeling of the teams and the amount of information used. The evaluation and the comparison of different approaches can be troublesome in the absence of one absolute ground truth ranking. We therefore only aim to show that CDMs can be leveraged in this application using real data.

Next, we demonstrate the application of CDMs to position calibration of sensor arrays by external sources. We formulate the problem as structure from sound with known incident angles of the plane waves. We show that this additional knowledge can be used to construct an elegant and optimal CDM-based algorithm for point recovery and apply it to real data.

When external sources are not available, we consider the self-localization of nodes in a sensor network. We assume that the nodes can measure both distances and angles to a subset of other nodes in the network, and we propose to combine these measurements into CDMs. This allows us to localize sensors in closed form. We show that our point recovery algorithm outperforms state-of-the-art methods for source localization.

Lastly, we demonstrate an application of CDMs in the phase retrieval problem, which attempts to recover a signal from its auto-correlation function. Under the assumption that the original signal is sparse, we propose an algorithm that solves the problem in three stages: 1) we leverage the finite rate of innovation sampling theory to super-resolve the auto-correlation function from a limited number of samples, 2) we design a greedy algorithm that identifies the support of a sparse solution given the super-resolved auto-correlation function, 3) we recover the

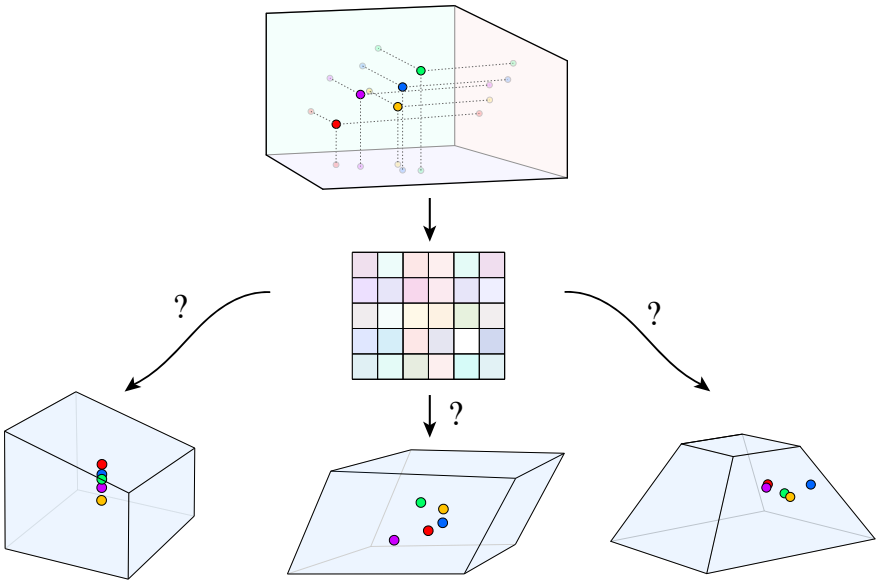
amplitudes of the signals given their locations and the measured auto-correlation function. In this chapter, we focus on the second step of the algorithm and formulate it in terms of CDMs as follows: given a complete set of unlabeled noisy entries of a CDM, arrange the entries in a matrix that is the closest to a CDM.

Summary of Contributions in Chapter 7

- We apply CDMs to rank sport teams and showcase the application of our algorithm to real NBA data.
- We apply the multidimensional generalization of CDMs to calibration of sensor arrays. Using real acoustic measurements, we demonstrate the precision and efficiency of our algorithm.
- We apply CDMs to multimodal sensor localization. Numerical simulations show that our algorithm outperforms the state-of-the-art in multimodal localization.
- We propose a three-step algorithm that resolves the phase retrieval problem on a continuous domain. We devise a greedy algorithm to solve the labeling problem in CDMs.

Chapter 2

Point-to-Plane Distance Matrices: Uniqueness Question*



A theory is the more impressive the greater the simplicity of its premises is, the more different kinds of things it relates, and the more extended its area of applicability.

Albert Einstein: Philosopher-Scientist
Albert Einstein

*The material in this chapter is the result of joint work of the author (MK) with Ivan Dokmanić (ID) and Martin Vetterli (MV). Author contributions: MK, ID, and MV designed research; MK performed research and wrote the chapter based on [95–97], which were written by herself.

2.1 Introduction

Localization methods are traditionally based on geometric information (angles, distances, or both) about known objects, often referred to as landmarks or anchors. Famous examples include global positioning by measuring distances to satellites and navigation at sea by measuring angles of celestial bodies. More recent work on simultaneous localization and mapping (SLAM) addresses the case where the positions of landmarks are also unknown.

In this chapter, we address localization from distances to (unknown) planes instead of the more extensively studied localization from distances to points. Concretely, given pairwise distances between a set of points and a set of planes, we wish to localize both the planes and the points. It is clear that a single point does not allow unique localization. As we will show, localization is in general possible with multiple points, but there are surprising exceptions.

Localization from point-to-plane distances models many practical problems. Our motivation comes from indoor localization with sound. Imagine a mobile device equipped with a single omnidirectional source and a single omnidirectional receiver that measures its distance to the surrounding reflectors, for example by emitting acoustic pulses and receiving echoes. The times of flight of the first-order echoes recorded by the device correspond to point-to-plane distances. They could be used to pinpoint its location given the positions of the walls, but the problem is harder and more interesting when we do not know where the walls are. A similar principle is used by bats to echolocate, although we do not assume having any directional information. Another problem that can be cast in this mold is the well-known "structure from sound" [167], where the task is to localize a set of microphones from phase differences induced by a set of unknown far field sources.

In this chapter, we focus on uniqueness of reconstruction from point-to-plane distance matrices (PPDMs). Unlike in the case of localization from points, where with sufficiently many points the only possible ambiguities are those of translation, rotation, and reflection [48], our analysis shows that localization from PPDMs exhibits additional ambiguities that correspond to certain continuous deformations of the points–planes system.

2.1.1 Related work

The PPDM problem is related to the more standard multidimensional unfolding [154]: localization of a set of points from distances to a set of point landmarks. There are several variations of this problem that correspond to different assumptions on what is known: 1) given distances to known landmarks, localize unknown points (i.e., estimate the unknown trajectory), 2) given distances to known points, reconstruct unknown landmarks (i.e., map the unknown environment), 3) estimate both unknown landmarks and unknown points from their pairwise distances.

The first variation is solved by simple multilateration when the association between the landmarks and the received signals is known [8]. When the association is unknown, it must be inferred jointly with the positions [111]. The second scenario is a topic of active research in signal processing and room acoustics, where it is known as "hearing the shape of a room" [5, 44, 49]. Much of that work assumes that the geometry of the microphone array is known. Then, since the source is fixed, the landmarks are modeled by points that correspond to virtual sources. In the third scenario, when neither the landmarks nor the points are known, we obtain an instance of SLAM. In general SLAM, the task is to simultaneously build some representation of the map of the environment and estimate the trajectory. Different flavors of SLAM involve different sensing modalities, but our interest is primarily in SLAM from reflections of sound or radio

waves from walls, as well as solutions based on multiple sensor modalities that provide range measurements [30, 55, 104, 125]. In this context, the "map" consists of the positions of the planar reflectors.

Prior work on indoor localization mostly focuses on computational aspects. However, more recent research in the context of 5G multipath-based positioning and mapping leverages geometry-based stochastic models of the received signal to derive theoretical position error bounds [110, 124, 156, 179]. As an example, the Cramér-Rao bound on the estimation uncertainty for the receiver position and orientation using a single transmitter in a multiple-input-multiple-output (MIMO) system has been provided [156]. Moreover, it has been shown that the state of the agent (pose, orientation and time synchronization) and the geometry of a room can be reconstructed from at least three non-line-of-sight multipath components [124] .

In the context of localization and mapping from point-to-plane distances, prior studies have also been primarily computational [90, 94, 101]. Several papers point out problems with uniqueness [26, 95, 96, 140], but a complete study was up to now absent. Peng et al. consider a setup like ours and show that the uniqueness of the mapping between the first-order echoes and the room geometry is guaranteed for all polygons except parallelograms [140]. Boutin et al. show that one can reconstruct a room from the first-order echoes from one omnidirectional speaker to four non-planar microphones, located on a drone with generic position and orientation [26].

2.1.2 Main contributions and outline

An appeal of our setup with a collocated source and receiver is that it does not require any preinstalled infrastructure. Unlike many other methods, localization from PPDMs corresponds to range-only SLAM, as the omni-directionality assumption prevents us from having any knowledge about the angles of arrival of echoes. Moreover, conventional approaches to SLAM rely on some noisy estimate of the trajectory, which is not assumed to be available in our setup. We also do not assume any motion model and allow for arbitrary configurations of waypoints.

We study uniqueness of reconstruction of points–planes configurations from their pairwise distances. We derive conditions under which the localization is unique, and provide a complete characterization of non-uniqueness by enumerating the equivalence classes of configurations that lead to same PPDMs. Since we are motivated by SLAM, we refer to points–planes configurations as *rooms* and *trajectories*. The conclusions, however, are general, and can be applied to any problem where the measurements are modeled as point-to-plane distances.

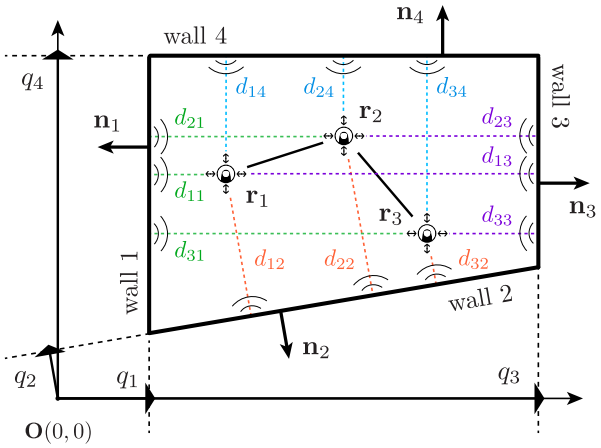
2.2 Problem setup

Suppose that a mobile device carrying an omnidirectional source and an omnidirectional receiver traverses a trajectory described by N waypoints $\{\boldsymbol{r}_n\}_{n=1}^N$. At every waypoint, the source produces a pulse, and the receiver registers the echoes. In a collocated setup the propagation times of the first-order echoes give the distances between the waypoints and walls. The distance d_{nk} between the nth waypoint and the kth wall is given by

$$d_{nk} = \frac{1}{2} c_s \tau_{nk}, \tag{2.1}$$

where c_s is the speed of sound and τ_{nk} is the propagation time of the first-order echo.

To describe a room, we consider K walls $\{\mathcal{P}_k\}_{k=1}^K$ (lines in 2D and planes in 3D) defined by their unit normals $\boldsymbol{n}_k \in \mathbb{R}^D$ and any point $\boldsymbol{p}_k \in \mathbb{R}^D$ on the wall, where $D \in \{2, 3\}$ is the



	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$
$\mathbf{r}_1$	d_{11}	d_{12}	d_{13}	d_{14}
$\mathbf{r}_2$	d_{21}	d_{22}	d_{23}	d_{24}
$\mathbf{r}_3$	d_{31}	d_{32}	d_{33}	d_{34}

$$d_{nk} = \operatorname{dist}(\mathbf{r}_n, \mathcal{P}_k) = q_k - \langle \mathbf{r}_n, \mathbf{n}_k \rangle$$

$$\mathbf{D} = [d_{nk}]_{n,k=1}^{N,K}$$

Figure 2.1: Illustration of $N = 3$ points $\{\mathbf{r}_n\}_{n=1}^{N}$ and $K = 4$ walls $\{\mathcal{P}_k\}_{k=1}^{K}$ with the corresponding PPDM.

dimension of the space. For any $\boldsymbol{x} \in \mathcal{P}_k$ we have $\langle \boldsymbol{n}_k, \boldsymbol{x} \rangle = q_k$, where $q_k = \langle \boldsymbol{n}_k, \boldsymbol{p}_k \rangle$ is the distance of the wall from the origin. We illustrate the setup in Fig. 2.1.

Given the distances between walls and waypoints,

$$d_{nk} = \operatorname{dist}(\boldsymbol{r}_n, \mathcal{P}_k) = \langle \boldsymbol{p}_k - \boldsymbol{r}_n, \boldsymbol{n}_k \rangle = q_k - \langle \boldsymbol{r}_n, \boldsymbol{n}_k \rangle, \tag{2.2}$$

for $n = 1, ..., N$ and $k = 1, ..., K$, we define

$$\boldsymbol{D} \stackrel{\text{def}}{=} [d_{nk}]_{n,k=1}^{N,K} \in \mathbb{R}^{N \times K}$$

to be the *point-to-plane distance matrix* (PPDM); we always assume $N \geq K$. By setting

$$\boldsymbol{q} \stackrel{\text{def}}{=} \begin{bmatrix} q_1 \\ \vdots \\ q_K \end{bmatrix}, \boldsymbol{R} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{r}_1 & \dots & \boldsymbol{r}_N \end{bmatrix} \text{ and } \boldsymbol{N} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{n}_1 & \dots & \boldsymbol{n}_K \end{bmatrix}, \tag{2.3}$$

we can express a PPDM as

$$\boldsymbol{D} = \mathbf{1}\boldsymbol{q}^\top - \boldsymbol{R}^\top \boldsymbol{N}, \tag{2.4}$$

where $\boldsymbol{q}$ is the vector of distances between the planes and the origin, columns of $\boldsymbol{R} \in \mathbb{R}^{D \times N}$ are the waypoint coordinates, and columns of $\boldsymbol{N} \in \mathbb{R}^{D \times K}$ outward looking normal vectors of the planes. Letting $\boldsymbol{P} \stackrel{\text{def}}{=} [\boldsymbol{p}_1, \dots, \boldsymbol{p}_K]$, the vector $\boldsymbol{q}$ can be written as $\boldsymbol{q} = \operatorname{diag}(\boldsymbol{P}^\top \boldsymbol{N})$, where $\operatorname{diag}(\boldsymbol{M})$ denotes the vector formed from the diagonal of $\boldsymbol{M}$.

PROPOSITION 2.1 With $\boldsymbol{D}$ defined as above, we have

$$\operatorname{rank}(\boldsymbol{D}) \leq D + 1.$$

PROOF. Since $\operatorname{rank}(\boldsymbol{R}^\top \boldsymbol{N}) \leq D$ and $\operatorname{rank}(\mathbf{1}\boldsymbol{q}^\top) = 1$, the statement follows by the rank inequalities. ■

This property (or its approximate version in the noisy case) is useful for 1) echo sorting, 2) completing the matrix $\boldsymbol{D}$ and estimating the unobserved distances in real situations when echoes come in and out of existence.

A pair of planes and waypoints defines a *room-trajectory* configuration

$$\mathcal{R} = \left(\{\mathcal{P}_k\}_{k=1}^K, \{\boldsymbol{r}_n\}_{n=1}^N \right),$$

and the corresponding PPDM $\boldsymbol{D}(\mathcal{R})$. In realistic convex configurations, all entries of the PPDM (2.4) are non-negative. However, in our generalized definition of a room, the waypoints can lie on either side of a wall, so we allow for signed distances.

Our central question is whether a given PPDM $\boldsymbol{D}(\mathcal{R})$ specifies a unique room-trajectory configuration $\mathcal{R}$, or, equivalently, whether the map $\mathcal{R} \mapsto \boldsymbol{D}(\mathcal{R})$ is injective. In Section 2.3 we show that rotated, translated, and reflected versions of $\mathcal{R}$ trivially all give the same $\boldsymbol{D}$, so we consider them to be the same configuration (we consider the equivalence class of all room–trajectory configurations modulo rigid motions and reflections).

We formalize the uniqueness question as follows:

PROBLEM 2.2 Are there distinct room–trajectory configurations $\mathcal{R}^1 = \left(\{\mathcal{P}_k^1\}_{k=1}^K, \{\boldsymbol{r}_n^1\}_{n=1}^N \right)$ and $\mathcal{R}^2 = \left(\{\mathcal{P}_k^2\}_{k=1}^K, \{\boldsymbol{r}_n^2\}_{n=1}^N \right)$ which are not rotated, translated, and reflected versions of each other, such that $\boldsymbol{D}(\mathcal{R}^1) = \boldsymbol{D}(\mathcal{R}^2)$?

2.3 Uniqueness of the reconstruction

Invariance to rigid motions. Let $\mathcal{R}_T$ and $\mathcal{R}_R$ denote translated and rotated/reflected versions of a setup $\mathcal{R}$, respectively. It is easy to see that if we translate, rotate or reflect walls of the room together with their waypoints, the transformed setups will have the same PPDMs $\boldsymbol{D}(\mathcal{R}) = \boldsymbol{D}(\mathcal{R}_T) = \boldsymbol{D}(\mathcal{R}_R)$: let $\boldsymbol{c} \in \mathbb{R}^D$ be a translation vector and $\boldsymbol{Q} \in \mathbb{R}^{D \times D}$ an orthogonal matrix, $\boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}$. Then,

$$\begin{aligned}
\boldsymbol{D}(\mathcal{R}_T) &= \operatorname{diag}((\boldsymbol{P} + \boldsymbol{c}\mathbf{1}^\top)^\top \boldsymbol{N})\mathbf{1}^\top - (\boldsymbol{R} + \boldsymbol{c}\mathbf{1}^\top)^\top \boldsymbol{N} \\
&= \operatorname{diag}(\mathbf{1}\boldsymbol{c}^\top \boldsymbol{N})\mathbf{1}^\top + \operatorname{diag}(\boldsymbol{P}^\top \boldsymbol{N})\mathbf{1}^\top - \boldsymbol{R}^\top \boldsymbol{N} - \mathbf{1}\boldsymbol{c}^\top \boldsymbol{N} \\
&= \operatorname{diag}(\boldsymbol{P}^\top \boldsymbol{N})\mathbf{1}^\top - \boldsymbol{R}^\top \boldsymbol{N} = \boldsymbol{D}(\mathcal{R}). \\
\boldsymbol{D}(\mathcal{R}_R) &= \operatorname{diag}((\boldsymbol{Q}\boldsymbol{P})^\top \boldsymbol{Q}\boldsymbol{N})\mathbf{1}^\top - (\boldsymbol{Q}\boldsymbol{R})^\top \boldsymbol{Q}\boldsymbol{N} \\
&= \operatorname{diag}(\boldsymbol{P}^\top \boldsymbol{N})\mathbf{1}^\top - \boldsymbol{R}^\top \boldsymbol{N} = \boldsymbol{D}(\mathcal{R}).
\end{aligned}$$

We therefore consider translated, rotated and reflected versions of $\mathcal{R}$ as the same $\mathcal{R}$. Another consequence of this invariance is that the absolute position and orientation of points and planes cannot be recovered from distances only, and the corresponding $D(D+1)/2$ degrees of freedom need to be specified separately. For instance, we can fix the translation by translating all configurations $\mathcal{R}$ by $-\boldsymbol{r}_1$; we thus assume $\boldsymbol{r}_1 = 0$ without loss of generality. As we will see later, this simplifies the analysis.

$$\overline{\boldsymbol{N}}_{2D}^{\top} = \begin{bmatrix} \cos\varphi_1^0 & \sin\varphi_1^0 & \cos\varphi_1 & \sin\varphi_1 \\ \cos\varphi_2^0 & \sin\varphi_2^0 & \cos\varphi_2 & \sin\varphi_2 \\ \vdots & \vdots & \vdots & \vdots \\ \cos\varphi_K^0 & \sin\varphi_K^0 & \cos\varphi_K & \sin\varphi_K \end{bmatrix},$$

$$\overline{\boldsymbol{N}}_{3D}^{\top} = \begin{bmatrix} \sin\theta_1^0\cos\varphi_1^0 & \sin\theta_1^0\sin\varphi_1^0 & \cos\theta_1^0 & \sin\theta_1\cos\varphi_1 & \sin\theta_1\sin\varphi_1 & \cos\theta_1 \\ \sin\theta_2^0\cos\varphi_2^0 & \sin\theta_2^0\sin\varphi_2^0 & \cos\theta_2^0 & \sin\theta_2\cos\varphi_2 & \sin\theta_2\sin\varphi_2 & \cos\theta_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sin\theta_K^0\cos\varphi_K^0 & \sin\theta_K^0\sin\varphi_K^0 & \cos\theta_K^0 & \sin\theta_K\cos\varphi_K & \sin\theta_K\sin\varphi_K & \cos\theta_K \end{bmatrix} \tag{2.5}$$

Beyond rigid motions. Perhaps surprisingly, in addition to the invariance to rigid transformations, there are many examples of rooms from Problem 2.2. The main tool in identifying the sought equivalence classes is the following lemma.

LEMMA 2.3 (NON-UNIQUENESS CRITERION) Two room–trajectory configurations $\mathcal{R}^0 = \left(\{\mathcal{P}_k^0\}_{k=1}^K, \{\boldsymbol{r}_n^0\}_{n=1}^N\right)$ and $\mathcal{R} = \left(\{\mathcal{P}_k\}_{k=1}^K, \{\boldsymbol{r}_n\}_{n=1}^N\right)$ have the same distance measurements, $\boldsymbol{D}(\mathcal{R}^0) = \boldsymbol{D}(\mathcal{R})$, if and only if

$$\overline{\boldsymbol{R}}^{\top}\overline{\boldsymbol{N}} = \boldsymbol{0}, \tag{2.6}$$

where

$$\overline{\boldsymbol{R}} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{R}^0 \\ -\boldsymbol{R} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_1^0 & \dots & \boldsymbol{r}_N^0 \\ -\boldsymbol{r}_1 & \dots & -\boldsymbol{r}_N \end{bmatrix}, \tag{2.7}$$

$$\overline{\boldsymbol{N}} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{N}^0 \\ \boldsymbol{N} \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_1^0 & \dots & \boldsymbol{n}_K^0 \\ \boldsymbol{n}_1 & \dots & \boldsymbol{n}_K \end{bmatrix}. \tag{2.8}$$

PROOF. $\mathcal{R}^0$ and $\mathcal{R}$ have the same PPDM if and only if for every $1 \le k \le K$ and $1 \le n \le N$,

$$\begin{aligned} & d_{nk}^0 = d_{nk} \\ \iff & q_k^0 - (\boldsymbol{r}_n^0)^{\top}\boldsymbol{n}_k^0 = q_k - \boldsymbol{r}_n^{\top}\boldsymbol{n}_k. \end{aligned}$$

The fact that $\boldsymbol{r}_1^0 = \boldsymbol{r}_1 = \boldsymbol{0}$ for $n = 1$ implies that $q_k^0 = q_k$ for every $1 \le k \le K$. Thus, in the matrix form we have,

$$\begin{aligned} & \boldsymbol{D}(\mathcal{R}^0) = \boldsymbol{D}(\mathcal{R}) \\ \iff & (\boldsymbol{R}^0)^{\top}\boldsymbol{N}^0 - \boldsymbol{R}^{\top}\boldsymbol{N} = 0 \\ \iff & \begin{bmatrix} \boldsymbol{R}^0 \\ -\boldsymbol{R} \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{N}^0 \\ \boldsymbol{N} \end{bmatrix} = 0. \end{aligned}$$

■

From (2.6), it follows that given a PPDM $\boldsymbol{D}(\mathcal{R}^0)$, both $\mathcal{R}^0$ and $\mathcal{R}$ are valid solutions to the problem of reconstructing rooms and trajectories from PPDMs. In other words, $\mathcal{R}^0$ and $\mathcal{R}$ belong to the same equivalence class with respect to PPDMs, which we define as

$$\left[\mathcal{R}^0\right] \stackrel{\text{def}}{=} \left\{\mathcal{R} \in \mathcal{T} \;\middle|\; \boldsymbol{D}(\mathcal{R}) = \boldsymbol{D}(\mathcal{R}^0)\right\}. \tag{2.9}$$

Here, $\mathcal{R}^0$ is a generator of the class and $\mathcal{T}$ is a collection of all room-trajectory configurations $\left(\{\mathcal{P}_k\}_{k=1}^K, \{\boldsymbol{r}_n\}_{n=1}^N\right)$, such that $\boldsymbol{n}_k \in \mathbb{R}^D$ with $\boldsymbol{n}_k^\top \boldsymbol{n}_k = 1$, $q_k \in \mathbb{R}$, and $\boldsymbol{r}_n \in \mathbb{R}^D$ for $1 \leq k \leq K$, $1 \leq n \leq N$. From (2.9) and Lemma 2.3, it further follows that the equivalence class of room-trajectory configurations can be specified as:

$$\left[\mathcal{R}^0\right] = \left\{\mathcal{R} \in \mathcal{T} \;\middle|\; \overline{\boldsymbol{R}}^\top \overline{\boldsymbol{N}} = \boldsymbol{0}\right\}, \tag{2.10}$$

where $\overline{\boldsymbol{R}}$ contains the coordinates of the waypoints of the two equivalent rooms $\mathcal{R}^0$ and $\mathcal{R}$, while the columns of $\overline{\boldsymbol{N}}$ are the wall normals of $\mathcal{R}^0$ and $\mathcal{R}$; they are given in (2.7) and (2.8), respectively.

We now characterize the equivalence classes (2.10) by analyzing $\overline{\boldsymbol{R}}^\top \overline{\boldsymbol{N}} = \boldsymbol{0}$. This relation is satisfied when the columns of $\overline{\boldsymbol{R}}$ are in the nullspace of $\overline{\boldsymbol{N}}^\top$. We parameterize the unit-norm columns of $\overline{\boldsymbol{N}}^\top = \begin{bmatrix}\boldsymbol{N}^{0\top} & \boldsymbol{N}^\top\end{bmatrix}$ as

$$\boldsymbol{n}_k^0 = \begin{bmatrix}\cos\varphi_k^0 \\ \sin\varphi_k^0\end{bmatrix} \quad \text{and} \quad \boldsymbol{n}_k = \begin{bmatrix}\cos\varphi_k \\ \sin\varphi_k\end{bmatrix} \tag{2.11}$$

in 2D, and

$$\boldsymbol{n}_k^0 = \begin{bmatrix}\sin\theta_k^0\cos\varphi_k^0 \\ \sin\theta_k^0\sin\varphi_k^0 \\ \cos\theta_k^0\end{bmatrix} \quad \text{and} \quad \boldsymbol{n}_k = \begin{bmatrix}\sin\theta_k\cos\varphi_k \\ \sin\theta_k\sin\varphi_k \\ \cos\theta_k\end{bmatrix} \tag{2.12}$$

in 3D; $\overline{\boldsymbol{N}}^\top$ is written out in (2.5). The wall normals $\boldsymbol{N}^0$ and $\boldsymbol{N}$ of the two room-trajectory configurations $\mathcal{R}^0$ and $\mathcal{R}$ are uniquely determined by the angles $\left\{\varphi_k^0\right\}_{k=1}^K$ and $\{\varphi_k\}_{k=1}^K$ in 2D, or by the pairs of angles $\left\{\theta_k^0, \varphi_k^0\right\}_{k=1}^K$ and $\{\theta_k, \varphi_k\}_{k=1}^K$ in 3D, where $\varphi_k^0, \varphi_k \in [0, 2\pi)$ and $\theta_k^0, \theta_k \in [0, \pi)$. As the converse is also true—the matrix $\overline{\boldsymbol{N}}$ uniquely determines the angles—we interchangeably use both notations.

To find the configurations that are not uniquely determined by PPDMs, we impose linear dependencies among the columns of $\overline{\boldsymbol{N}}^\top$: we select any r linearly independent columns of $\overline{\boldsymbol{N}}^\top$ and assume that the remaining columns are their linear combinations. Restricting the analysis to a particular column selection does not reduce generality, as shown in Appendix 2.A.

In addition to these linear dependencies, the columns in (2.5) are also subject to non-linear relationships due to the normalization constraint. Indeed, $\overline{\boldsymbol{N}}^\top$ has K rows, $2D$ columns, and only $2(D-1)K$ free parameters. The combination of these linear and non-linear dependencies determines the equivalence classes of the rooms and trajectories with respect to PPDMs. Our goal is to characterize these classes.

Specifically, for every equivalence class we want to characterize reference rooms $\mathcal{R}^0$ that identify the class, and a rule that generates other $\mathcal{R}$ with the same PPDM. Letting $r = \operatorname{rank}(\overline{\boldsymbol{N}})$, the analysis is performed for every $r \in \{1, \ldots, 2D-1\}$ in six steps. We introduce and explain those steps on the case $r = 2$ in 2D, rather than $r = 1$ which gives degenerate solutions (we analyze $r = 1$ subsequently).

As we will see, most of the identified cases correspond to rooms that are in some sense degenerate (for example, a "room" with all walls parallel), although as point–plane configurations they are perfectly reasonable.

The analysis in Section 2.4 and Section 2.5 together with the fact that Lemma 2.3 is necessary and sufficient proves that the union of all equivalence classes described in this chapter (see Fig. 2.2) is in fact the set of all possible configurations that are not uniquely determined by their PPDM. In other words, a room can be uniquely reconstructed from a PPDM (modulo rigid motions) if and only if it does not belong to one of the classes illustrated in Fig. 2.2.

THEOREM 2.4 In 2D, a room–trajectory configuration is not uniquely determined by its PPDM if and only if at least one of the following holds: 1) waypoints are collinear, 2) all walls are parallel (infinitely long corridors), 3) walls form a parallelogram possibly extended by parallel walls (see Fig. 2.2).

In 3D, a room–trajectory configuration is not uniquely determined by its PPDM if and only if at least one of the following holds: 1) $K < 6$, 2) waypoints are coplanar, 3) the configuration is in one of the classes summarized in Fig. 2.2.

2.4 Classification of 2D configurations

We begin by the easier 2D analysis, i.e. $D = 2$. For $\overline{\boldsymbol{N}}^\top$ to have a nullspace, we must have $r \in \{1, 2, 3\}$. For all r the analysis is performed as a sequence of six steps, which we describe in detail for $r = 2$.

2.4.1 2D rank-2: Parallelogram rooms

1. *Linear dependence:* We select r linearly independent columns of $\overline{\boldsymbol{N}}^\top$, denoted $\boldsymbol{c}_i \in \mathbb{R}^K$, $i = 1, \dots, r$, and denote the remaining columns of $\overline{\boldsymbol{N}}^\top$ by $\boldsymbol{c}_k \in \mathbb{R}^K$, $k = r+1, \dots, 2D$. We let $\boldsymbol{c}_k$ for $k > r$ be linear combinations of $\boldsymbol{c}_k$ for $k \leq r$:

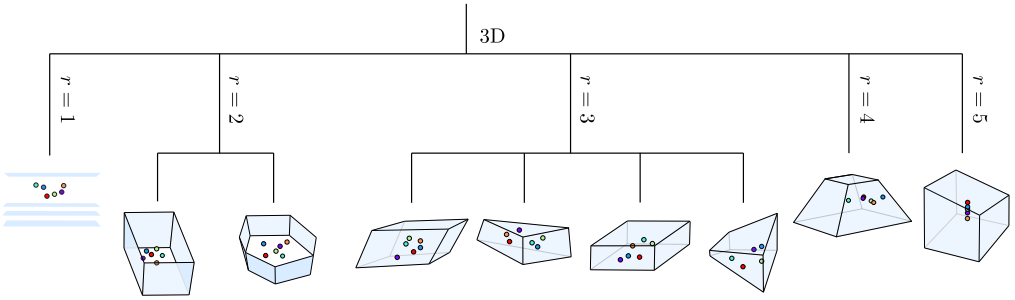
$$\begin{bmatrix} \boldsymbol{c}_{r+1} & \dots & \boldsymbol{c}_{2D} \end{bmatrix}^\top = \boldsymbol{T} \begin{bmatrix} \boldsymbol{c}_1 & \dots & \boldsymbol{c}_r \end{bmatrix}^\top, \tag{2.13}$$

for some $\boldsymbol{T} \in \mathbb{R}^{(2D-r)\times r}$.

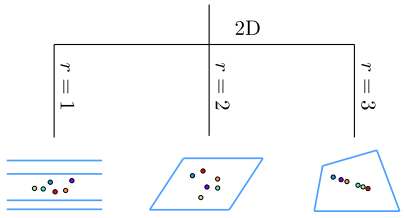
Concretely, for $r = 2$, we assume that the first two columns of $\overline{\boldsymbol{N}}^\top$ are linearly independent, while the third and the fourth column are their linear combinations. We prove in Appendix 2.A that this particular choice of columns does not incur a loss of generality in this or any of the other cases. From (2.5), for every k we have that

$$\begin{bmatrix} \cos\varphi_k \\ \sin\varphi_k \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \cos\varphi_k^0 \\ \sin\varphi_k^0 \end{bmatrix}, \text{ where } \boldsymbol{T} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}. \tag{2.14}$$

2. *Reparametrization:* When $r \leq D$, we can rearrange the columns so that the right-hand side of (2.13) contains the normals of the reference configuration $\mathcal{R}^0$, while the left-hand side has the



		(2.39)	(2.49)	(2.49), cstr: (2.50)		(2.60)	(2.60)	(2.63)	(2.67)	(2.71)
class generator	room	φ_k^0: (2.36) θ_k^0: (2.36)	φ_k^0: (2.46) θ_k^0: (2.43)	φ_k^0: any θ_k^0: (2.43) cstr: (2.50)	φ_k^0: any θ_k^0: (2.55)	φ_k^0: any θ_k^0: any $K_0 < 6$	φ_k^0: any θ_k^0: any $K_0 < 6$	$\varphi_{1,2}^0$: any θ_k^0: any	φ_k^0: any θ_k^0: any	φ_k^0: any θ_k^0: any
		all walls parallel	2 pair of parallel walls	all walls parallel to a line		any with less than 6 walls		two sets of parallel walls		
	trajectory	any	any	any	any	any	any	any	planar	linear
transformation	room	none	φ_k: (2.47) θ_k: (2.47)	none	φ_k: (2.58) θ_k: (2.58)	φ_k: (2.58) θ_k: (2.58)	φ_k: (2.58) θ_k: (2.58)	φ_k: (2.58) θ_k: (2.58)	φ_k: (2.65) θ_k: (2.65)	φ_k: any θ_k: (2.69)
	trajectory	(2.38) move points in a plane	(2.48)	(2.49), cstr: (2.50) move points on a line parallel to the walls	(2.59)	(2.59)	(2.59)	(2.59)	(2.66)	(2.70) scale



		(2.29)	(2.22)	(2.34)
class generator	room	φ_k^0: (2.26) all walls parallel	φ_k^0: (2.18) parallelograms	φ_k^0: any
	trajectory	any	any	linear
transformation	room	none	φ_k: (2.19)	φ_k: (2.31)
	trajectory	(2.28) move points on a line	(2.21) scale and shear	(2.33) scale

Figure 2.2: An overview of the equivalence classes of the room-trajectory configurations with respect to PPDMs in 2D and 3D. The first row references the class-defining equation; determines the class. The middle part characterizes the class generator (wall orientations in the reference room in the gray box; information about the corresponding trajectories in the white box). The bottom part describes transformations of the reference to equivalent walls and waypoints and references corresponding equations; "cstr" indicates additional constraints on $\boldsymbol{T}$. Where appropriate, we provide a short description of the transformation.

normals of the putative equivalent configuration $\mathcal{R}$. In particular, we obtain

$$\boldsymbol{N}^\top = \boldsymbol{T}'\boldsymbol{N}^{0\top}, \tag{2.15}$$

where $\boldsymbol{T}' \in \mathbb{R}^{D\times D}$. $\boldsymbol{T}'$ can be decomposed as a product $\boldsymbol{T}' = \boldsymbol{Q}\boldsymbol{U}$ of an orthogonal matrix $\boldsymbol{Q}$ and an upper triangular matrix $\boldsymbol{U}$. $\boldsymbol{Q}$ acts as a rotation, so without loss of generality we set $\boldsymbol{Q} = \boldsymbol{I}$ and $\boldsymbol{T}' = \boldsymbol{U}$. That is, we assume that the entries of $\boldsymbol{T}'$ below the diagonal are 0, which removes the $\binom{D}{2}$ rotational degrees of freedom. Since (2.15) contains a subset of equations from (2.13), we propagate this change back to (2.13) by modifying the corresponding elements of $\boldsymbol{T}$.

When $r = D$, the original system of equations (2.14) already has a form of (2.15). Therefore, we only need to set $c = 0$ and obtain an upper triangular matrix,

$$\boldsymbol{T} = \begin{bmatrix} a & b \\ 0 & d \end{bmatrix}.$$

3. *Reference room:* To find a reference room, we select an arbitrary $\boldsymbol{T}$ (respecting the zero entries from step 2), and solve for the normals satisfying (2.13). From (2.14), we observe that

$$(a\cos\varphi_k^0 + b\sin\varphi_k^0)^2 + (d\sin\varphi_k^0)^2 = \cos\varphi_k^2 + \sin\varphi_k^2 = 1, \tag{2.16}$$

so the angles of the reference room cannot be chosen arbitrarily. To find the values of $\{\varphi_k^0\}_{k=1}^K$ with respect to free parameters a, b and d, we solve (2.16) and obtain

$$A\cos^2(2\varphi_k^0) + B\cos(2\varphi_k^0) + C = 0, \tag{2.17}$$

where

$$\begin{aligned} A &= (a^2 - b^2 + d^2)^2 + 4a^2b^2, \\ B &= 2(a^2 - b^2 + d^2)(a^2 + b^2 + d^2 - 2), \\ C &= (a^2 + b^2 + d^2 - 2)^2 - 4a^2b^2. \end{aligned}$$

Let first $A = 0$. Then (2.17) has two solutions: $a = 0$, $b^2 = -d^2$ and $b = 0$, $a^2 = d^2$. The first one implies that $b = d = 0$, which makes (2.14) inconsistent. The second one leads to $\boldsymbol{T}$ being a reflection matrix:

$$\boldsymbol{T} = \begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix} \quad \text{or} \quad \boldsymbol{T} = \begin{bmatrix} \pm 1 & 0 \\ 0 & \mp 1 \end{bmatrix},$$

which is quotioned out in Problem 2.2.

For $A \neq 0$, we have

$$\cos(2\varphi_k^0) = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}. \tag{2.18}$$

There are eight solutions for φ_k^0, four of which satisfy (2.14). The valid solutions always come as pairs $(\varphi_1^0, \varphi_2^0) = (\varphi_1^0, \varphi_1^0 + \pi)$ and $(\varphi_3^0, \varphi_4^0) = (\varphi_3^0, \varphi_3^0 + \pi)$.

4. *Equivalent rooms:* From (2.13), we identify the transformation that takes the normals of the reference room to the normals of an equivalent room. The corresponding angles in the equivalent room are computed from (2.14),

$$\varphi_k = f(\varphi_k^0, s_k, a, b, d) = \arctan\frac{d\sin\varphi_k^0}{a\cos\varphi_k^0 + b\sin\varphi_k^0} + s_k\pi, \tag{2.19}$$

where $s_k \in \{0, 1\}$.

5. *Corresponding trajectories:* Next, we find the waypoints $\{\boldsymbol{r}_n^0\}_{n=1}^N$ and $\{\boldsymbol{r}_n\}_{n=1}^N$ that lie in the nullspace of $\overline{\boldsymbol{N}}^\top$. The nullspace is spanned by:

$$\boldsymbol{v}_1 = \begin{bmatrix} -a, & -b, & 1, & 0 \end{bmatrix}^\top, \quad \boldsymbol{v}_2 = \begin{bmatrix} 0, & -d, & 0, & 1 \end{bmatrix}^\top,$$

so the columns of $\overline{\boldsymbol{R}}$ are of the form

$$\begin{bmatrix} \boldsymbol{r}_n^0 \\ -\boldsymbol{r}_n \end{bmatrix} = \boldsymbol{v}_1\gamma_1 + \boldsymbol{v}_2\gamma_2, \tag{2.20}$$

where $\gamma_1, \gamma_2 \in \mathbb{R}$. The waypoints in the reference room are chosen without restrictions, while the waypoints in the equivalent room are obtained by applying a non-rigid transformation

$$\boldsymbol{r}_n = (\boldsymbol{T}^\top)^{-1}\boldsymbol{r}_n^0. \tag{2.21}$$

This transformation corresponds to shearing followed by scaling. To show that, without loss of generality we can consider a reference room to be rectangular, so that $\varphi_1^0 = \pi/2$ and $\varphi_2^0 = \pi$. Then, (2.17) simplifies to $A - B + C = 0$ and $A + B + C = 0$ for $k = 1$ and $k = 2$, respectively, which further implies that $a^2 + d^2 = 1$ and $b^2 = 1$. The matrix $(\boldsymbol{T}^\top)^{-1}$ can be factorized into a product of a scaling matrix and a shear matrix as:

$$(\boldsymbol{T}^\top)^{-1} = \begin{bmatrix} \frac{1}{a} & 0 \\ 0 & \frac{1}{d} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \pm\frac{1}{a} & 1 \end{bmatrix},$$

with $a^2 + d^2 = 1$.

6. *Equivalence class:* The solutions of (2.18) suggest that we can construct a reference room by arbitrarily choosing two wall normals, φ_1^0 and φ_3^0, and solving the system of two equations (2.17) with $k \in \{1, 3\}$. This fixes two parameters (e.g., a and b) in $\boldsymbol{T}$ and leaves the third (e.g., d) free to generate an infinite number of rooms equivalent to the reference room. Reference rooms are not restricted to only two walls; we can have any number of additional walls parallel to those determined by φ_1^0 and φ_3^0, since they also satisfy (2.18).

A room-trajectory configuration $\mathcal{R}^0$ with walls $\{\mathcal{P}_k^0\}_{k=1}^K = \{(\boldsymbol{n}_k^0, q_k^0)\}_{k=1}^K$ and waypoints $\{\boldsymbol{r}_n^0\}_{n=1}^N$ is a generator of a class of room-trajectory configurations with identical PPDMs. The wall normals $\boldsymbol{n}_k^0$ are chosen as described above and the waypoints are arbitrary, $\boldsymbol{r}_n^0 \in \mathbb{R}^D$ for $1 \le n \le N$. The above analysis defines the following equivalence class of room-trajectory configurations with the same PPDMs:

$$\begin{aligned}\left[\mathcal{R}^0\right] = \Big\{\mathcal{R} \,\big|\, \varphi_k = f(\varphi_k^0, s_k, a, b, c), a, b \in \mathbb{R} \text{ s.t. (2.17) holds}, d \in \mathbb{R}, s_k \in \{0, 1\},\\ q_k = q_k^0 \text{ for } 1 \le k \le K, \boldsymbol{r}_n \text{ s.t. (2.20) holds for } 1 \le n \le N\Big\}.\end{aligned} \tag{2.22}$$

There are no constraints on the distances of walls from the origin in the reference room and we can set $\boldsymbol{q}^0$ arbitrarily. We note that this class includes parallelogram rooms for $K = 4$, $\varphi_2^0 = \varphi_1^0 + \pi$ and $\varphi_4^0 = \varphi_3^0 + \pi$.

An example of three parallelogram configurations with the same PPDM is illustrated in Fig. 2.3.

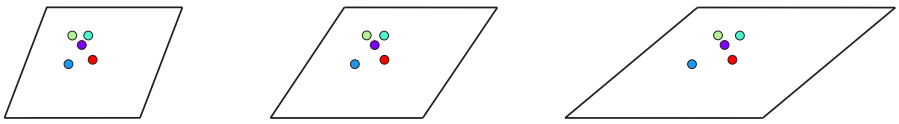


Figure 2.3: Parallelogram rooms with the same PPDM.

2.4.2 2D rank-1: Infinitely long corridors

1. *Linear dependence:* In 2D, setting $\operatorname{rank}(\overline{\boldsymbol{N}}) = 1$ leads to degenerate rooms. To show that, assume that every column of $\overline{\boldsymbol{N}}^\top$ is a scaled version of the first column,

$$\begin{bmatrix} \sin\varphi_k^0 \\ \cos\varphi_k \\ \sin\varphi_k \end{bmatrix} = \boldsymbol{T}\cos\varphi_k^0, \text{ where } \boldsymbol{T} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \tag{2.23}$$

2. *Reparametrization:* These dependencies can be partially expressed as a transformation of the normals of the reference room to those of the equivalent room. From (2.23) we have:

$$\begin{bmatrix} \cos\varphi_k \\ \sin\varphi_k \end{bmatrix} = \boldsymbol{T}' \begin{bmatrix} \cos\varphi_k^0 \\ \sin\varphi_k^0 \end{bmatrix}, \text{ where } \boldsymbol{T}' = \begin{bmatrix} b & 0 \\ c & 0 \end{bmatrix}.$$

With $c = 0$, $\boldsymbol{T}'$ becomes upper triangular. This eliminates rotations. Propagating back to $\boldsymbol{T}$, we get:

$$\boldsymbol{T} = \begin{bmatrix} a, & b, & 0 \end{bmatrix}^\top. \tag{2.24}$$

3. *Reference room:* We see that (2.23) constrains the normals of the reference room, since

$$\tan\varphi_k^0 = a \tag{2.25}$$

must hold for every k. That is, the wall normals of the reference room cannot be chosen arbitrarily. Letting $s_k \in \{0, 1\}$, we summarize both solutions to (2.25) as

$$\varphi_k^0 = f(s_k, a) = \arctan a + s_k\pi. \tag{2.26}$$

For $K \geq 2$ walls, (2.26) implies that every φ_k^0 can only assume two values. These correspond to parallel walls since $\varphi_i^0 = \varphi_k^0 + \pi$ for $s_i = 0$ and $s_k = 1$.

4. *Equivalent rooms:* From (2.23) and (2.24) we have $\varphi_k \in \{0, \pi\}$ and

$$a^2 + 1 = b^2. \tag{2.27}$$

5. *Corresponding trajectories:* Though all rooms in this class have the same geometry, there are infinitely many trajectories that lead to the same PPDM. To see this, imagine an infinite corridor

with two parallel walls. The points on any line parallel to the walls cannot be discriminated from distances to walls. Formally, a basis for the nullspace of $\overline{\boldsymbol{N}}^\top$ is

$$\boldsymbol{v}_1 = \begin{bmatrix} -a & 1 & 0 & 0 \end{bmatrix}^\top, \boldsymbol{v}_2 = \begin{bmatrix} -b & 0 & 1 & 0 \end{bmatrix}^\top, \boldsymbol{v}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top,$$

so the columns of $\overline{\boldsymbol{R}}$ have to be of the form

$$\begin{bmatrix} \boldsymbol{r}_n^0 \\ -\boldsymbol{r}_n \end{bmatrix} = \gamma_1 \boldsymbol{v}_1 + \gamma_2 \boldsymbol{v}_2 + \gamma_3 \boldsymbol{v}_3, \tag{2.28}$$

where γ_1, γ_2 and $\gamma_3 \in \mathbb{R}$. This further implies that the waypoints of the reference room $\{\boldsymbol{r}_n^0\}_{n=1}^N$ and the y coordinates of $\{\boldsymbol{r}_n\}_{n=1}^N$ in the equivalent rooms are independent and they can be chosen arbitrarily. The x coordinates of $\{\boldsymbol{r}_n\}_{n=1}^N$ are given by (2.28).

6. *Equivalence class:* This trivial case results in the equivalence class of room-trajectory configurations, in which the rooms have parallel walls. They are generated by a reference room $\{\mathcal{P}_k^0\}_{k=1}^K$ with the wall normals from (2.26), $\boldsymbol{q}^0 \in \mathbb{R}^K$, and arbitrary waypoints $\boldsymbol{r}_n^0 \in \mathbb{R}^D$ for $1 \leq n \leq N$,

$$\begin{aligned} \left[\mathcal{R}^0\right] = \Big\{ \mathcal{R} \,\big|\, & \varphi_k \in \{0, \pi\}, a \in \mathbb{R}, b \in \mathbb{R} \text{ s.t. (2.27) holds}, s_k \in \{0, 1\}, \\ & q_k = q_k^0, \text{for } 1 \leq k \leq K, \boldsymbol{r}_n \text{ s.t. (2.28) holds for } 1 \leq n \leq N \Big\}. \end{aligned} \tag{2.29}$$

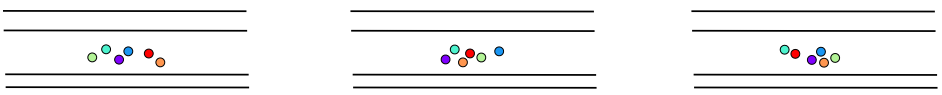


Figure 2.4: Example of three equivalent infinitely long corridors.

2.4.3 2D rank-3: Linear trajectories

1. *Linear dependence:* We assume $\text{rank}(\overline{\boldsymbol{N}}) = 3$ so that

$$\cos\varphi_k = \boldsymbol{T} \begin{bmatrix} \sin\varphi_k \\ \cos\varphi_k^0 \\ \sin\varphi_k^0 \end{bmatrix}, \text{ where } \boldsymbol{T} = \begin{bmatrix} a & b & c \end{bmatrix}. \tag{2.30}$$

2. *Reparametrization:* As $r > D$, we cannot rewrite (2.30) such that the wall normals of $\mathcal{R}$ and $\mathcal{R}^0$ are on the opposite sides of the equation, so we omit this step.

3. *Reference room:* From (2.30), we observe that the wall orientations of the reference room are unconstrained.

4. *Equivalent rooms:* We can express the wall orientations φ_k in the equivalent room as a function of φ_k^0 and entries in $\boldsymbol{T}$,

$$\varphi_k = f(\varphi_k^0, s_k, a, b, c) = s_k \arccos \frac{b\cos\varphi_k^0 + c\sin\varphi_k^0}{\sqrt{a^2+1}} - \arctan a, \tag{2.31}$$

where $s_k \in \{-1, 1\}$.

5. *Corresponding trajectories:* The nullspace of $\overline{\boldsymbol{N}}^\top$ is spanned by $\boldsymbol{v}_1 = \begin{bmatrix} -b & -c & 1 & -a \end{bmatrix}^\top$, so the columns of $\overline{\boldsymbol{R}}$ satisfy

$$\begin{bmatrix} \boldsymbol{r}_n^0 \\ -\boldsymbol{r}_n \end{bmatrix} = \boldsymbol{v}_1 \gamma, \tag{2.32}$$

where $\gamma \in \mathbb{R}$. This can be further rewritten as

$$\boldsymbol{r}_n = \boldsymbol{S}\boldsymbol{r}_n^0 \quad \text{and} \quad r_{n,y}^0 = \frac{c}{b} r_{n,x}^0, \tag{2.33}$$

where $\boldsymbol{S}$ is a scaling matrix with $\frac{1}{b}$ and $-\frac{a}{c}$ on a diagonal, and $\boldsymbol{r}_n^0 = [r_{n,x}^0, r_{n,y}^0]^\top$. This suggests that the x and y coordinates of the waypoints in both rooms are dependent, and the trajectories are linear.

6. *Equivalence class:* A room-trajectory configuration $\mathcal{R}^0$, with an arbitrary room $\{\mathcal{P}_k^0\}_{k=1}^K$ and a linear trajectory $\{\boldsymbol{r}_n^0\}_{n=1}^N$ satisfying (2.33), generates the following equivalence class:

$$\begin{aligned} \left[\mathcal{R}^0\right] = \Big\{ \mathcal{R} \,\big|\, & \varphi_k = f(\varphi_k^0, s_k, a, b, c), a, b, c \in \mathbb{R}, s_k \in \{-1, 1\}, q_k = q_k^0, \\ & \text{for } 1 \le k \le K, \boldsymbol{r}_n \text{ s.t. (2.32) holds for } 1 \le n \le N \Big\}. \end{aligned} \tag{2.34}$$

For any arbitrary room with K walls and a PPDM measured at collinear waypoints, we can find another room with the same PPDM obtained at different collinear waypoints; an example is shown in Fig. 2.5.

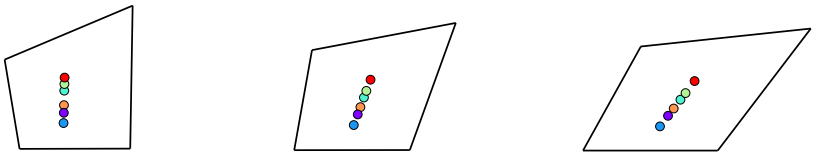


Figure 2.5: Example of equivalent rooms with linear trajectories.

2.5 Classification of 3D configurations

In 3D, we analyze the cases $r \in \{1, 2, 3, 4, 5\}$.

2.5.1 3D rank-1: Infinitely long and tall corridors

1. *Linear dependence:* When $\operatorname{rank}(\overline{\boldsymbol{N}}) = 1$ in 3D, five columns of $\overline{\boldsymbol{N}}^\top$ are scaled version of a single non-zero column,

$$\begin{bmatrix} \sin\theta_k^0 \sin\varphi_k^0 \\ \cos\theta_k^0 \\ \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \\ \cos\theta_k \end{bmatrix} = \boldsymbol{T} \sin\theta_k^0 \cos\varphi_k^0, \quad \text{where } \boldsymbol{T} = \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix}. \tag{2.35}$$

2. *Reparametrization:* The requirement (2.35) implies the following relationship between the wall normals of the reference room and those of the equivalent room:

$$\begin{bmatrix} \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \\ \cos\theta_k \end{bmatrix} = \begin{bmatrix} c & 0 & 0 \\ d & 0 & 0 \\ e & 0 & 0 \end{bmatrix} \begin{bmatrix} \sin\theta_k^0 \cos\varphi_k^0 \\ \sin\theta_k^0 \sin\varphi_k^0 \\ \cos\theta_k^0 \end{bmatrix},$$

As before, we set $d = e = 0$ to get an upper triangular matrix.

3. *Reference room:* From (2.35), it follows that

$$\tan\varphi_k^0 = a \quad \text{and} \quad \tan\theta_k^0 = \frac{1}{b\cos\varphi_k^0}$$

for every k. Then,

$$\begin{aligned} \varphi_k^0 &= \arctan a + s_k\pi, \\ \theta_k^0 &= \begin{cases} \arctan\frac{\sqrt{a^2+1}}{b} + t_k\pi & s_k = 0, \\ -\arctan\frac{\sqrt{a^2+1}}{b} + t_k\pi & s_k = 1, \end{cases} \end{aligned} \tag{2.36}$$

where s_k, $t_k \in \{0, 1\}$ are independent binary variables. That is, the reference room cannot be chosen arbitrarily; the angles can only assume two values that yield parallel walls.

4. *Equivalent rooms:* From (2.35) we also find that $\sin\varphi_k = 0$ and $\cos\theta_k = 0$, so the angle θ_k takes a value of $\pi/2$, while φ_k is either 0 or π, depending on the value of t_k. This dependence arises from the fact that

$$c = \begin{cases} b\cos\varphi_k\sqrt{\frac{a^2+b^2+1}{b^2}} & t_k = 0, \\ -b\cos\varphi_k\sqrt{\frac{a^2+b^2+1}{b^2}} & t_k = 1, \end{cases} \tag{2.37}$$

needs to be satisfied for every k for (2.35) to be consistent.

5. *Corresponding trajectories:* Analogously to the rank-1 case in 2D, the ambiguity in the reconstruction is due to the multitude of consistent trajectories. Points in planes parallel to the walls cannot be uniquely determined from distances to the walls. The nullspace of $\overline{\boldsymbol{N}}^\top$ is spanned by five vectors,

$$\begin{aligned} &\boldsymbol{v}_1 = \begin{bmatrix} -a & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^\top, \quad \boldsymbol{v}_2 = \begin{bmatrix} -b & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^\top, \quad \boldsymbol{v}_3 = \begin{bmatrix} -c & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^\top, \\ &\boldsymbol{v}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^\top, \quad \boldsymbol{v}_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top, \end{aligned}$$

so the columns of $\overline{\boldsymbol{R}}$ are

$$\begin{bmatrix} \boldsymbol{r}_n^0 \\ -\boldsymbol{r}_n \end{bmatrix} = \gamma_1\boldsymbol{v}_1 + \gamma_2\boldsymbol{v}_2 + \gamma_3\boldsymbol{v}_3 + \gamma_4\boldsymbol{v}_4 + \gamma_5\boldsymbol{v}_5, \tag{2.38}$$

where $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ and $\gamma_5 \in \mathbb{R}$. This implies that the waypoints $\{\boldsymbol{r}_n^0\}_{n=1}^N$ in the reference room and the y and z coordinates of $\{\boldsymbol{r}_n\}_{n=1}^N$ in the equivalent rooms are independent and can be chosen arbitrarily, whereas the x coordinates of $\{\boldsymbol{r}_n\}_{n=1}^N$ are given by (2.38).

6. *Equivalence class:* An equivalence class of these degenerate room-trajectory configurations with parallel walls is generated by a reference room $\{\mathcal{P}_k^0\}_{k=1}^K$ with the wall normals from (2.36), $\boldsymbol{q}^0 \in \mathbb{R}^K$, and arbitrary waypoints $\{\boldsymbol{r}_n^0\}_{n=1}^N$,

$$\left[\mathcal{R}^0\right] = \Big\{\mathcal{R} \,\big|\, \varphi_k \in \{0,\pi\}, \theta_k = \pi/2, a, b \in \mathbb{R}, c \in \mathbb{R} \text{ s.t. (2.37) holds}, s_k \in \{0,1\}, t_k \in \{0,1\},$$
$$q_k = q_k^0, \text{ for } 1 \le k \le K, \boldsymbol{r}_n \text{ s.t. (2.38) holds for } 1 \le n \le N\Big\} \tag{2.39}$$

An example of such room-trajectory configurations is shown in Fig. 2.6.

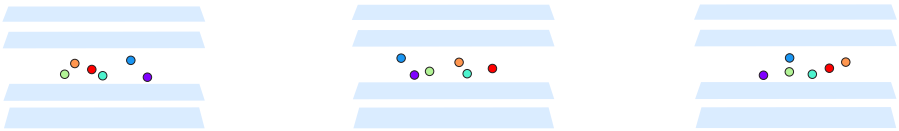


Figure 2.6: Three equivalent infinitely long and tall corridors.

2.5.2 3D rank-2: Parallelepipeds without bases

1. *Linear dependence:* Assume that the first and the second column are linearly independent, and the others are their linear combinations. Thus, for every wall k we have

$$\begin{bmatrix} \cos\theta_k^0 \\ \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \\ \cos\theta_k \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \sin\theta_k^0 \cos\varphi_k^0 \\ \sin\theta_k^0 \sin\varphi_k^0 \end{bmatrix}, \text{ where } \boldsymbol{T} = \begin{bmatrix} a & b \\ c & d \\ e & f \\ g & h \end{bmatrix}. \tag{2.40}$$

2. *Reparametrization:* As before, (2.40) implies a relationship between the normals of the reference and the equivalent room,

$$\begin{bmatrix} \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \\ \cos\theta_k \end{bmatrix} = \begin{bmatrix} c & d & 0 \\ e & f & 0 \\ g & h & 0 \end{bmatrix} \begin{bmatrix} \sin\theta_k^0 \cos\varphi_k^0 \\ \sin\theta_k^0 \sin\varphi_k^0 \\ \cos\theta_k^0 \end{bmatrix}. \tag{2.41}$$

By setting e, g and h to 0, we obtain the desired upper triangular matrix and propagate this change into $\boldsymbol{T}$,

$$\boldsymbol{T} = \begin{bmatrix} a & c & 0 & 0 \\ b & d & f & 0 \end{bmatrix}^\top.$$

3. *Reference room:* The sum of the squares of the last three equations in (2.40) has to be 1 for every wall k,

$$(c \sin\theta_k^0 \cos\varphi_k^0 + d \sin\theta_k^0 \sin\varphi_k^0)^2 + (f \sin\theta_k^0 \sin\varphi_k^0)^2 = 1, \tag{2.42}$$

so the reference room cannot be chosen arbitrarily. From (2.42), we can express θ_k^0 as a function of φ_k^0 and the entries of $\boldsymbol{T}$.

The first equation in (2.40) additionally constrains θ_k^0 and φ_k^0,

$$\tan\theta_k^0 = (a\cos\varphi_k^0 + b\sin\varphi_k^0)^{-1}. \tag{2.43}$$

We obtain a quadratic equation with respect to $\cos(2\varphi_k^0)$,

$$(A^2+B^2)\cos^2(2\varphi_k^0) - 2AC\cos(2\varphi_k^0) + (C^2-B^2) = 0, \tag{2.44}$$

where

$$\begin{aligned} A &= -a^2+b^2+c^2-d^2-f^2, \\ B &= 2(ab-cd), \\ C &= a^2+b^2-c^2-d^2-f^2+2. \end{aligned} \tag{2.45}$$

We first assume $A^2+B^2 \neq 0$ and solve (2.44) for φ_k^0,

$$\cos(2\varphi_k^0) = \frac{AC \pm \sqrt{A^2C^2-(A^2+B^2)(C^2-B^2)}}{A^2+B^2}. \tag{2.46}$$

We obtain four solutions for φ_k^0 to (2.44) that satisfy (2.40). For each value of φ_k^0 we can find the corresponding θ_k^0 from (2.42) or (2.43). Valid solutions always generate two pairs of wall normals: $\{\theta_k^0,\varphi_k^0\}_{k=1}^2 = \{(\theta_1^0,\varphi_1^0),(-\theta_1^0,\varphi_1^0+\pi)\}$ and $\{\theta_k^0,\varphi_k^0\}_{k=3}^4 = \{(\theta_3^0,\varphi_3^0),(-\theta_3^0,\varphi_3^0+\pi)\}$. Therefore, each reference room is made of two arbitrarily chosen walls and two walls parallel to them, resulting in parallelepipeds without its two bases.

As the case of $A^2+B^2=0$ results in rather different geometries, it is analyzed separately in Section 2.5.3.

4. *Equivalent rooms:* The corresponding angles in the equivalent room are computed from (2.40),

$$\theta_k = \pi/2, \quad \varphi_k = g(\theta_k^0,\varphi_k^0,c,d,f) + s_k\pi, \tag{2.47}$$

where $s_k \in \{0,1\}$ and

$$g(\theta_k^0,\varphi_k^0,c,d,f) = \arctan\frac{f\sin\theta_k^0\sin\varphi_k^0}{c\sin\theta_k^0\cos\varphi_k^0 + d\sin\theta_k^0\sin\varphi_k^0}.$$

5. *Corresponding trajectories:* The nullspace of $\overline{\boldsymbol{N}}^\top$ is spanned by four vectors,

$$\begin{aligned} \boldsymbol{v}_1 &= \begin{bmatrix} -a & -b & 1 & 0 & 0 & 0 \end{bmatrix}^\top, & \boldsymbol{v}_2 &= \begin{bmatrix} -c & -d & 0 & 1 & 0 & 0 \end{bmatrix}^\top, \\ \boldsymbol{v}_3 &= \begin{bmatrix} 0 & -f & 0 & 0 & 1 & 0 \end{bmatrix}^\top, & \boldsymbol{v}_4 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top, \end{aligned}$$

so the waypoints in $\overline{\boldsymbol{R}}$ are related as

$$\begin{bmatrix} \boldsymbol{r}_n^0 \\ -\boldsymbol{r}_n \end{bmatrix} = \gamma_1\boldsymbol{v}_1 + \gamma_2\boldsymbol{v}_2 + \gamma_3\boldsymbol{v}_3 + \gamma_4\boldsymbol{v}_4, \tag{2.48}$$

where $\gamma_1,\gamma_2,\gamma_3$ and $\gamma_4 \in \mathbb{R}$. It follows that the waypoints of the reference room are independent and can be chosen arbitrarily, whereas the corresponding waypoints of the equivalent rooms are given by (2.48).

6. *Equivalence class:* We can set two wall orientations of a reference room by arbitrarily choosing φ_1^0 and φ_3^0, and by computing θ_1^0 and θ_3^0 from (2.43). By solving the system of two equations (2.44) with $k \in \{1,3\}$, we fix two parameters (e.g., c and d) and leave the third parameter (e.g., f) free to generate new rooms equivalent to the reference. Walls parallel to those defined by $(\theta_1^0, \varphi_1^0)$ and $(\theta_3^0, \varphi_3^0)$ are determined by $(-\theta_1^0, \varphi_1^0 + \pi)$ and $(-\theta_3^0, \varphi_3^0 + \pi)$. Recall that the solutions of (2.44) always come in pairs $(\theta_k^0, -\theta_k^0)$ and $(\varphi_k^0, \varphi_k^0 + \pi)$, so adding walls parallel to the two fixed ones does not violate (2.44).

As usual, we can choose $\boldsymbol{q}^0 \in \mathbb{R}^K$ arbitrarily, and define an equivalence class of room-trajectory configurations generated by $\{\mathcal{P}_k^0\}_{k=1}^K$ as described above and arbitrary $\{\boldsymbol{r}_n^0\}_{n=1}^N$ as

$$\begin{aligned}\left[\mathcal{R}^0\right] = \Big\{\mathcal{R} \,\big|\, &\varphi_k = g(\theta_k^0, \varphi_k^0, c, d, f) + s_k\pi, \theta_k = \pi/2, s_k \in \{0,1\},\\ &f \in \mathbb{R}, c, d \in \mathbb{R} \text{ s.t. (2.44) holds}, q_k = q_k^0,\\ &\text{for } 1 \le k \le K, \boldsymbol{r}_n \text{ s.t. (2.48) holds for } 1 \le n \le N\Big\}. \end{aligned} \tag{2.49}$$

An example is illustrated in Fig. 2.7.

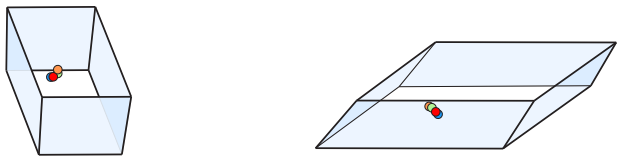


Figure 2.7: Two parallelepipeds without bases with the same PPDM.

2.5.3 3D rank-2: Prisms without bases

In step 3 of the previous case, we studied $A^2 + B^2 \neq 0$. Now we focus on $A^2 + B^2 = 0$ and omit steps 1, 2 and 5 as they are identical to Section 2.5.2.

3. *Reference room:* The case of $A^2 + B^2 = 0$ leads to $A = B = C = 0$ and (2.44) being satisfied for any value of φ_k^0. By solving $A = B = C = 0$, we find explicit expressions for three dependent parameters in $\boldsymbol{T}$,

$$c = \pm\sqrt{a^2+1},\ d = \pm\frac{ab}{\sqrt{a^2+1}},\ f = \pm\sqrt{b^2 - \frac{a^2b^2}{a^2+1} + 1}. \tag{2.50}$$

Then, from arbitrarily chosen angles φ_k^0, and the parameters in $\boldsymbol{T}$ that satisfy (2.50), we compute θ_k^0 from (2.42) or (2.43). Such a room consists of K walls parallel to a fixed line; this means that every triplet of walls forms a prismatic surface, or equivalently, every wall intersects the other two along lines.

To see this, observe that the rank of the coefficient matrix $\boldsymbol{N}^0$ is 2, while the rank of the augmented matrix $\boldsymbol{M}^0$,

$$\boldsymbol{M}^{0\top} = \begin{bmatrix}\boldsymbol{N}^{0\top} & \boldsymbol{q}\end{bmatrix}, \tag{2.51}$$

can be 2 or 3. Indeed, the coefficient matrix from (2.43) is

$$\boldsymbol{n}_k^0 = \frac{1}{\sqrt{1+(a\cos\theta_k^0 + b\sin\theta_k^0)^2}} \begin{bmatrix} \cos\varphi_k^0 \\ \sin\varphi_k^0 \\ a\cos\varphi_k^0 + b\sin\varphi_k^0 \end{bmatrix}.$$

The third row of ${\boldsymbol{N}^0}^\top$ is a linear combination of the first two rows so $\operatorname{rank}(\boldsymbol{N}^0) = 2$. From (2.51) it follows that $\operatorname{rank}(\boldsymbol{M}^0) = 3$, except for a set of $\boldsymbol{q}$ of Lebesgue measure zero. A specific case of $\operatorname{rank}(\boldsymbol{M}^0) = 2$ occurs when the values of $\boldsymbol{q}$ are chosen so that all walls intersect in one line.

4. *Equivalence rooms:* The angles of the equivalent room θ_k and φ_k are computed from (2.47). We show that the equivalent room is a rotated version of the reference room.

The rotation ambiguity exists despite the reparametrization in step 2 because the normals in any equivalent room lie in a plane (the xy-plane in the reference room). Then, transformation of the normals of $\{\mathcal{P}_k^0\}_{k=1}^K$ to those of $\{\mathcal{P}_k\}_{k=1}^K$ is determined by two angles, instead of three for a general rotation. We can factor any upper triangular matrix into a product of a rotation matrix around two axes and a square matrix by two Givens rotations [65]. Thus, $\boldsymbol{T}$ being upper-triangular still allows for rotations specified by two angles.

We introduce a matrix $\boldsymbol{R} = (r_{ij})_{i,j=1}^3$ such that

$$\begin{bmatrix} \sin\theta_k\cos\varphi_k \\ \sin\theta_k\sin\varphi_k \\ \cos\theta_k \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} \sin\theta_k^0\cos\varphi_k^0 \\ \sin\theta_k^0\sin\varphi_k^0 \\ \cos\theta_k^0 \end{bmatrix} \quad \text{for } 1 \le k \le K.$$

Together with (2.41), we obtain

$$\begin{aligned} c &= r_{11} + a r_{13}, \quad & d &= r_{12} + b r_{13}, \quad & 0 &= r_{21} + a r_{23}, \\ f &= r_{22} + b r_{23}, \quad & 0 &= r_{31} + a r_{33}, \quad & 0 &= r_{32} + b r_{33}, \end{aligned}$$

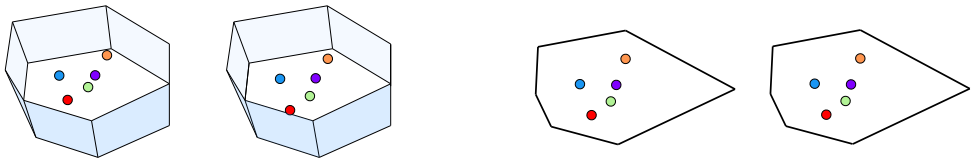
so we can rewrite $\boldsymbol{R}$ as

$$\boldsymbol{R} = \begin{bmatrix} c - a r_{13} & d - b r_{13} & r_{13} \\ -a r_{23} & f - b r_{23} & r_{23} \\ -a r_{33} & -b r_{33} & r_{33} \end{bmatrix} = \begin{bmatrix} c & d & 0 \\ 0 & f & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} \begin{bmatrix} a & b & -1 \end{bmatrix}. \tag{2.52}$$

To see that $\boldsymbol{R}$ is a rotation, note that from $A = B = C = 0$, (2.52), and (2.45), the columns of $\boldsymbol{R}$ are orthonormal.

5. *Corresponding trajectories:* The dependence of the corresponding waypoints is given in (2.48) with an additional constraint on the parameters in (2.50). Intuitively, any waypoint that lies on a line parallel to walls generates the same PPDM.

6. *Equivalence classes:* It follows that two equivalent rooms in the rank-2 case in 3D with $A^2 + B^2 = 0$ have identical geometries, but could have different waypoints lying on a line parallel to all walls; see Fig. 2.8.



a) b)

Figure 2.8: Two prisms without bases with the same PPDM. a) The rooms are identical, but the waypoints differ. b) A bird's eye view. The configurations from this angle seem identical.

2.5.4 3D rank-3: Miscellaneous geometries

1. *Linear dependence:* The practically relevant shoebox rooms generate configurations that are not uniquely determined by PPDMs. For $\operatorname{rank}(\overline{\boldsymbol{N}}) = 3$,

$$\begin{bmatrix} \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \\ \cos\theta_k \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \sin\theta_k^0 \cos\varphi_k^0 \\ \sin\theta_k^0 \sin\varphi_k^0 \\ \cos\theta_k^0 \end{bmatrix}, \quad \text{where } \boldsymbol{T} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}. \tag{2.53}$$

2. *Reparametrization:* As usual, we make $\boldsymbol{T}$ upper triangular matrix by setting d, g and h to 0.

3. *Reference room:* Since in (2.53) we have three equations with four angles for every wall k, we can express θ_k^0, θ_k and φ_k in terms of an arbitrarily chosen angle φ_k^0 and the parameters in $\boldsymbol{T}$. Squaring and summing (2.53) gives

$$0 = (A^2 + B^2)\sin^2(2\theta_k^0) + 2(A + 2C)B\sin(2\theta_k^0) + 4C(A + C), \tag{2.54}$$

where

$$\begin{aligned} A &= a^2\cos^2\varphi_k^0 + (b^2 + e^2)\sin^2\varphi_k^0 + 2ab\sin\varphi_k^0\cos\varphi_k^0 - C - 1, \\ B &= ac\cos\varphi_k^0 + (bc + ef)\sin\varphi_k^0, \\ C &= c^2 + f^2 + i^2 - 1. \end{aligned}$$

To find θ_k^0, we solve (2.54) and obtain

$$\cos(2\theta_k^0) = x_1 \quad \text{or} \quad \cos(2\theta_k^0) = x_2, \tag{2.55}$$

with

$$x_{1,2} = \frac{A(A + 2C) \pm B\sqrt{B^2 - 4AC - 4C^2}}{A^2 + B^2}. \tag{2.56}$$

We first consider $A^2 + B^2 \neq 0$, while the case of $A^2 + B^2 = 0$ is analyzed separately in Section 2.5.5. Analogously to the rank-2 case in 2D or 3D, not all solutions to (2.55) satisfy (2.53); the four valid values of θ_k^0 are identified by verifying

$$1 = (a\sin\theta_k^0\cos\varphi_k^0 + b\sin\theta_k^0\sin\varphi_k^0 + c\cos\theta_k^0)^2 + (e\sin\theta_k^0\sin\varphi_k^0 + f\cos\theta_k^0)^2 + i^2\cos^2\theta_k^0, \tag{2.57}$$

for $1 \leq k \leq K$. Contrary to the rank-2 case in 2D or 3D, the values of A, B and C in (2.55) depend on φ_k^0 and the solutions to (2.55) vary for different walls k. We denote them $\theta_{k,1}^0$, $\theta_{k,2}^0$,

$\theta^0_{k,3}$ and $\theta^0_{k,4}$, where $\theta^0_{k,1}$ and $\theta^0_{k,2}$ are computed from x_1, while $\theta^0_{k,3}$ and $\theta^0_{k,4}$ from x_2. They satisfy $\theta^0_{k,2} = \theta^0_{k,1} + \pi$ and $\theta^0_{k,4} = \theta^0_{k,3} + \pi$.

For some fixed parameters in $\boldsymbol{T}$, there are infinitely many ways to arrange the walls of the reference room. The angles $\{\varphi^0_k\}_{k=1}^K$ are chosen from $[0, 2\pi)$, while $\{\theta^0_k\}_{k=1}^K$ are computed from (2.55) and (2.56). For any arbitrarily chosen φ^0_k, there are four values of θ^0_k that satisfy (2.57), $\theta^0_{k,1} \ldots, \theta^0_{k,4}$. This allows us to create up to four different walls for one fixed value of φ^0_k. For example, in one case, for a chosen φ^0_k we can pick only one value θ^0_{k,j_k}, $1 \leq j_k \leq 4$, and create wall normals $\{\theta^0_{k,j_k}, \varphi^0_k\}_{k=1}^K$. Such rooms have different angles for every wall. In another case, some rooms can have one value φ^0_k associated to four walls, $\{\theta^0_{k,1}, \varphi^0_k\}_{k=1}^{K/4}$, $\{\theta^0_{k,2}, \varphi^0_k\}_{k=1}^{K/4}$, $\{\theta^0_{k,3}, \varphi^0_k\}_{k=1}^{K/4}$ and $\{\theta^0_{k,4}, \varphi^0_k\}_{k=1}^{K/4}$.

We denote the number of independent walls (i.e., different values of φ^0_k) by K_0. We can create reference rooms for any K_0 and given $\boldsymbol{T}$ by the procedure described above. However, as the parameters in $\boldsymbol{T}$ need to be fixed to compute θ^0_k from φ^0_k, for every reference room there is only one equivalent room, computed from (2.53).

A result that is in line with our previous analysis occurs for $K_0 < 6$. Then, we can construct a reference room with *any* K_0 walls $\{\theta^0_k, \varphi^0_k\}_{k=1}^{K_0}$. Furthermore, we can add walls parallel to these K_0 walls without additionally constraining (2.57), so we can have up to $K = 2K_0 < 12$ walls in the reference room. Then, we can solve the system of K_0 equations (2.57) with $1 \leq\ k \leq K_0$ to find K_0 dependent parameters in $\boldsymbol{T}$, and generate new equivalent rooms from (2.53) by changing the remaining $6 - K_0$ free parameters in $\boldsymbol{T}$.

4. *Equivalent rooms:* The transformation to equivalent rooms is the same for all reference rooms and we find it from (2.53),

$$\theta_k = t_k f(\theta^0_k, i), \qquad \varphi_k = g(\theta^0_k, \varphi^0_k, \boldsymbol{T}) + s_k \pi, \tag{2.58}$$

where

$$f(\theta^0_k, i) = \arccos(i \cos\theta^0_k),$$
$$g(\theta^0_k, \varphi^0_k, \boldsymbol{T}) = \arctan \frac{e \sin\theta^0_k \sin\varphi^0_k + f \cos\theta^0_k}{\sin\theta^0_k (a \cos\varphi^0_k + b \sin\varphi^0_k) + c \cos\theta^0_k},$$

$t_k \in \{-1, 1\}$ and $s_k \in \{0, 1\}$. The choice of t_k uniquely determines s_k, such that (2.53) is satisfied.

5. *Corresponding trajectories:* The nullspace of $\overline{\boldsymbol{N}}^\top$ is spanned by three vectors in all of the aforementioned cases,

$$\boldsymbol{v}_1 = \begin{bmatrix} -a, & -b, & -c, & 1, & 0, & 0 \end{bmatrix}^\top, \boldsymbol{v}_2 = \begin{bmatrix} 0, & -e, & -f, & 0, & 1, & 0 \end{bmatrix}^\top, \boldsymbol{v}_3 = \begin{bmatrix} 0, & 0, & -i, & 0, & 0, & 1 \end{bmatrix}^\top.$$

Then,

$$\begin{bmatrix} \boldsymbol{r}^0_n \\ -\boldsymbol{r}_n \end{bmatrix} = \boldsymbol{v}_1 \gamma_1 + \boldsymbol{v}_2 \gamma_2 + \boldsymbol{v}_3 \gamma_3, \tag{2.59}$$

where γ_1, γ_2 and $\gamma_3 \in \mathbb{R}$. The waypoints in one room are chosen arbitrarily and a non-rigid transformation $\boldsymbol{T}^\top$ is applied to compute the waypoints in the equivalent room, $\boldsymbol{r}^0_n = \boldsymbol{T}^\top \boldsymbol{r}_n$.

6. *Equivalence class:* An equivalence class of room-trajectory configurations with respect to PPDM is given as

$$\begin{aligned}\left[\mathcal{R}^0\right] = \Big\{\mathcal{R} \,\big|\, &\varphi_k = g(\theta_k^0, \varphi_k^0, \boldsymbol{T}) + s_k\pi, \theta_k = t_k f(\theta_k^0, i), a, b, c, e, f, i \in \mathbb{R} \\ &\text{s.t. (2.57) holds}, t_k \in \{-1, 1\}, s_k \in \{0, 1\} \text{ s.t. (2.53) holds}, \\ &q_k = q_k^0, \text{ for } 1 \le k \le K, \boldsymbol{r}_n \text{ s.t. (2.59) holds for } 1 \le n \le N\Big\}, \end{aligned} \tag{2.60}$$

where the waypoints in the reference room are chosen arbitrarily, while the reference room $\{\mathcal{P}_k^0\}_{k=1}^K$ can have at most $K_0 < 6$ arbitrarily chosen walls and K_0 walls parallel to them.

Two room-trajectory configurations that correspond to $K_0 = K = 6$ are shown in Fig. 2.9. The angles $\{\varphi_k^0\}_{k=1}^K$ of the reference room (left) and the parameters in $\boldsymbol{T}$ are chosen arbitrarily, while $\theta_{k_j}^0$ for $j = 1, \ldots, 4$ are computed from (2.55) and (2.56). For every k, only one value of $\{\theta_{k_j}^0\}_{k=1}^K$ is assigned to the wall k. For such a reference room, there is only one equivalent room (right), with wall normals from (2.58).

Figure 2.9: A pair of equivalent rooms in 3D.

An example of an arbitrarily chosen room with five walls ($K_0 = K = 5$) together with the two rooms from the same equivalence class (2.60) is shown in Fig. 2.10.

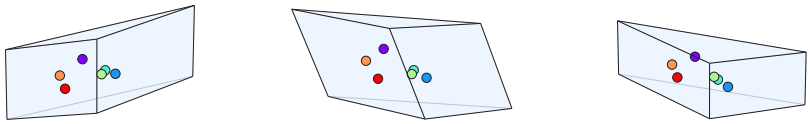


Figure 2.10: Rooms with less than six walls in 3D that belong to the same equivalence class.

Fig. 2.11 illustrates an example of arbitrarily chosen three pairs of parallel walls in a room, $K_0 = 3$, together with the two rooms from the same equivalence class (2.60).

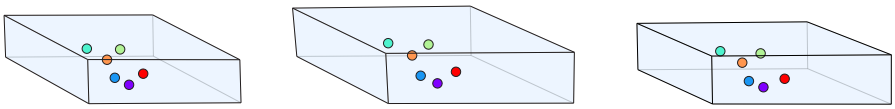


Figure 2.11: An example of equivalent rooms with three pairs of parallel walls.

2.5.5 3D rank-3: Two sets of parallel walls

There is another equivalence class arising from $\operatorname{rank}(\overline{\boldsymbol{N}}) = 3$ for $A^2 + B^2 = 0$ and $\cos\varphi_k^0 \neq 0$. One can show that these constraints lead to rooms with arbitrarily chosen angles θ_k^0 and constant values for φ_k^0 (up to a shift by π), i.e., rooms with all walls parallel to a line. An analysis similar

to that in Section 2.5.3 shows that the rooms in the same equivalence class are simply rotated versions of the reference room.

3. *Reference room:* We continue with $A^2+B^2=0$ which implies $A=B=C=0$, and in addition we assume that $\cos\varphi_k^0=0$. We omit steps 1, 2 and 5 as they are identical to Section 2.5.3. From $B=0$, it follows that

$$ac=0 \quad \text{and} \quad bc+ef=0. \tag{2.61}$$

From (2.61), we conclude that either $a\neq 0, c=0$, or $a=0, c\neq 0$, or $a=c=0$. The last two cases are not of our interest as $a=0$ implies that the x coordinates of $\boldsymbol{r}_n^0$ are 0, and the points lie in the yz-plane. Such a degenerate trajectory is covered in our next case, $\operatorname{rank}(\overline{\boldsymbol{N}})=4$, so we do not study it further here. A similar observation can be made for $a\neq 0, c=0, e=0$; the y coordinates of $\boldsymbol{r}_n^0$ are proportional to their x coordinates, so the points lie in a plane, which corresponds to $\operatorname{rank}(\overline{\boldsymbol{N}})=4$.

A new equivalence class arises for $a\neq 0, c=0, f=0$. From $C=0$, we obtain that $i=\pm 1$, while $A=0$ defines φ_k^0,

$$(a\cos\varphi_k^0+b\sin\varphi_k^0)^2+e^2\sin^2\varphi_k^0=1. \tag{2.62}$$

By introducing $u=\tan\frac{\varphi_k^0}{2}$ and $z=\frac{u^2-1}{u}$, we can find the solutions of (2.62) in terms of z,

$$z_{1,2}=\frac{2ab\pm 2\sqrt{-a^2e^2+a^2+b^2-e^2-1}}{a^2-1},$$

from which we can express the four solutions of φ_k^0,

$$\varphi_k^0=2\arctan\frac{z_i\pm\sqrt{z_i^2+4}}{2},$$

for $i\in\{1,2\}$. We observe that the normals computed from z_1 generate rooms with walls parallel to a certain line ℓ_1. Analogously, the normals generated by z_2 are parallel to another line ℓ_2. Therefore, to construct the reference room, we can arbitrarily choose two values φ_1^0 and φ_2^0 from $[0,2\pi)$ and K values of the angle $\{\theta_k^0\}_{k=1}^K$ from $[0,\pi)$. Then, we match K_1 values of $\{\theta_k^0\}_{k=1}^{K_1}$ with φ_1^0 and the remaining $K_2=K-K_1$ values of $\{\theta_k^0\}_{k=K_1+1}^{K_2}$ with φ_2^0. This results in a room with the two sets of walls, where all walls in one set are parallel to a line. We can then solve the system of two equations (2.62) for φ_1^0 and φ_2^0, to find the values of two parameters (e.g. a and b) and leave the third one (e.g. e) free to generate equivalent rooms.

4. *Equivalent room:* We find the equivalent rooms from (2.53) by the same computations as in Section 2.5.4.

6. *Equivalence class:* The equivalence class also corresponds to the one in Section 2.5.4 with $c=0, f=0$ and $i=\pm 1$. The free parameter e generates equivalent room-trajectory configurations,

$$\begin{aligned}\left[\mathcal{R}^0\right]=\Big\{\mathcal{R}\,\big|\,&\varphi_k=g(\theta_k^0,\varphi_k^0,s_k,a,b,c=0,e,f=0),\theta_k=f(\theta_k^0,t_k,i=\pm 1),\\ &t_k,s_k\in\{0,1\}\text{ s.t. (2.53) holds},a,b\in\mathbb{R}\text{ s.t. (2.62) holds},e\in\mathbb{R},\\ &q_k=q_k^0,\ \text{for } 1\leq k\leq K,\boldsymbol{r}_n\text{ s.t. (2.59) holds for } 1\leq n\leq N\Big\},\end{aligned} \tag{2.63}$$

where the waypoints in the reference room are chosen arbitrarily, while the reference room is constructed from two sets of walls, with walls within each set being parallel to a line.

Note that the walls computed from z_1 do not have to enclose any specific shape, as long as they are equally inclined to all the walls obtained from z_2.

An interesting realistic room that belongs to this class is a room made up of four parallel walls that are perpendicular to the ceiling and the floor. By tilting the ceiling and the floor (changing the value of e), we can generate infinitely many equivalent rooms with respect to PPDM, see Fig. 2.12.

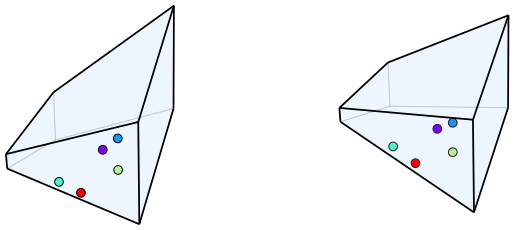


Figure 2.12: Equivalent rooms with two groups of walls enclosing a prismatic surface.

2.5.6 3D rank-4: Planar trajectories

1. *Linear dependence:* To achieve $\operatorname{rank}(\overline{\boldsymbol{N}}) = 4$, we assume that the fourth and the fifth column of $\overline{\boldsymbol{N}}$ are linear combinations of the remaining four,

$$\begin{bmatrix} \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \sin\theta_k^0 \cos\varphi_k^0 \\ \sin\theta_k^0 \sin\varphi_k^0 \\ \cos\theta_k^0 \\ \cos\theta_k \end{bmatrix}, \text{ where } \boldsymbol{T} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \end{bmatrix}. \tag{2.64}$$

2. *Reparametrization:* As $r > D$, we cannot rewrite (2.64) so that the normals of $\mathcal{R}$ and $\mathcal{R}^0$ are on different sides.

3. *Reference room*: In (2.64) we have two equations with four unknown angles for every k. Since the system is underdetermined, we can choose $\left\{\theta_k^0, \varphi_k^0\right\}_{k=1}^K$ arbitrarily.

4. *Equivalent rooms*: We solve (2.64) for θ_k and φ_k, and express their dependence on θ_k^0, φ_k^0 and the parameters in $\boldsymbol{T}$,

$$\theta_k = s_k f(\theta_k^0, \varphi_k^0, \boldsymbol{T}), \qquad \varphi_k = t_k h(\theta_k^0, \varphi_k^0, \boldsymbol{T}), \tag{2.65}$$

where $s_k, t_k \in \{-1, 1\}$, and

$$f(\theta_k^0, \varphi_k^0, \boldsymbol{T}) = \arccos \frac{-dG_a - hG_e \pm \sqrt{G}}{1 + d^2 + h^2}, \quad h(\theta_k^0, \varphi_k^0, \boldsymbol{T}) = \arccos \frac{d\cos\theta_k + G_a}{\sin\theta_k},$$

and we introduced the following shortcuts:

$$\begin{aligned} G_a &:= a\sin\theta_k^0\cos\varphi_k^0 + b\sin\theta_k^0\sin\varphi_k^0 + c\cos\theta_k^0, \\ G_e &:= e\sin\theta_k^0\cos\varphi_k^0 + f\sin\theta_k^0\sin\varphi_k^0 + g\cos\theta_k^0, \\ G &:= (dG_a + hG_e)^2 - (1 + d^2 + h^2)(G_a^2 + G_e^2 - 1). \end{aligned}$$

5. *Corresponding trajectories:* The nullspace of $\overline{\boldsymbol{N}}^\top$ is spanned by two vectors,

$$\boldsymbol{v}_1 = \begin{bmatrix} -a, & -b, & -c, & 1, & 0, & -d \end{bmatrix}^\top, \boldsymbol{v}_2 = \begin{bmatrix} -e, & -f, & -g, & 0, & 1, & -h \end{bmatrix}^\top,$$

so the nth row of $\overline{\boldsymbol{R}}$ is

$$\begin{bmatrix} \boldsymbol{r}_n^0 \\ -\boldsymbol{r}_n \end{bmatrix} = \boldsymbol{v}_1\gamma_1 + \boldsymbol{v}_2\gamma_2, \tag{2.66}$$

where $\gamma_1, \gamma_2 \in \mathbb{R}$. From (2.66) we have that one coordinate of the waypoints $\boldsymbol{r}_n^0$ and $\boldsymbol{r}_n$ is a linear combination of the remaining two, meaning that the waypoints lie in a plane.

6. *Equivalence class:* An equivalence class of room-trajectory configurations generated by $\mathcal{R}^0$ with an arbitrary room $\{\mathcal{P}_k^0\}_{k=1}^K$ and a planar trajectory $\{\boldsymbol{r}_n^0\}_{n=1}^N$ is given as:

$$\begin{aligned}\left[\mathcal{R}^0\right] = \Big\{\mathcal{R} \,\big|\, &\theta_k = s_k f(\theta_k^0, \varphi_k^0, \boldsymbol{T}), \varphi_k = t_k h(\theta_k^0, \varphi_k^0, \boldsymbol{T}), a, b, c, d, e, f, g, h \in \mathbb{R}, \\ &s_k, t_k \in \{0, 1\}, q_k = q_k^0 \text{ for } 1 \leq k \leq K, \boldsymbol{r}_n \text{ s.t. (2.66) holds for } 1 \leq n \leq N\Big\}.\end{aligned} \tag{2.67}$$

We conclude that for arbitrarily chosen wall normals of the reference room, we can always find another room with identical distance measurements, as long as the trajectories in both rooms are planar, as in Fig. 2.13.

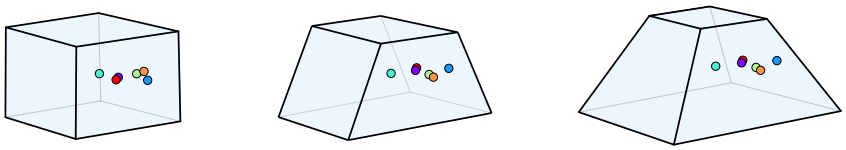


Figure 2.13: Rooms with planar trajectories and the same PPDM.

2.5.7 3D rank-5: Linear trajectories

1. *Linear dependence:* Finally, let $\operatorname{rank}(\overline{\boldsymbol{N}}) = 5$, so that one column of $\overline{\boldsymbol{N}}^\top$ is a linear combination of the remaining independent columns,

$$\cos\theta_k = \boldsymbol{T} \begin{bmatrix} \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \\ \sin\theta_k^0 \cos\varphi_k^0 \\ \sin\theta_k^0 \sin\varphi_k^0 \\ \cos\theta_k^0 \end{bmatrix}, \text{ where } \boldsymbol{T} = \begin{bmatrix} a & b & c & d & e \end{bmatrix}. \tag{2.68}$$

2. *Reparametrization:* Since $r > D$, this step is a no-op.

3. *Reference room:* From (2.68), we can choose walls of the reference room arbitrarily.

4. *Equivalent rooms:* Furthermore, we can express θ_k as a function of φ_k, θ_k^0, φ_k^0 and the parameters in $\boldsymbol{T}$,

$$\theta_k = f(\varphi_k, \theta_k^0, \varphi_k^0, s_k, \boldsymbol{T}), \tag{2.69}$$

where

$$\begin{aligned} h(\varphi_k, a, b) &= a\cos\varphi_k + b\sin\varphi_k, \\ g(\varphi_k, \theta_k^0, \varphi_k^0, \boldsymbol{T}) &= \arccos \frac{d\sin\theta_k^0\sin\varphi_k^0 + e\cos\theta_k^0 + c\sin\theta_k^0\cos\varphi_k^0}{\sqrt{h(a,b,\varphi_k)^2+1}}, \\ f(\varphi_k, \theta_k^0, \varphi_k^0, s_k, \boldsymbol{T}) &= s_k g(\varphi_k, \theta_k^0, \varphi_k^0, \boldsymbol{T}) - \arctan h(\varphi_k, a, b), \end{aligned}$$

with $s_k \in \{-1, 1\}$.

5. *Corresponding trajectories:* The nullspace of $\overline{\boldsymbol{N}}^\top$ is spanned by

$$\boldsymbol{v}_1 = \begin{bmatrix} -c, & -d, & -e, & -a, & -b, & 1 \end{bmatrix}^\top,$$

so the columns of $\overline{\boldsymbol{R}}$ have to be of the form

$$\begin{bmatrix} \boldsymbol{r}_n^0 \\ -\boldsymbol{r}_n \end{bmatrix} = \boldsymbol{v}_1\gamma, \tag{2.70}$$

where $\gamma \in \mathbb{R}$. The above equation can be rewritten as $\boldsymbol{r}_n = \boldsymbol{S}\boldsymbol{r}_n^0$, where $\boldsymbol{S}$ is a scaling matrix with $-\frac{a}{c}$, $-\frac{b}{d}$ and $\frac{1}{e}$ on a diagonal. Moreover, x and y coordinates of the waypoints $\{\boldsymbol{r}_n^0\}_{n=1}^N$ are only scaled values of the z coordinates, so the trajectories are linear.

6. *Equivalence class:* A room-trajectory configuration $\mathcal{R}^0$ with arbitrary wall normals $\{\mathcal{P}_k^0\}_{k=1}^K$ and a linear trajectory $\{\boldsymbol{r}_n^0\}_{n=1}^N$ generates the following equivalence class with respect to PPDMs:

$$\begin{aligned} \left[\mathcal{R}^0\right] = \Big\{ \mathcal{R} \,\big|\, &\theta_k = f(\varphi_k, \theta_k^0, \varphi_k^0, s_k, \boldsymbol{T}), \varphi_k \in [0, 2\pi\}, a, b, c, d, e \in \mathbb{R}, s_k \in \{-1, 1\}, \\ &q_k = q_k^0, \text{ for } 1 \le k \le K, \boldsymbol{r}_n \text{ s.t. (2.70) holds for } 1 \le n \le N \Big\}. \end{aligned} \tag{2.71}$$

We conclude that for any arbitrarily chosen room, we can always find another room with the same PPDM, as long as the trajectories in both rooms are linear. While linear trajectories may seem a special case of the previous one, the room transformations are rather different. One example of such configurations is illustrated in Fig. 2.14.

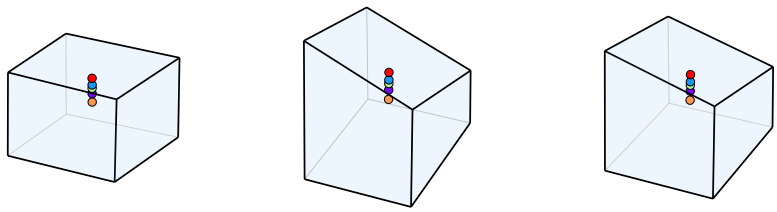


Figure 2.14: Rooms with linear trajectories and the same PPDM.

2.6 Implications on practical development of SLAM algorithms

The analysis in Section 2.4 and Section 2.5 shows that we cannot always uniquely reconstruct points and planes from their pairwise distances. In practice, this indicates that solving the problem of simultaneous localization and mapping with a mobile device that measures its distance from the walls of a room does not have a unique solution in all environments. Here we discuss two directions to reduce the solution space: by adding constraints on the room geometry or by adding constraints on the trajectory of the device.

2.6.1 Constraints on room geometries

As shown in the analysis, linear and planar trajectories introduce additional ambiguities in the room reconstruction, so one should avoid them. Let us thus assume in this discussion that a trajectory is non-linear in 2D and non-planar in 3D, and provide an overview of rooms that cannot be uniquely reconstructed from the distance measurements.

In 2D, this concerns only parallelogram rooms. We showed in Section 2.4.1 that they have one degree of freedom to generate new rooms within the same equivalence class. Hence, if one knows that the reconstruction takes places in a rectangular room, it is sufficient to fix one angle of the room to be the right angle and obtain a unique reconstruction and localization.

In 3D, rooms from Section 2.5.2 are not uniquely determined by the distance measurements; we call them parallelepipeds without bases. It is clear that in reality, "rooms" without ceiling and floor are not common, but it can happen that the ceiling and the floor are covered by materials that are not reflective (for example, metal mesh system for the ceiling and carpets on the floor), so that the device cannot measure its distance from the ceiling and the floor by radio or sound waves. Similar to parallelograms, we showed that the rooms from the same equivalence class can be generated by changing one degree of freedom, and thus, if one knows that the reconstruction takes places in a room with a rectangular floor plan, it is sufficient to fix one angle of the room to be the right angle and obtain a unique reconstruction and localization.

A similar argument is valid for the rooms from Section 2.5.5 that can be constructed from two independent sets of walls, where all walls within a set are parallel to a line. If one set is floor and ceiling and the other comprises side walls, we get familiar, realistic rooms. If one knows that the floor is perpendicular to the side walls, it is sufficient to fix the angle between two sets of walls to be the right angle to get uniqueness.

Lastly, it is not possible to identify uniqueness conditions for the rooms in Section 2.5.4, but they can be constrained using trajectory information discussed in Section 2.6.2. In conclusion, most rooms we know from daily life belong to one of the three cases mentioned above: parallelograms, parallelepipeds without bases, and two sets of walls parallel to a line. For these rooms, uniqueness is guaranteed already if we fix one degree of freedom, for example one angle. However, as these three cases do not cover *all* rooms affected by non-unique reconstruction, in the following we discuss the possibility of adding constraints on trajectories instead of rooms.

2.6.2 Constraints on trajectories

In addition to constraining rooms, it might be natural to constrain trajectories. In practice, various techniques can be used to get a noisy estimate of the motion of the device, such as odometry, and acquire some knowledge about the trajectories. If this information was noiseless and the

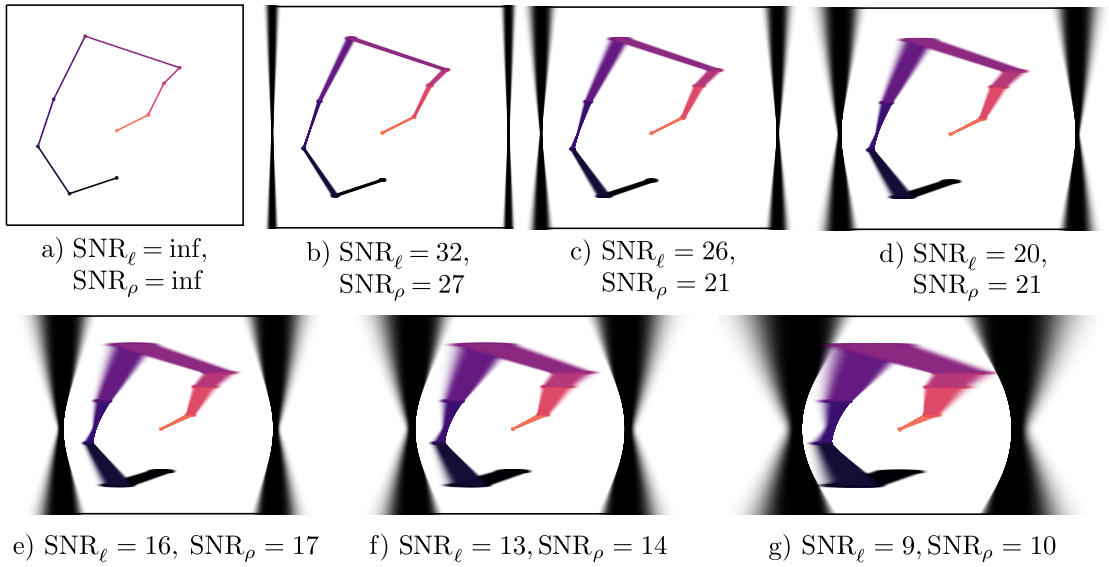


Figure 2.15: Room-trajectory configurations equivalent to a rectangular room for different noise levels in the motion model. We consider independent Gaussian noise on the length and angle of the vectors between two waypoints. The noiseless case in a) shows that there is only one room-trajectory configuration for the given PPDM and the noiseless measurement of the trajectory. The more noise we assume in the motion model, the more room-trajectory configurations become feasible. Different values of signal-to-noise ratio (SNR) of lengths and angles are illustrated from b) to g). The opacity of the rooms and trajectories is proportional to the probability of their realization. All values are in decibels.

trajectories were non-planar, we could uniquely reconstruct all rooms from the distance measurements and known trajectories [94]. Even when noisy, these additional measurements constrain the plausible reconstructions to a neighborhood of the correct one in the room–trajectory space. This is illustrated in Fig. 2.15.

2.7 Conclusion

We derived necessary and sufficient conditions for unique reconstruction of point–plane configurations from their pairwise distances. Our analysis hinges on a new algebraic tool called point-to-plane distance matrix. We exhaustively identify the geometries of points and planes that cannot be distinguished given their PPDMs.

Our motivation comes from the challenging problem of multipath-based simultaneous localization and mapping and our study has consequences for practical indoor localization problems. Picture an unknown room with no preinstalled infrastructure and a mobile device equipped with a single omnidirectional source and a single omnidirectional receiver. The distance measurements between the points and planes are given as the time-of-flights of the first-order echoes recorded by the device. Therefore, our theoretical results provide a fundamental understanding

and constraints under which rooms can uniquely be reconstructed from only first-order echoes.

While our analysis here starts with the PPDM, preparing the PPDM in real scenarios puts forward additional challenges, namely PPDM completion and denoising, and echo sorting. The next chapter includes the development and implementation of computational tools for localization from noisy, incomplete, and unlabeled PPDMs.

2.A Equivalent analysis for different choices of columns

For all D and r we worked with a particular selection of r independent columns. We prove here that this choice can be made without loss of generality. We will call the particular column choice in Sections 2.4 and 2.5 *the original choice*.

First note that there is symmetry between reference and equivalent rooms. For example, for $r = 1$ in 2D, given the original choice of r independent columns we have

$$\left[\sin\varphi_k, \cos\varphi_k^0, \sin\varphi_k^0\right]^\top = \left[a, b, c\right]^\top \cos\varphi_k. \tag{2.72}$$

We can swap the normals $\{\varphi_k^0\}_{k=1}^K$ and $\{\varphi_k\}_{k=1}^K$ for every k, and obtain a new, symmetric choice of r independent columns

$$\left[\sin\varphi_k^0, \cos\varphi_k, \sin\varphi_k\right]^\top = \left[a, b, c\right]^\top \cos\varphi_k^0. \tag{2.73}$$

The two systems (2.72) and (2.73) give the same equivalence class.

A similar conclusion follows if the new choice is obtained by rearranging the order of the coordinates of the normals. Again, for $r = 1$ in 2D we have that

$$\left[\cos\varphi_k^0, \cos\varphi_k, \sin\varphi_k\right]^\top = \left[a, b, c\right]^\top \sin\varphi_k^0$$

can be transformed to the studied case of (2.23). Indeed, by applying a rotation by $\pi/2$ to the normals of the reference room, we obtain a new reference room which satisfies (2.23), but rotated configurations are considered to be equivalent.

In the following we show that any choice of r independent columns not covered by the two previous examples can be transformed into one of the cases analyzed in Sections 2.4 and 2.5 (for $r = 2$ in 2D and $r \in \{2, 3, 4\}$ in 3D).

1. **2D rank-2.** By symmetry, it is sufficient to show that

$$\begin{bmatrix}\cos\varphi_k \\ \cos\varphi_k^0\end{bmatrix} = \begin{bmatrix}a & b \\ c & d\end{bmatrix}\begin{bmatrix}\sin\varphi_k \\ \sin\varphi_k^0\end{bmatrix}$$

can be transformed into (2.14). For $c \neq 0$, it follows directly:

$$\begin{bmatrix}\cos\varphi_k \\ \sin\varphi_k\end{bmatrix} = \frac{1}{c}\begin{bmatrix}a & bc - ad \\ 1 & -d\end{bmatrix}\begin{bmatrix}\cos\varphi_k^0 \\ \sin\varphi_k^0\end{bmatrix}.$$

For $c = 0$ we have $\tan\varphi_k^0 = \frac{1}{d}$, addressed in (2.23).

2. **3D rank-2.** By symmetry, we only analyze

$$\begin{bmatrix}\cos\theta_k^0 \\ \sin\theta_k^0\sin\varphi_k^0 \\ \sin\theta_k\sin\varphi_k \\ \cos\theta_k\end{bmatrix} = \begin{bmatrix}a & b \\ c & d \\ e & f \\ g & h\end{bmatrix}\begin{bmatrix}\sin\theta_k^0\cos\varphi_k^0 \\ \sin\theta_k\cos\varphi_k\end{bmatrix} \tag{2.74}$$

and transform it into (2.40) as

$$\begin{bmatrix} \cos\theta_k^0 \\ \sin\theta_k\cos\varphi_k \\ \sin\theta_k\sin\varphi_k \\ \cos\theta_k \end{bmatrix} = \frac{1}{d}\begin{bmatrix} ad-bc & b \\ -c & 1 \\ ef-cf & f \\ gd-ch & h \end{bmatrix}\begin{bmatrix} \sin\theta_k^0\cos\varphi_k^0 \\ \sin\theta_k^0\sin\varphi_k^0 \end{bmatrix}$$

for $d \neq 0$. If $d = 0$ and $b \neq 0$, a substitution $\sin\theta_k\cos\varphi_k = \frac{1}{b}(\cos\theta_k^0 - a\sin\theta_k^0\cos\theta_k^0)$ from the first equation of (2.74) into the last two equations of (2.74) gives a system equivalent to (2.40). For $b = d = 0$, we get constant normals, discussed in (2.35).

3. **3D rank-3.** Again, we only analyze

$$\begin{bmatrix} \sin\theta_k\cos\varphi_k \\ \sin\theta_k\sin\varphi_k \\ \cos\theta_k^0 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}\begin{bmatrix} \sin\theta_k^0\cos\varphi_k^0 \\ \sin\theta_k^0\sin\varphi_k^0 \\ \cos\theta_k \end{bmatrix} \tag{2.75}$$

and show that we can transform it into (2.53). Indeed, for $i \neq 0$,

$$\begin{bmatrix} \sin\theta_k\cos\varphi_k \\ \sin\theta_k\sin\varphi_k \end{bmatrix} = \frac{1}{i}\begin{bmatrix} ai-cg & bi-ch & c \\ di-fg & ei-fh & f \\ -g & -h & 1 \end{bmatrix}\begin{bmatrix} \sin\theta_k^0\cos\varphi_k^0 \\ \sin\theta_k^0\sin\varphi_k^0 \\ \cos\theta_k^0 \end{bmatrix}.$$

For $i = 0, g \neq 0$ or $i = g = 0, h \neq 0$ we can substitute either $\sin\theta_k^0\cos\varphi_k^0$ or $\sin\theta_k^0\sin\varphi_k^0$ from the last equation of (2.75) into the first two equations of (2.75), getting (2.64). The same holds for $i = g = h = 0$, with an additional constraint $\cos\theta_k^0 = 0$ on the reference normals.

4. **3D rank-4.** Let us assume

$$\begin{bmatrix} \sin\theta_k\cos\varphi_k \\ \sin\theta_k^0\cos\varphi_k^0 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \end{bmatrix}\begin{bmatrix} \sin\theta_k\cos\varphi_k \\ \sin\theta_k^0\sin\varphi_k^0 \\ \cos\theta_k^0 \\ \cos\theta_k \end{bmatrix}. \tag{2.76}$$

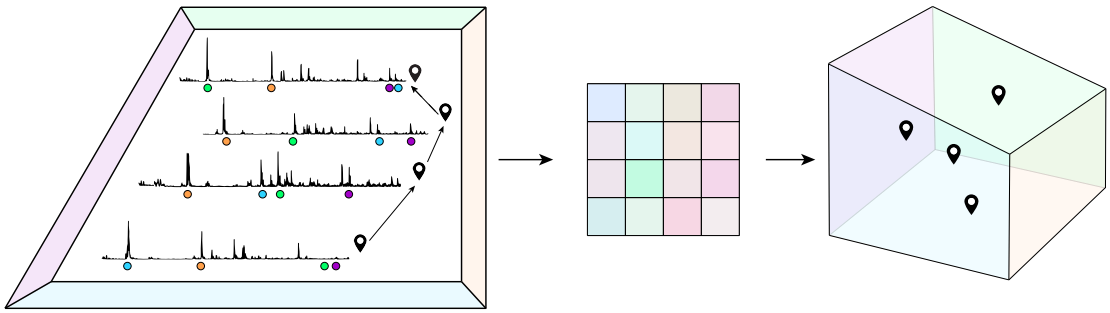
Thanks to symmetry, this is the only case of our interest and we transform it to the well-studied system (2.64) for $e \neq 0$:

$$\begin{bmatrix} \sin\theta_k\cos\varphi_k \\ \sin\theta_k\sin\varphi_k \end{bmatrix} = \frac{1}{e}\begin{bmatrix} a & 1 \\ be-af & -f \\ ce-ag & -g \\ de-ah & -h \end{bmatrix}^\top\begin{bmatrix} \sin\theta_k^0\cos\varphi_k^0 \\ \sin\theta_k^0\sin\varphi_k^0 \\ \cos\theta_k^0 \\ \cos\theta_k \end{bmatrix}.$$

If $e = 0$ and $h \neq 0$, substituting $\cos\theta_k$ from the second into the first equation of (2.76) gives (2.64). By similar substitutions for $e = h = 0$, $f \neq 0$, and $e = f = h = 0$, $g \neq 0$, we get (2.68). Finally, $e = f = h = g = 0$ also corresponds to $r = 5$ in 3D, with an additional constraint $\sin\theta_k^0\cos\varphi_k^0 = 0$.

Chapter 3

Point-to-Plane Distance Matrices in Practice: Room Reconstruction*



No theory is good unless it permits, not rest, but the greatest work. No theory is good except on condition that one use it to go on beyond.

André Gide

3.1 Introduction

Our primary motivation for studying point-to-plane distance matrices (PPDMs) in Chapter 2 comes from simultaneous localization and mapping (SLAM). Location-based services have become an integral part of our daily life but no single technology can provide reliable indoor positioning; thus, we propose a solution which leverages echoes. The goal of this chapter is to show that multipath propagation has a great potential and can be used for indoor positioning [44].

*The material in this chapter is the result of joint work of the author (MK) with Gilles Baechler (GB), Mihailo Kolundžija (MKo), Ivan Dokmanić (ID) and Martin Vetterli (MV) [90, 98]. Author contributions: MK, ID, and MV designed research; MK and MKo devised the echo labeling algorithm from Section 3.3 and undertook real experiments; MK and GB devised the reconstruction algorithm from Section 3.4; MK performed research, implemented all algorithms, ran simulations, processed experimental data, created all figures and wrote the chapter. She would like to express her sincere gratitude to Mihailo Kolundžija for his constructive and valuable feedback which helped to improve the chapter, and his constant support.

One can distinguish between two approaches to acoustic geometry reconstruction problems. The first category includes methods that address the problem from a distance geometry viewpoint; they are based on geometric information obtained from multipath propagation. In the second category we have probabilistic algorithms; they gradually build the map of the room and simultaneously localize the measurement device or a robot within the map. In this thesis, we explore both directions: in the present chapter we rely on the distance geometry of reflections, while in Chapter 4 we consider a probabilistic model. We present the related literature review in the two chapters accordingly.

In both cases, our goal is to make the solution broad and suitable for various setups, so we investigate polygonal rooms of unknown geometries and assume that there is no preinstalled infrastructure to serve as anchors. Moreover, we consider a mobile device with a very basic measurement equipment—one omnidirectional loudspeaker and one omnidirectional microphone. By using such a device to measure room impulse responses (RIRs) along its trajectory, we can obtain propagation times of first-order echoes and convert them to distances between waypoints and walls. Thus, from the acoustic measurements we create PPDMs. While PPDMs present a good basic model for SLAM from echoes, the full SLAM raises a number of additional challenges such as distinguishing first-order from higher-order echoes and associating echoes to walls; these problems are addressed in this chapter.

In the most common probabilistic formulations of SLAM, an initial noisy estimate of the trajectory is required for algorithms to work. In our geometric approach, however, we can assume that the trajectory of a device or its kinematics are a priori completely unknown: the entries of PPDMs are computed only from the propagation times of the first-order echoes. Hence, an algorithm for the factorization of PPDMs jointly recovers rooms and trajectories solely from the corresponding point-to-plane distances. But to obtain the desired distances, one first needs to detect first-order echoes in RIRs and match them with the walls from which they originate, that is, *label* them, which is a hard problem when the trajectory is unknown. There exists a combinatorial solution to the labeling problem [49]. In this chapter, however, we show that if we *do* have some estimate of the trajectory, we can propose a more efficient algorithm to label first-order echoes in measured RIRs, create PPDMs from their propagation times, and jointly recover rooms and localize the device.

3.1.1 Related work

Acoustic room reconstruction as a distance geometry problem. Methods in this category typically assume a set of microphones or speakers with fixed locations, so that the echoes correspond to virtual beacons and their time delays carry geometric information about the system comprised of sources, receivers and a room. The time delays are converted to range measurements, and the problem is recast as retrieving original point configurations in a multidimensional space from a set of pairwise Euclidean distances, which has been thoroughly studied [42, 61, 71, 137, 154].

The described geometric formulation is adopted by many existing solutions to the room reconstruction problem [5, 39, 44, 45, 49, 81, 128, 144, 146, 147, 166]. Most of these techniques rely on fixed microphone or source arrays, and/or known relative positions within the array. In the case of a single channel impulse response [47, 128], the proposed methods require the measurements of all first- and second-order echoes, which is difficult in practice. Other related methods assume rooms to be known and exclusively focus on indoor localization from multipath reflections [85, 86, 123]. We are interested in joint reconstruction of rooms and trajectories with a single mobile device; similar setups have been already studied [102, 140]. Kuang et al.

present a single-antenna anchor-free positioning from acoustic echoes [102]. They assume that emitter is also unknown, but fixed, and jointly reconstruct its location together with the receiver's trajectory and the room geometry under a far-field approximation. The work of Peng et al. [140] assumes a single collocated source and receiver, but the presented method only works in 2D with noiseless distance measurements and known echo labeling.

Low-rank matrix factorization. Many seemingly different problems can be formulated as a PPDM factorization. A famous example in computer vision is "structure from motion" (SFM)—recovery of scene geometry and camera motion from images [169]. Thrun has adapted the original SFM algorithm to the joint localization of microphones and acoustic events with unknown emission times, and named it "structure from sound" (SFS) [167]. When the microphones are synchronized and the sources are in the far field, the times of arrival of the acoustic events can be arranged in a matrix that is essentially a PPDM: it is a low-rank measurement matrix that can be factorized into a product of a coordinate matrix and a column-unitary projection matrix.

The main shortcoming of the Thrun's algorithm [167] is the inability to handle missing data. Similarly, the ellipsoid time difference of arrival (TDOA) method [178] requires not only a set of measurements with no missing values but also exactly three receivers in the plane. A thorough study of this factorization problem is presented by Kuang et al. [101]. The authors implement the original SFS algorithm [167] in 3D and analyze its failure modes. In addition, they propose two minimization strategies—an alternating optimization and a solution relying on the Levenberg–Marquardt algorithm—and evaluate them with regard to accuracy and convergence rate.

3.1.2 Main contributions

As all the aforementioned techniques fail without at least one complete row in a PPDM, we propose an iterative algorithm that can handle these cases and factorize a noisy and incomplete PPDM both in 2D and 3D.

In this chapter we show that acoustic SLAM with a single collocated source and receiver can be cast as localization of points and planes from PPDMs. To transform RIRs to PPDMs, we propose a method that jointly detects first-order echoes in RIRs and assigns them to reflecting walls. To transform noisy and incomplete PPDMs to room-trajectory configurations, we formulate an optimization problem to find the best estimates with respect to the mean squared error. This method can be applied to any of the abovementioned problems (e.g. SFM and SFS). We run a number of numerical simulations to show that our formulation not only works in a regime where existing algorithms fail, but also outperforms them in terms of accuracy. We test our end-to-end pipeline of algorithms using real data and demonstrate that the proposed approach is efficient and robust to noise in indoor environments.

3.2 Problem setup

As in Chapter 2, we consider the room to be a polygon with K sides in 2D or a polyhedron with K faces in 3D, where the sides and faces model walls in a room. We use notation from Chapter 2: a wall $\mathcal{P}_k$ is defined by its unit normal $\boldsymbol{n}_k \in \mathbb{R}^m$ and the distance q_k of the wall from the origin.

According to the image source model [4, 24], we represent reflections inside a room as signals produced by mirror images of the real sources across the walls. Sound propagation is modeled with a family of RIRs, where each RIR $h_n(t)$ is idealized as a train of Dirac delta pulses produced

by the real source and its images, and recorded by the microphone,

$$h_n(t) = \sum_{\ell \geq 0} a_\ell \delta(t - \tau_{n\ell}).$$

The propagation times of echoes are denoted by $\tau_{n\ell}$. The received magnitudes a_ℓ depend on the wall absorption coefficients and the distance of the image source from the microphone[1].

When the RIRs are collected by a collocated microphone and loudspeaker along N waypoints of the trajectory, $\{\boldsymbol{r}_n\}_{n=1}^N$, the propagation times of the first-order echoes reveal the distances between the measurement locations (points $\boldsymbol{r}_n$) and walls (planes $\mathcal{P}_k$), as observed in Chapter 2. Given the set of propagation times $\{\tau_{nk}\}_{n=1,k=1}^{N,K}$ of the first-order echoes, we can compute these point-to-plane distances $\{d_{nk}\}_{n=1,k=1}^{N,K}$ by (2.1). In Chapter 2, we assumed that they have been given to us in a form of a PPDM $\boldsymbol{D}$,

$$\boldsymbol{D} = \mathbf{1}\boldsymbol{q}^\top - \boldsymbol{R}^\top \boldsymbol{N}, \tag{3.1}$$

where $\boldsymbol{q}, \boldsymbol{R}$ and $\boldsymbol{N}$ are defined in (2.3), and we focused on the question of uniqueness: whether a given PPDM $\boldsymbol{D}(\mathcal{R})$ uniquely specifies the room-trajectory configuration $\mathcal{R} = \left(\{\mathcal{P}_k\}_{k=1}^K, \{\boldsymbol{r}_n\}_{n=1}^N\right)$ that generates it. We proved that the answer is negative even beyond rigid transformations; we provided all equivalence classes of rooms and trajectories that lead to the same PPDMs.

This is however not a concern in this chapter. Here we consider any room from the same equivalence class to be a correct solution and turn our focus towards real-world conditions. We aim to reconstruct room-trajectory configurations from a given set of noisy RIRs recorded by a collocated microphone and loudspeaker at several locations inside a room. If we can detect first-order echoes in the RIRs and correctly assign them to the reflecting walls, and if we can factorize a noisy and incomplete PPDM, then we can jointly reconstruct rooms and trajectories from the RIRs. The above problems are illustrated in Fig 3.1 and formalized below.

PROBLEM 3.1 Given a collection of room impulse responses, find a PPDM $\widetilde{\boldsymbol{D}}$ whose entries are the distances of waypoints to the walls.

PROBLEM 3.2 Given a noisy and incomplete PPDM $\widetilde{\boldsymbol{D}}$, estimate the room-trajectory configuration $\mathcal{R}$, such that its PPDM $\boldsymbol{D}(\mathcal{R})$ is close to $\widetilde{\boldsymbol{D}}$ in some chosen norm.

We propose a two-step algorithm to solve Problem 3.1 and Problem 3.2. In Section 3.3, we address Problem 3.1. We first devise a method to detect first-order echoes in RIRs and associate them with the walls from which they originate. Then, we simply construct a PPDM $\widetilde{\boldsymbol{D}}$ from the propagation times of the detected first-order echoes. In Section 3.4 we find a room-trajectory configuration $\mathcal{R}$ from the created $\widetilde{\boldsymbol{D}}$, which solves Problem 3.2. We validate the performance of the algorithm on simulated data in Section 3.5 and demonstrate the applicability of our solution to real-world measurements in Section 3.6.

[1] We assume flat fading, that is, all frequency components of the signal have the same magnitude of attenuation.

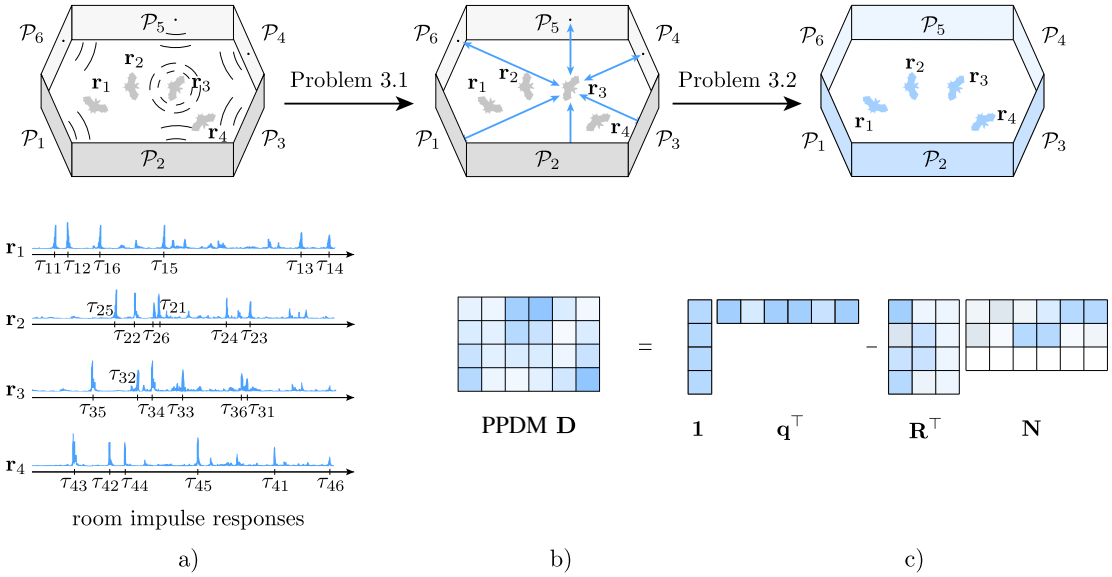


Figure 3.1: An overview of the room-trajectory reconstruction algorithm from the measurements of room impulse responses. a) Collocated omnidirectional source and receiver, embodied as a bat, traverse an unknown room. At randomly chosen waypoints $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_4$, the source emits a pulse and the receiver records RIRs. From RIRs, we detect the pulses that correspond to the first-order echoes and compute their propagation times τ_{nk}, for every $n = 1, \ldots, 4$ and $k = 1, \ldots, 6$. b) The propagation times reveal the distances between the bat and the walls, $d_{nk} = c_s \tau_{nk}/2$, with the speed of sound c_s, arranged in a PPDM $\boldsymbol{D}$. c) We factorize the matrix $\boldsymbol{D}$ as $\boldsymbol{D} = \mathbf{1}\boldsymbol{q}^\top - \boldsymbol{R}^\top \boldsymbol{N}$, to find the unknown walls $\{\mathcal{P}_k\}_{k=1}^6$ specified by $\boldsymbol{N}$ and $\boldsymbol{q}$, and the waypoints $\{\boldsymbol{r}_n\}_{n=1}^4$ specified by $\boldsymbol{R}$.

3.3 From RIRs to PPDMs

In the following, we propose a room model and formulate an efficient and robust algorithm that simultaneously detects and labels first-order echoes in RIRs.

Considerations on measurement setup and room model. We consider a room that consists of a floor, ceiling and any number of side walls. Most real rooms have side walls perpendicular to the floor and ceiling, so we model our rooms as such.

In practical setups, the microphone and the loudspeaker are mounted together on a mobile platform or a robot of a fixed height. This is also the case in our experiments (for more details about the experimental platform see Section 3.6.1). The fact that the z-coordinate in $\boldsymbol{R}$ does not change allows us to fix some degrees of freedom in the reconstruction and achieve a unique solution for some equivalence classes. Moreover, it makes it easy to detect the first-order reflections from the floor and the ceiling; their propagation times are the same at every waypoint, so they appear at the same position in every measured RIR.

By assuming both that the side walls are perpendicular to the floor and ceiling, and that the height of our measurement device does not change along the trajectory, the practical problem effectively becomes 2D. This means that for every wall k, we only need to find two parameters:

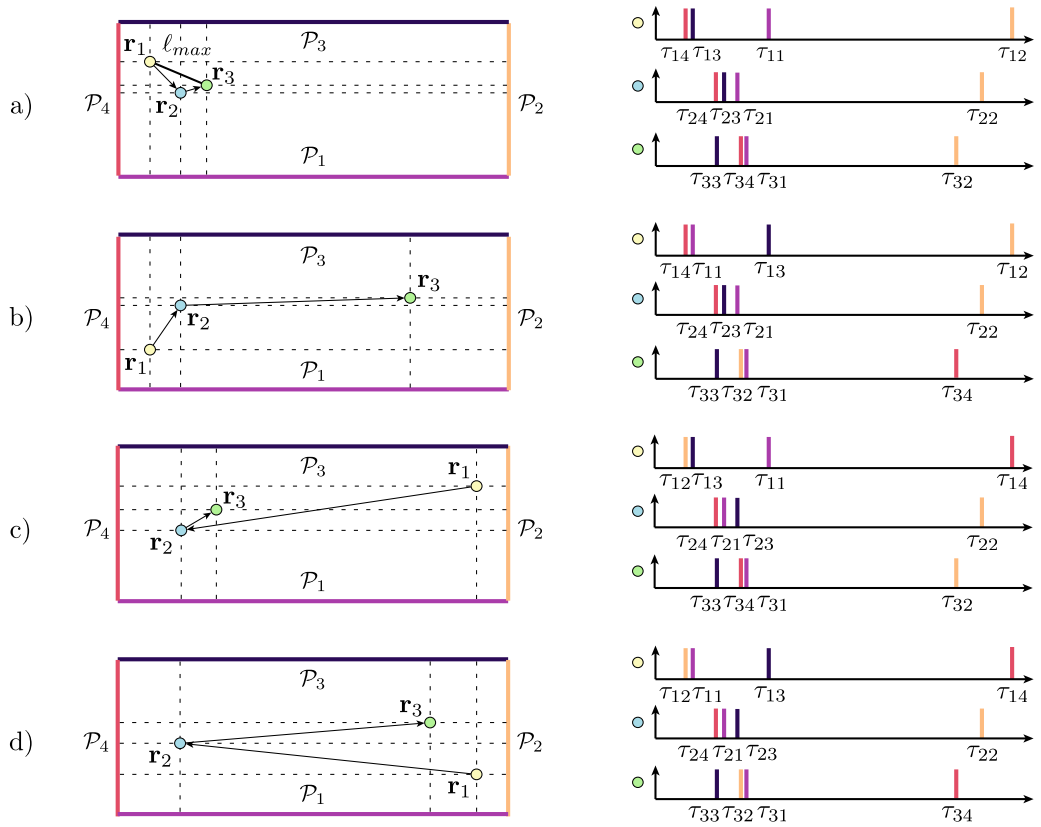


Figure 3.2: *Left*: One room with four different realizations of the trajectory $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3\}$. *Right*: RIRs that correspond to the room-trajectory configurations from the left side, and simplified such that they contain only first-order echoes. For different trajectories from a) to d), the RIRs are identical. The colors of the pulses match with the colors of the walls from which the echoes reflect.

the distance from the origin, q_k, and the angle of the wall normal, φ_k.

Echo labeling: uniqueness claim. Imagine a simple problem in which we know the geometry of a room and we have a set of distances that correspond to the first-order echoes. Our only task is to match these distances with the walls. If our room is for instance a parallelogram and we do not have any information about the trajectory, it is impossible to uniquely resolve the labeling problem and localize the waypoints. The reason is that there exist multiple points inside the room with the same set of distances. An example with four equivalent configurations is given in Fig. 3.2; without knowing the association between the first-order echoes and the walls, all these configurations constitute a valid solution.

We can resolve this ambiguity with additional a priori knowledge. For example, if there is a constraint on the maximum distance between two consecutive waypoint, ℓ_{max}, we can use the triangle inequality to upper bound the difference of the propagation times of two consecutive first-order echoes from the same wall k: $c_s|\tau_{n,k} - \tau_{n+1,k}| \leq 2\ell_{max}$. This can significantly reduce the set of ambiguous solutions in the labeling problem. For example, in Fig. 3.2 we can constrain

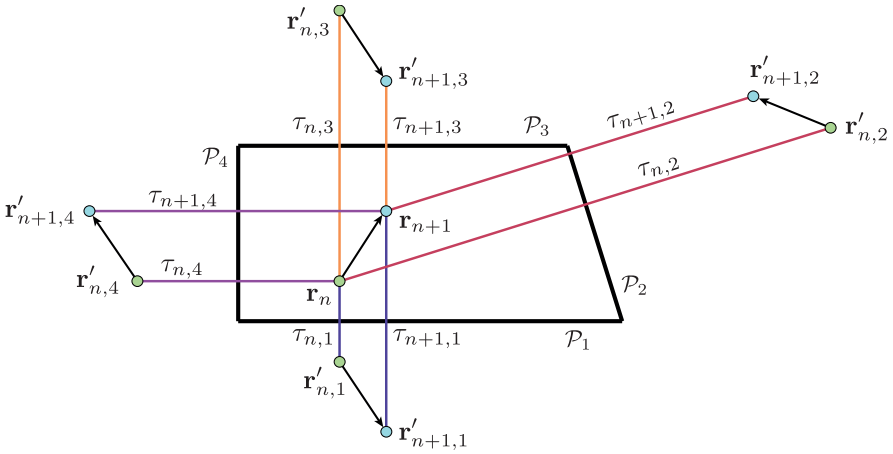


Figure 3.3: Image source model for first-order echoes with collocated microphone and loudspeaker. Sound rays at the measurements n and $n+1$ are shown in the same color for the same wall. For every two consecutive steps, the robot's waypoints, together with their image sources, define isosceles trapezoids with sides of the same length—the length of the robot's step.

the distance between two consecutive waypoints to be smaller than the value of $\ell_{max} = \|\boldsymbol{r}_1 - \boldsymbol{r}_3\|$ from Fig. 3.2a. Then, we can eliminate all rooms illustrated in Fig. 3.2 except Fig. 3.2a; this is the only room for which $|d_{n,k} - d_{n+1,k}| < \ell_{max}$ for every $n = 1, 2, 3$ and $k = 1, 2, 3, 4$.

Under the assumption that the robot's step $\|\boldsymbol{r}_n - \boldsymbol{r}_{n+1}\|$ is known up to some uncertainty, we claim that given the propagation times of the first-order echoes, we can almost always correctly assign them to the corresponding walls. This is based on the fact that the robot's waypoints $\boldsymbol{r}_n$ and $\boldsymbol{r}_{n+1}$, together with their image sources, define isosceles trapezoids with sides of the same length, equal to the length of the robot's step. This is illustrated in Fig. 3.3. As the length of the step is known up to some uncertainty, there is only one way to arrange the given propagation times, $\{\tau_{n-1,k}\}_{k=1}^{K}$ and $\{\tau_{n,k}\}_{k=1}^{K}$, to obtain such isosceles trapezoids.

In practice, we can use different means, such as odometry, control and motion sensors, to measure not only the distance between two waypoints, but also the orientation. Given noisy lengths and orientations, we can estimate a piecewise linear trajectory of the robot, and use it to detect and label first-order echoes in the measured RIRs. In the following sections we propose an algorithm to convert RIRs to PPDMs. It consists of 1) eliminating the echoes from the floor and ceiling, 2) extracting the most prominent peaks from RIRs, and 3) reconstructing the walls.

3.3.1 Elimination of echoes from the floor and ceiling

We assume that we are given a set of recorded RIRs aligned such that the maximum amplitude of every RIR (i.e., the peak that corresponds to the direct sound) is at position $t = 0$.

We assume that the height of the robot is fixed, so the echoes from the floor and ceiling arrive with the same time delays in all RIRs. We denote them by τ_f and τ_c, respectively. In other words, we have $\tau_f = \tau_{f_1} = \ldots = \tau_{f_N}$ and $\tau_c = \tau_{c_1} = \ldots = \tau_{c_N}$. To estimate the delays τ_f and τ_c, we average all RIRs; this reduces the noise and amplifies the amplitude of the reflections from the floor and ceiling relative to other echoes. As the peak with the largest amplitude corresponds

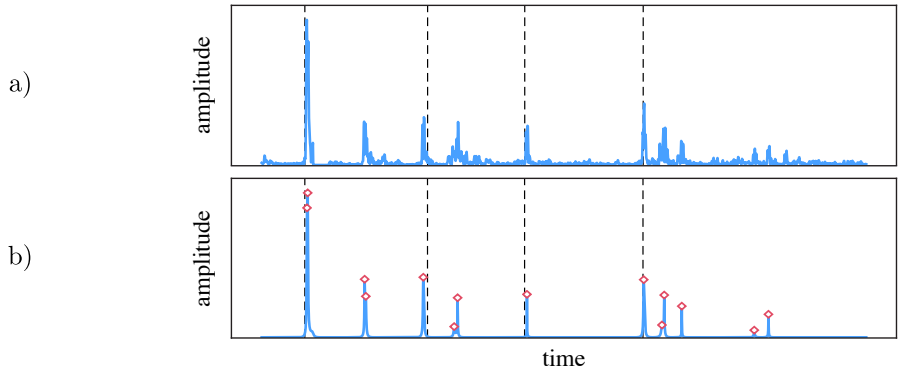


Figure 3.4: a) A squared room impulse response measured at a certain waypoint, where the reflections from the floor and ceiling have been removed. b) Peaks selected from the RIR by applying the variant of the MUSIC algorithm [114]. Red diamonds indicate the locations of the estimated peaks, while the black dashed lines denote the ground truth locations of the first-order echoes from four walls.

to the direct sound, we detect two next largest peaks in the averaged RIRs and associate them to the floor and ceiling.

Before we eliminate the peaks at τ_f and τ_c from every RIR, we need to verify that there is no other echo overlapping with them. To do so, we extract short windows around τ_f and τ_c in the original and averaged RIRs. A high discrepancy between the two windowed signals is a good indicator that there is another pulse nearby or overlapping with the floor or ceiling reflection. We therefore keep it, as it is potentially a first-order echo from a wall. On the contrary, if we observe that the shapes of the windowed signals are similar, we deduce that no other echo is present, and we remove the detected pulse from the RIR as it belongs to the floor or ceiling.

3.3.2 Extraction of the most prominent peaks

The goal of this step is to obtain the times of arrival of the most prominent peaks in RIRs. We apply the variant of the Multiple Signal Classification (MUSIC) algorithm [114] summarized in Appendix 3.A. This method assumes that every RIR consists of M complex exponentials in the presence of Gaussian noise and relies on eigenspace decomposition of the Toeplitz matrix, whose columns are constructed by translates of the squared samples of RIRs in the Fourier domain. The algorithm transforms the recorded RIR into a stream of M Dirac pulses, where some belong to first-order echoes, some to higher-order echoes and some are spurious peaks. We are interested only in their propagation times, τ_{nm}. We arrange them in a vector $\boldsymbol{\tau}_n = [\tau_{n1}, \ldots, \tau_{nM}]$ for every waypoint $n = 1, \ldots, N$, and we construct a matrix $\boldsymbol{T} = [\boldsymbol{\tau}_n]_{n=1}^{N} \in \mathbb{R}^{N \times M}$. Observe that in the noiseless case, all entries of a PPDM are also in $c_s \boldsymbol{T}/2$.

To initialize the MUSIC algorithm, we need to provide a number of peaks to be detected, M. For the next step of our proposed approach, losing a peak from a first-order echo is much more critical than having spurious peaks. Therefore, we keep M large and select up to $M = 30$ strongest pulses in each recording. Fig. 3.4 illustrates an example of a squared RIR along with its MUSIC estimate and the picked times of flight.

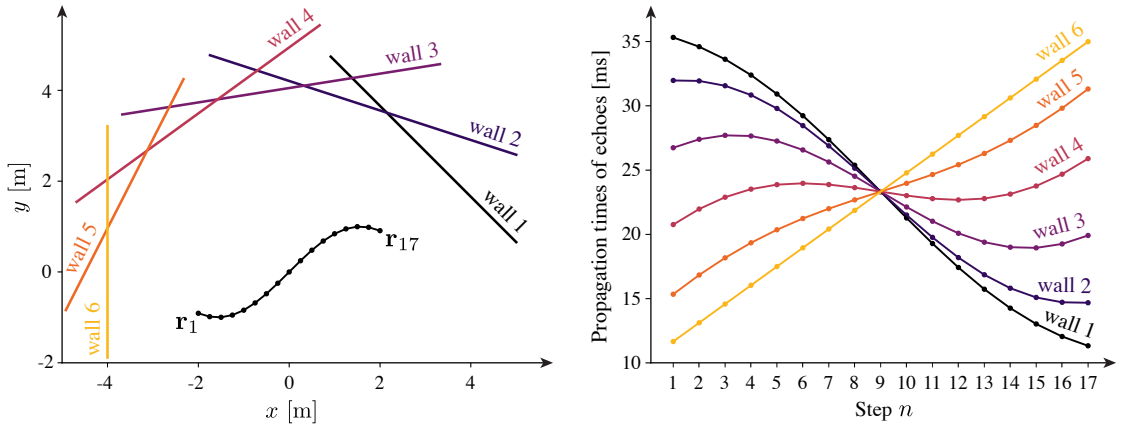


Figure 3.5: Influence of a wall's slope on the measured propagation times of the first-order echoes. a) The trajectory is made of $N = 17$ waypoints. x-coordinates $r_{n,x}$ are distributed uniformly between -2 and 2, while the y-coordinates are $r_{n,y} = \sin(r_{n,x})$. The wall normals are distributed uniformly between $\varphi_1 = \pi/4$ and $\varphi_6 = \pi$. The distance of walls from the origin is the same for all walls ($q_k = 4$). b) Projections $\text{proj}_{\varphi_1}\boldsymbol{R}$ of the trajectories onto six walls.

3.3.3 Reconstruction of the walls

This section describes the core of our reconstruction algorithm that transforms RIRs to PPDMs. It detects first-order echoes in the stream of Diracs from the previous step and labels these echoes with the matching walls, creating a PPDM $\widetilde{\boldsymbol{D}}$. To lay down the main idea of the algorithm, we first assume to have information about the noiseless, ground truth trajectory $\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N]$. Later we relax this assumption and build up the algorithm to take into account the noise on trajectories.

3.3.3.1 Noiseless trajectory

To illustrate the underlying idea, let us first look at the example presented in Fig. 3.5. It shows the time-of-flight measurements of the first-order echoes computed for a fixed trajectory and six walls with different slopes. The trajectory consists of $N = 17$ waypoints $\boldsymbol{r}_n = [r_{n,x}, r_{n,y}]^\top$ defined by $r_{n,x} = -2+(n-1)/4$ and $r_{n,y} = \sin(r_{n,x})$. The wall normals are distributed uniformly between $\varphi_1 = \pi/4$ and $\varphi_6 = \pi$, as depicted in Fig. 3.5a. By projecting the trajectory $\boldsymbol{R}$ onto wall normals,

$$
\begin{aligned}
\varphi_1 = \pi/4: \quad & \text{proj}_{\varphi_1}\boldsymbol{R} = \Big(-2+\frac{n-1}{4}\Big)\cos\frac{\pi}{4} + \sin\Big(-2+\frac{n-1}{4}\Big)\sin\frac{\pi}{4},\\
\varphi_2 = 2\pi/5: \quad & \text{proj}_{\varphi_2}\boldsymbol{R} = \Big(-2+\frac{n-1}{4}\Big)\cos\frac{2\pi}{5} + \sin\Big(-2+\frac{n-1}{4}\Big)\sin\frac{2\pi}{5},\\
& \ldots\\
\varphi_6 = \pi: \quad & \text{proj}_{\varphi_6}\boldsymbol{R} = 2-\frac{n-1}{4},
\end{aligned}
$$

we obtain 6 unique vector components of $\boldsymbol{R}$ in the directions of φ_k, denoted by $\text{proj}_{\varphi_k}\boldsymbol{R}$ and illustrated in Fig. 3.5b. The relation between $\text{proj}_{\varphi_k}\boldsymbol{R}$ and the propagation time τ_{nk} of the first-order echo reflected from the wall k and recorded at the waypoint $\boldsymbol{r}_n$ is:

$$\tau_{nk} = 2(q_k - \text{proj}_{\varphi_k}\boldsymbol{r}_n)/c_s. \tag{3.2}$$

Imagine now that you are given the propagation times $\{\tau_{n1}\}_{n=1}^N$ that correspond to the distances between the waypoints $\{\boldsymbol{r}_n\}_{n=1}^N$ and the first wall, and you need to identify which of the 6 candidate walls is the most likely to give rise to such measurements. An idea based on (3.2) would be to first minimize the disparity between the measurements $\{\tau_{n1}\}_{n=1}^N$ and the precomputed projections $\text{proj}_{\varphi_k}\boldsymbol{R}$ to eliminate an unknown offset q_k for every $k = 1, \ldots, 6$, and then to pick k with the smallest disparity. This is indeed an outline of our algorithm, formally defined in the following.

In practice, we do not have a set of candidate walls to choose from. Instead, we propose to uniformly sample the space of possible angles of the wall normals $\varphi_k \in (0, 2\pi]$ as:

$$\bar{\varphi}_\ell = \ell\Omega_s \text{ for } \ell = 1, \ldots, L,$$

where $L \in \mathbb{N}$ and $\Omega_s \in \mathbb{R}$ is a sampling period such that $L\Omega_s = 2\pi$. We precompute the corresponding projections $\text{proj}_{\bar{\varphi}_\ell}\boldsymbol{R}$ for every $\bar{\varphi}_\ell$ as:

$$\text{proj}_{\bar{\varphi}_\ell}\boldsymbol{R} = \begin{bmatrix}\cos\bar{\varphi}_\ell & \sin\bar{\varphi}_\ell\end{bmatrix}\boldsymbol{R}. \tag{3.3}$$

In the simple case when $\bar{\varphi}_j$ is precisely the angle of the wall normal k,

$$\bar{\varphi}_j = \varphi_k, \tag{3.4}$$

we can find the exact values (up to a shift) of $\text{proj}_{\bar{\varphi}_j}\boldsymbol{R}$ in the measured propagation times of the first-order echoes,

$$q - \text{proj}_{\bar{\varphi}_j}\boldsymbol{r}_n \in c_s\boldsymbol{\tau}_n/2 \text{ for } q \in \mathbb{R} \text{ and every } n = 1, \ldots, N. \tag{3.5}$$

The criterion in (3.5) is satisfied for every pair (j, k) for which (3.4) is true; therefore, if we use (3.5) to find the correct walls $\bar{\varphi}_j$, we do not have to know their number in advance.

In practice, the measurements $\boldsymbol{T}$ are noisy, and our space of discrete angles $\{\bar{\varphi}_\ell\}_{\ell=1}^L$ is finite, so we cannot rely on the criterion in (3.5). Instead, we propose to measure the cumulative mismatch e_ℓ between the projections $\text{proj}_{\bar{\varphi}_\ell}\boldsymbol{R}$ and the closest measured propagation times in $\boldsymbol{T}$. For every waypoint $\boldsymbol{r}_n$ projected onto $\bar{\varphi}_\ell$, the nearest peak in $\boldsymbol{T}$ with respect to the ℓ^2-norm is found by minimizing $|c_s\tau_{nm}/2 - (q - \text{proj}_{\bar{\varphi}_\ell}\boldsymbol{r}_n)|^2$ over $\tau_{n1}, \ldots, \tau_{nM}$ for some fixed q. Therefore, the mismatch e_ℓ can be computed as:

$$e_\ell = \min_q \sum_{n=1}^N \min_{\tau_{n1},\ldots,\tau_{nM}} |c_s\tau_{nm}/2 - (q - \text{proj}_{\bar{\varphi}_\ell}\boldsymbol{r}_n)|^2, \tag{3.6}$$

for every ℓ. The algorithm is summarized in Algorithm 3.1.

The estimated walls are those for which the mismatch is the smallest. More precisely, we refer again to the MUSIC algorithm [114] and apply it to $\boldsymbol{e} = [e_1, e_2, \ldots, e_L]$ to detect K smallest minima in $\boldsymbol{e}$. The indices $\ell_1, \ldots, \ell_K$ indicate the positions of these K local minima in $\boldsymbol{e}$. Hence,

Algorithm 3.1 Reconstruction of the walls.

Input: $\boldsymbol{R}$, $\boldsymbol{T}$

Output: errors $e_1, e_2, \ldots, e_L$

Define a set of discrete angles $\bar{\boldsymbol{\varphi}} = [\bar{\varphi}_1, \bar{\varphi}_2, \ldots, \bar{\varphi}_L]$ with $\bar{\varphi}_\ell = \ell\Omega_s$ for $\ell = 1, \ldots, L$, where $L \in \mathbb{N}$ and $\Omega_s \in \mathbb{R}$ are such that $L\Omega_s = 2\pi$.

Repeat for every $\bar{\varphi}_\ell$ in $\bar{\boldsymbol{\varphi}}$:

Compute $\text{proj}_{\bar{\varphi}_\ell}\boldsymbol{R}$ from (3.3).

Compute e_ℓ from (3.6).

return $e_1, e_2, \ldots, e_L$

the angles of the wall normals with the smallest error are $\widehat{\boldsymbol{\varphi}} = [\bar{\varphi}_{\ell_1}, \ldots, \bar{\varphi}_{\ell_K}]^\top$, or given as normal vectors in matrix form,

$$\widehat{\boldsymbol{N}} = \begin{bmatrix} \cos\bar{\varphi}_{\ell_1} & \cos\bar{\varphi}_{\ell_2} & \ldots & \cos\bar{\varphi}_{\ell_K} \\ \sin\bar{\varphi}_{\ell_1} & \sin\bar{\varphi}_{\ell_2} & \ldots & \sin\bar{\varphi}_{\ell_K} \end{bmatrix}.$$

The values of q that minimize (3.6) for every $\bar{\varphi}_{\ell_k}$ are denoted by $q_{\ell_1}, \ldots, q_{\ell_K}$ and they reveal the estimated distances of the walls from the origin, $\widehat{\boldsymbol{q}} = [q_{\ell_1}, \ldots, q_{\ell_K}]^\top$.

Remark that in Algorithm 3.1 we do not only estimate the walls, but we also detect the first-order echoes in RIRs and label them with their associated walls in the following way. For every estimated wall $\widehat{\varphi}_k \in \widehat{\boldsymbol{\varphi}}$, we obtain $\text{proj}_{\widehat{\varphi}_k}\boldsymbol{R}$ from (3.3) with $\bar{\varphi}_\ell = \widehat{\varphi}_k$, and together with $\hat{q}_k \in \widehat{\boldsymbol{q}}$, in (3.6) we compute:

$$\widetilde{\tau}_{nk} = \underset{\tau_{n1},\ldots,\tau_{nM}}{\arg\min} \; |c_s\tau_{nm}/2 - (\hat{q}_k - \text{proj}_{\widehat{\varphi}_k}\boldsymbol{r}_n)|^2, \tag{3.7}$$

for every n. Hence, for every wall k and every waypoint $\boldsymbol{r}_n$, we find the best matching propagation time in $\boldsymbol{\tau}_n$ as $\widetilde{\tau}_{nk}$. In other words, for every peak in every RIR we can tell if it is a first-order echo, and if so, which is the associated wall of reflection. Note that we can associate one peak to different walls; this is a desired behavior as it can happen that the robot is equally far from several walls, so that the corresponding echoes collide. By arranging the values of the propagation times in a matrix, we construct a PPDM $\widetilde{\boldsymbol{D}}$ as $\widetilde{\boldsymbol{D}} = [c_s\widetilde{\tau}_{nk}/2]_{n=1,k=1}^{N,K}$.

3.3.3.2 Noisy estimate of the trajectory

In real-world scenarios, the sensors are subject to errors, so the estimated trajectory is different from the true path that the device takes. We propose to enhance Algorithm 3.1 to be more robust to noise.

We use $\boldsymbol{v}_n = \boldsymbol{r}_n - \boldsymbol{r}_{n-1}$ to denote the true motion vectors between two consecutive waypoints $\boldsymbol{r}_{n-1}$ and $\boldsymbol{r}_n$ and assume that they are corrupted by i.i.d Gaussian noise, $\widetilde{\boldsymbol{v}}_n = \boldsymbol{v}_n + \boldsymbol{z}_n$, where $\boldsymbol{z}_n \sim \mathcal{N}(\boldsymbol{0}, \sigma_v^2\boldsymbol{I})$. As we estimate $\hat{\boldsymbol{r}}_n$ simply as $\hat{\boldsymbol{r}}_n = \hat{\boldsymbol{r}}_{n-1} + \widetilde{\boldsymbol{v}}_n$, this corresponds to having Gaussian noise $\mathcal{N}(\boldsymbol{0}, n\sigma_v^2\boldsymbol{I})$ on the waypoints directly, which is a realistic model: the error accumulates over time, so the uncertainty of the device's location is larger for every new waypoint.

To gain some intuition for how the noise is hindering the proposed algorithm, we generate P realizations of the trajectory,

$$\widetilde{\boldsymbol{R}}^{[p]} = \boldsymbol{R} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{z}_1 & \boldsymbol{z}_1 + \boldsymbol{z}_2 & \ldots & \sum_{n=1}^{N-1}\boldsymbol{z}_n \end{bmatrix}, \tag{3.8}$$

run Algorithm 3.1 for each $\widetilde{\boldsymbol{R}}^{[p]}$ and observe the returned errors $\boldsymbol{e}^{[p]} = [e_1^{[p]}, e_2^{[p]}, \ldots, e_L^{[p]}]$. We compare them to $\boldsymbol{e} = [e_1, e_2, \ldots, e_L]$, the errors computed by Algorithm 3.1 for the ground truth trajectory $\boldsymbol{R}$. We apply the MUSIC algorithm to find the K smallest minima in $\boldsymbol{e}^{[p]}$ and $\boldsymbol{e}$. Our experiments with real RIRs are illustrated in Fig. 3.6; they show the following:

OBSERVATION 3.3 The positions of the K smallest local minima of $\boldsymbol{e}^{[p]}$ are close to the K smallest local minima of $\boldsymbol{e}$ for reasonable amounts of noise σ_v.[2] In other words, Algorithm 3.1 finds the correct first-order echoes in RIRs when the generated trajectories are not too far from the ground truth trajectory.

When we increase the amount of noise added to the trajectory, we achieve two opposite effects on the errors returned by Algorithm 3.1. In the first case, the noisy trajectory is such that no rooms match the geometry implied by the measured propagation times:

OBSERVATION 3.4 With high σ_v we generate realizations of the trajectory for which the propagation times of the first-order echoes are far from the measured propagation times, so the error is large for every potential wall $\bar{\varphi}_\ell$.

In the second case, we are less lucky; the noisy trajectory is such that some second-order echoes of the ground truth trajectory are close to the first-order echoes of the noisy trajectory:

OBSERVATION 3.5 With high σ_v we generate realizations of the trajectory for which the first-order echoes from some "ghost" wall are close to the measured propagation times of the second-order echoes. Even though this case occurs rarely, the errors returned by Algorithm 3.1 for such trajectories have small local minima; they are of the same order as the errors computed for the ground truth trajectory.

With the intuition gained from the experiments, we propose to extend Algorithm 3.1 and improve its accuracy for noisy trajectories by performing the following steps:

1. For a given noisy estimate $\widetilde{\boldsymbol{R}}$, we generate P realizations $\widetilde{\boldsymbol{R}}^{[p]}$ by (3.8), where we replace $\boldsymbol{R}$ with $\widetilde{\boldsymbol{R}}$.
2. For every realization $\widetilde{\boldsymbol{R}}^{[p]}$, we run Algorithm 3.1 and obtain the errors $\boldsymbol{e}^{[p]} = [e_1^{[p]}, \ldots, e_L^{[p]}]$.
3. We select aP trajectories with the smallest $\|\boldsymbol{e}^{[p]}\|$, where $1/P < a_{min} < a < a_{max} < 1$, and we average them.
4. The locations of the K smallest local minima of the averaged error reveal the wall normals, $\widehat{\boldsymbol{\varphi}} = [\bar{\varphi}_{\ell_1}, \ldots, \bar{\varphi}_{\ell_K}]^\top$.

[2] The observation is made for $\sigma_v < 4$ cm. Knowing that mobile robots can provide much better precision, this is a rather high amount of noise for many measurement methods. For instance, a real robot used in the experimental part of Chapter 4 achieves $\sigma_v < 0.5$ cm.

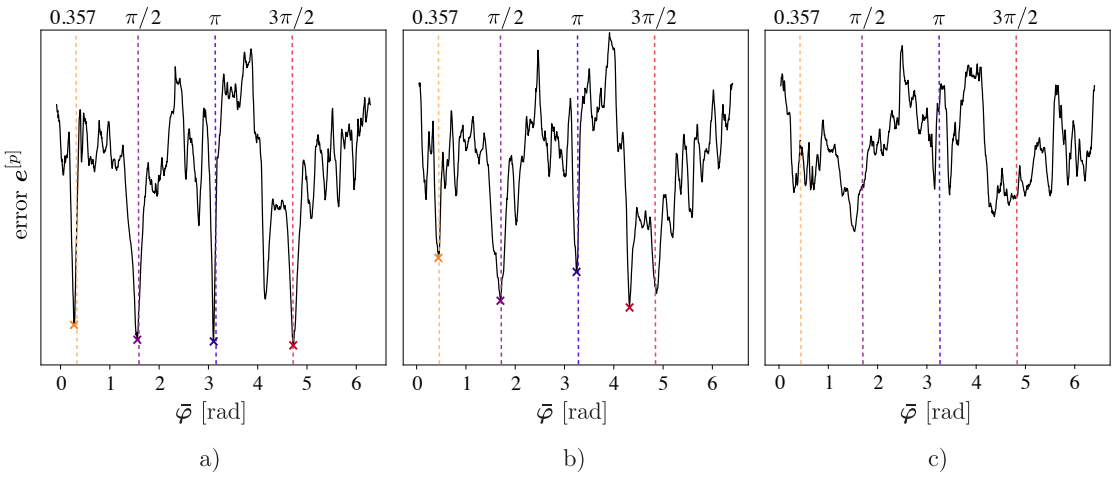


Figure 3.6: Errors computed by Algorithm 3.1 for three different noise levels. The ground truth angles are indicated with dashed lines and their values are written above the graphs. The angles for which the error is among the four smallest local minima are marked with $\times$. a) $\sigma_v = 1.7$ cm. This case is described in Observation 3.3. The four smallest local minima are very close to the ground truth values. b) $\sigma_v = 5.1$ cm. This case is described in Observation 3.5. We can see that the smallest local minimum is not close to the ground truth values; it corresponds to the ghost wall at $\bar{\varphi} \approx 4.2$ rad. c) $\sigma_v = 7.5$ cm. This case is described in Observation 3.4. The error is large for every possible wall normal $\bar{\varphi}_\ell$.

We propose to pick $a_{max} \approx 0.5$ as we want to compute the average of the errors that have small local minima and our Observation 3.3 confirms that this is the case for the trajectories that are close to the ground truth trajectory. Observation 3.5 however shows that there are trajectories that can be far from the ground truth trajectory but have a small value of $\|\boldsymbol{e}^{[p]}\|$; this indicates the presence of ghost walls. To generate one such trajectory by (3.8) is not very likely. Hence, even though they might have small $\|\boldsymbol{e}^{[p]}\|$, they are rare. To average out the ghost walls, we propose to set $a_{min} \approx 0.2$.

Our experiments show that the K smallest local minima of the averaged errors $\bar{e}_\ell = \frac{1}{aP}\sum e_\ell^{[p]}$ are indeed close to the correct K walls in a room, where the terms in the sum correspond to aP trajectories with the smallest $\|\boldsymbol{e}^{[p]}\|$ and $a = 0.2$. More details about our experimental setup and results are provided in Section 3.6.3.

Now that we have estimates of the angles $\widehat{\boldsymbol{\varphi}}$ of the wall normals, we can find the corresponding offsets $\widehat{\boldsymbol{q}}$ from (3.6). For the computed $\bar{\varphi}_\ell = \widehat{\varphi}_k$ and $\hat{q}_k \in \widehat{\boldsymbol{q}}$, we construct a PPDM $\widetilde{\boldsymbol{D}}$ as in Section 3.3.3.1 and provide it as an input to the next stage of our two-step approach.

3.4 From PPDMS to rooms and trajectories

In this section, we turn our focus to Problem 3.2 and assume that we are given a PPDM $\widetilde{\boldsymbol{D}}$. In the noiseless case, the room-trajectory configuration can be recovered from only a few distance measurements by solving polynomial equations (2.2). In practice however, noisy entries are common and solving polynomial equations might be problematic. Additionally, the algebraic approach makes it difficult to incorporate any prior knowledge we might have about the room

or the trajectory. It is easy to imagine scenarios where some information from inertial sensors is available, and the algebraic approach provides no simple way to integrate it.

3.4.1 Optimization formulation

To address these shortcomings, we formulate the joint recovery as an optimization problem and we aim to estimate a room-trajectory configuration that best fits given measured distances. More precisely, we describe our problem as a low-rank matrix factorization: given noisy and incomplete PPDM $\widetilde{\boldsymbol{D}}$, we want to jointly recover the waypoints $\widehat{\boldsymbol{R}}$, the wall normals in a column-unitary matrix $\widehat{\boldsymbol{N}}$ and the distances of the walls from the origin $\widehat{\boldsymbol{q}}$, such that

$$\widehat{\boldsymbol{R}}, \widehat{\boldsymbol{N}}, \widehat{\boldsymbol{q}} = \underset{\boldsymbol{R},\boldsymbol{N},\boldsymbol{q}}{\arg\min} \left\| \widetilde{\boldsymbol{D}} - \boldsymbol{W} \circ \left(\mathbf{1}\boldsymbol{q}^\top - \boldsymbol{R}^\top \boldsymbol{N} \right) \right\|^2. \tag{3.9}$$

A binary mask $\boldsymbol{W} = [w_{nk}]_{n,k=1}^{N,K} \in \mathbb{R}^{N \times K}$ indicates the known entries of a PPDM by 1, so the noisy and incomplete $\widetilde{\boldsymbol{D}}$ is defined as $\widetilde{\boldsymbol{D}} = \boldsymbol{W} \circ (\boldsymbol{D} + \boldsymbol{Z})$, where the entries of $\boldsymbol{Z} \in \mathbb{R}^{N \times K}$ are independent noise realizations and $\circ$ is the Hadamard product.

Even thought the algorithm from the previous section always provides complete PPDMs $\widetilde{\boldsymbol{D}}$, we do not want to restrict a solution to Problem 3.2 to work with complete matrices only. As a matter of fact, we study an identical optimization formulation (3.9) in our previous work [90], which arose from a distinct problem where the missing measurements are common—the problem of localizing a set of microphones together with a set of external acoustic events, emitted at unknown times and unknown locations. Therefore, to broaden the range of PPDM applications, we develop a general solution to PPDM factorization in the presence of missing entries and noise.

Similarly, the algorithm from the previous section does not only compute PPDMs, but it also provides estimates of walls for a given estimate of a trajectory. To keep the solution to Problem 3.2 general, we develop an iterative algorithm that can factorize PPDMs in both cases of known and unknown initial estimates of the walls and trajectories. When the estimates are known, we can leverage that information by initializing the algorithm with $\widehat{\boldsymbol{N}}$ and $\widetilde{\boldsymbol{R}}$; otherwise, we initialize the matrices randomly.

Problem 3.2 can be abstracted as the recovery of the point coordinates from a subset of their orthogonal projections on unknown lines or planes. A particular instance of this abstraction is when the planes pass through the origin; this is equivalent to setting $\boldsymbol{q} = \mathbf{0}$ in (3.1):

$$\boldsymbol{D} = -\boldsymbol{R}^\top \boldsymbol{N}. \tag{3.10}$$

It has been shown that (3.1) can be reduced to (3.10) at the expense of losing one measurement [101, 167]. Indeed, one can easily eliminate $\mathbf{1}\boldsymbol{q}^\top$ by choosing an arbitrary waypoint $\boldsymbol{r}_m$ to be the origin, $\boldsymbol{r}_m = \mathbf{0}$, and by subtracting the mth row of $\widetilde{\boldsymbol{D}}$ from all rows in $\widetilde{\boldsymbol{D}}$:

$$d_{mk} = \operatorname{dist}(\boldsymbol{r}_m, \mathcal{P}_k) = q_k - \boldsymbol{r}_m^\top \boldsymbol{n}_k = q_k.$$

Then,

$$\boldsymbol{D} - \mathbf{1}\boldsymbol{q}^\top = -\boldsymbol{R}^\top \boldsymbol{N} = \begin{bmatrix} d_{11} - d_{m1} & d_{12} - d_{m2} & \dots & d_{1K} - dmK \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} - d_{m1} & d_{N2} - d_{m2} & \dots & d_{NK} - dmK \end{bmatrix}, \tag{3.11}$$

where the mth column of $\boldsymbol{R}$ is $\mathbf{0}$. Therefore, after fixing $\boldsymbol{r}_m = \mathbf{0}$ and the substraction (3.11), we can remove a zero-row from $\boldsymbol{D} - \mathbf{1}\boldsymbol{q}^\top$ and factorize it to jointly recover the remaining $N-1$ waypoints in $\boldsymbol{R}$ and the K wall normals in $\boldsymbol{N}$. There are a number of techniques for solving (3.10). One way is to compute the singular value decomposition (SVD) $\widetilde{\boldsymbol{D}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ and perform the gradient descent algorithm to find the matrix $\boldsymbol{C} \in \mathbb{R}^{D\times D}$ such that $\widehat{\boldsymbol{R}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{C}^{-1}$ and $\widehat{\boldsymbol{N}} = \boldsymbol{C}\boldsymbol{V}^\top$, where D denotes the dimension of the space [167]; this ensures that the columns of $\widehat{\boldsymbol{N}}$ are unit-norm. The main limitation of SVD-based methods is that they require the complete measurement matrix $\widetilde{\boldsymbol{D}}$.

When $\widetilde{\boldsymbol{D}}$ is incomplete, but contains at least one complete row, the algorithm proposed by Kuang et al. [101] can be used to jointly estimate $\widehat{\boldsymbol{R}}$ and $\widehat{\boldsymbol{N}}$. The challenge of missing data becomes apparent when the distances of walls from the origin $\hat{\boldsymbol{q}}$ have to be estimated jointly with $\widehat{\boldsymbol{R}}$ and $\widehat{\boldsymbol{N}}$, which is the case when no rows of $\widetilde{\boldsymbol{D}}$ contain all elements. As the existing solutions cannot deal with that case, we provide an algorithm to resolve it.

In our previous work [90], we studied such an optimization problem and proposed to consider pairwise differences between the rows of $\boldsymbol{D}$ to eliminate the dependence of measurements on $\boldsymbol{q}$. We construct the measurement tensor $\boldsymbol{S} \in \mathbb{R}^{N\times N\times K}$ with relative distances $s_{nmk} = d_{nk} - d_{mk}$. When the measurements $\widetilde{\boldsymbol{D}}$ are incomplete and corrupted with noise, we obtain an incomplete and noisy tensor as well, denoted $\widetilde{\boldsymbol{S}}$, where $\widetilde{s}_{nmk} = \nu_{nmk}(\widetilde{d}_{nk} - \widetilde{d}_{mk})$. The element ν_{nmk} is equal to 1 when both distances $\widetilde{d}_{nk}$ and $\widetilde{d}_{mk}$ are measured, and 0 otherwise. It can be expressed as the product of the two corresponding elements of the mask $\boldsymbol{W}$, $\nu_{nmk} = w_{nk}w_{mk}$.

By observing the pairwise differences $\widetilde{\boldsymbol{S}}$, we eliminate $\boldsymbol{q}$ and reformulate (3.9) as

$$\begin{aligned}\widehat{\boldsymbol{R}}, \widehat{\boldsymbol{N}} = \underset{\boldsymbol{R},\boldsymbol{N}}{\arg\min} \sum_{k=1}^{K} \sum_{n,m=1}^{N} \left(\widetilde{s}_{nmk} - \nu_{nmk}\boldsymbol{n}_k^\top \boldsymbol{\rho}_{nm}\right)^2 \\ \text{s.t. } \|\boldsymbol{n}_k\|^2 = 1,\end{aligned} \tag{3.12}$$

where $\boldsymbol{\rho}_{nm} = \boldsymbol{r}_n - \boldsymbol{r}_m$.

3.4.2 Proposed method in 2D

For simplicity, we first analyze the 2D case; the generalization to 3D is detailed in Section 3.4.3. The cost function defined in (3.12) is non-convex with many local minima. We can however find the global minimizer of (3.12) over $\boldsymbol{N}$ for fixed $\boldsymbol{R}$, and analogously, the optimal solution of $\boldsymbol{R}$ for fixed $\boldsymbol{N}$. Therefore, we propose to alternate between estimates $\widehat{\boldsymbol{R}}$ and $\widehat{\boldsymbol{N}}$, where at each step $\widehat{\boldsymbol{R}}$ and $\widehat{\boldsymbol{N}}$ are optimized separately keeping the other one fixed.

Recall that the matrix $\boldsymbol{N}$ in 2D, defined in (2.11), is uniquely determined by the angles $\{\varphi_k\}_{k=1}^K$, and vice versa, from $\boldsymbol{N}$ we can unambiguously compute $\{\varphi_k\}_{k=1}^K$. To derive the optimal $\widehat{\boldsymbol{R}}$ for fixed $\widehat{\boldsymbol{N}}$, we take the first derivative of the cost function in (3.12) with respect to x and y coordinate of $\boldsymbol{r}_n$, and set the resulting equations to 0. We use $\boldsymbol{\chi}$ to represent the flattened matrix $\widehat{\boldsymbol{R}}$, $\boldsymbol{\chi} = [\hat{r}_{1,x}, \ldots, \hat{r}_{N,x}, \hat{r}_{1,y}, \ldots, \hat{r}_{N,y}]$. The system of equations is then written as

$$\begin{bmatrix}\boldsymbol{M}^{xx} & \boldsymbol{M}^{xy} \\ \boldsymbol{M}^{xy} & \boldsymbol{M}^{yy}\end{bmatrix}\boldsymbol{\chi} = \begin{bmatrix}\boldsymbol{p}^x \\ \boldsymbol{p}^y\end{bmatrix} \text{ or } \boldsymbol{M}\boldsymbol{\chi} = \boldsymbol{p}, \tag{3.13}$$

Algorithm 3.2 Proposed room reconstruction algorithm.

Input: Incomplete noisy measurement matrix $\widetilde{\boldsymbol{D}}$, mask matrix $\boldsymbol{W}$, convergence criterion ϵ. Optional: initial matrices $\boldsymbol{R}^0$ and $\boldsymbol{\varphi}^0$.
Output: $\widehat{\boldsymbol{R}}$ and $\widehat{\boldsymbol{N}}$ that minimize (3.12).
Let $\nu_{nmk} = w_{nk}w_{mk}$.
Let $\widetilde{s}_{nmk} = (\widetilde{d}_{nk} - \widetilde{d}_{mk})\nu_{nmk}$.
If $\boldsymbol{R}^0$ and $\boldsymbol{\varphi}^0$ not provided, randomly initialize $\boldsymbol{R}^0$ and $\boldsymbol{\varphi}^0$.
Repeat for each iteration i until $\|\widehat{\boldsymbol{S}}^i - \widehat{\boldsymbol{S}}^{i-1}\|_F^2 < \epsilon$:
Step 1. For fixed $\widehat{\boldsymbol{\varphi}}^{i-1}$, find $\widehat{\boldsymbol{R}}^i$ by solving the linear system (3.13).
Step 2. For fixed $\widehat{\boldsymbol{R}}^i$, find $\widehat{\boldsymbol{\varphi}}^i$ by solving (3.14).
Step 3. $\hat{s}^i_{nmk} = \nu_{nmk}\big(\hat{\rho}^i_{nm,x}\cos\hat{\varphi}^i_k + \hat{\rho}^i_{nm,y}\sin\hat{\varphi}^i_k\big)$.
return $\widehat{\boldsymbol{R}}$, $\widehat{\boldsymbol{\varphi}}$

where the entries of $\boldsymbol{M}^{**} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{p}^* \in \mathbb{R}^N$ are given as:

$$M^{xx}_{nm} = \begin{cases} \sum_{k=1}^K \sum_{m=1}^N \nu_{nmk}\cos^2\varphi_k & \text{if } n = m \\ -\sum_{k=1}^K \nu_{nmk}\cos^2\varphi_k, & \text{otherwise} \end{cases},$$

$$M^{yy}_{nm} = \begin{cases} \sum_{k=1}^K \sum_{m=1}^N \nu_{nmk}\sin^2\varphi_k & \text{if } n = m \\ -\sum_{k=1}^K \nu_{nmk}\sin^2\varphi_k & \text{otherwise} \end{cases},$$

$$M^{xy}_{nm} = M^{yx}_{mn} = \begin{cases} \sum_{k=1}^K \sum_{m=1}^N \nu_{nmk}\cos\varphi_k\sin\varphi_k & \text{if } n = m \\ -\sum_{k=1}^K \nu_{nmk}\cos\varphi_k\sin\varphi_k & \text{otherwise} \end{cases},$$

$$p^x_n = \sum_{k=1}^K \sum_{m=1}^N \widetilde{s}_{nmk}\nu_{nmk}\cos\varphi_k,$$

$$p^y_n = \sum_{k=1}^K \sum_{m=1}^N \widetilde{s}_{nmk}\nu_{nmk}\sin\varphi_k.$$

Therefore, by solving the linear system in (3.13), we can recover $\widehat{\boldsymbol{R}}$.

The second step of the algorithm estimates $\widehat{\boldsymbol{N}}$ for fixed $\widehat{\boldsymbol{R}}$ by following the same idea as in the first step. The problem decouples over k, so we can solve it for every k separately. We set the first derivatives of the cost function in (3.12) with respect to φ_k to 0:

$$\begin{aligned} &\sin\varphi_k \sum_{n,m=1}^N \widetilde{s}_{nmk}\rho_{nm,x} - \frac{1}{2}\sin 2\varphi_k \sum_{n,m=1}^N \nu_{nmk}(\rho^2_{nm,x} - \rho^2_{nm,y}) \\ &-\cos\varphi_k \sum_{n,m=1}^N \widetilde{s}_{nmk}\ \rho_{nm,y} + \cos 2\varphi_k \sum_{n,m=1}^N \nu_{nmk}\rho_{nm,x}\rho_{nm,y} = 0, \end{aligned} \tag{3.14}$$

where $\boldsymbol{\rho}_{nm} = [\rho_{nm,x}, \rho_{nm,y}]^\top$. By replacing trigonometric functions with complex exponentials and substituting $b_k = e^{j\varphi_k}$, we obtain quartic polynomials in b_k. To find the global solution, we evaluate the cost function for all four roots and choose b_k that yields the smallest value.

We iteratively alternate between these two steps and refine our estimates of $\widehat{\boldsymbol{N}}$ and $\widehat{\boldsymbol{R}}$. This procedure is summarized in Algorithm 3.2. We use superscript i to denote the values of the

estimates at iteration i. The algorithm stops when $\|\widehat{\boldsymbol{S}}^{i} - \widehat{\boldsymbol{S}}^{i-1}\|_F^2$ falls below some threshold ϵ. After having recovered $\widehat{\boldsymbol{R}}$ and $\widehat{\boldsymbol{N}}$, we estimate $\widehat{\boldsymbol{q}}$ by solving

$$\widehat{\boldsymbol{q}} = \arg\min_{\boldsymbol{q}} \left\| \widetilde{\boldsymbol{D}} - \boldsymbol{W} \circ \left(\mathbf{1}\boldsymbol{q}^\top - \widehat{\boldsymbol{R}}^\top \widehat{\boldsymbol{N}} \right) \right\|^2,$$

using the weighted least square method.

The proposed algorithm is an instance of more general alternating optimization problems, where the variables are partitioned in two subsets and the optimal solution is estimated for each subset separately at every iteration. Such algorithms are ubiquitous and their properties are well-studied [15, 67]. It was shown that this class of algorithms is locally, Q-linearly (i.e., quotient-linearly) convergent to some value x^*, which means that there exists $r \in (0, 1)$ such that $\frac{|x_{k+1}-x^*|}{|x_k-x^*|} \leq r$ for all k sufficiently large [16]. Note that this does not imply a global convergence of the algorithm.

3.4.3 Generalization to 3D

The generalization to 3D is straightforward; the most significant modification takes place in step 2 of Algorithm 3.2. To ensure that the columns of $\boldsymbol{N}$ have unit norms, we represent them in spherical coordinates as in (2.12).

Step 1 of Algorithm 3.2 is adjusted by fixing both vectors $\boldsymbol{\varphi} = [\varphi_1, \ldots, \varphi_K]$ and $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_K]$. Then, the minimization over $\boldsymbol{R}$ is formalized as in (3.13), with $\boldsymbol{M}$ having an additional column and row. Their values can be computed analogously to the 2D case by replacing the polar coordinates in $\boldsymbol{N}$ with the spherical coordinates. Step 2 however becomes slightly more complicated, as we need to take derivatives over two angles, φ_k and θ_k. We perform a similar substitution as in the 2D case, but instead of obtaining one quatric equation, we get two bivariate polynomials of degree 8. To find the roots, we use the Broyden's first Jacobian approximation [172].

3.5 Numerical simulations and the performance evaluation

In this section, we numerically evaluate and quantify the performance of Algorithm 3.2 by using the root mean squared error (RMSE) with respect to: 1) the noiseless measurement matrix $\boldsymbol{D}$, 2) the ground truth waypoints $\boldsymbol{R}$, and 3) the ground truth corners of a room. The RMSE($\boldsymbol{D}$) quantifies the consistency of the algorithms and it is computed as:

$$\text{RMSE}(\boldsymbol{D}) = \sqrt{\frac{\sum_{n=1}^{N} \sum_{k=1}^{K} \left(\widehat{d}_{nk} - d_{nk} \right)^2}{NK}}.$$

The RMSE of the waypoints and the corners of a room determine the capacity to accurately estimate the trajectories and room geometries, which is typically the goal. After the reconstruction algorithm, we first perform a Procrustes transformation to align the recovered points with the ground truth, and then find the error as:

$$\text{RMSE}(\boldsymbol{R}) = \sqrt{\frac{\sum_{n=1}^{N} \|\widehat{\boldsymbol{r}}_n - \boldsymbol{r}_n\|^2}{N}}.$$

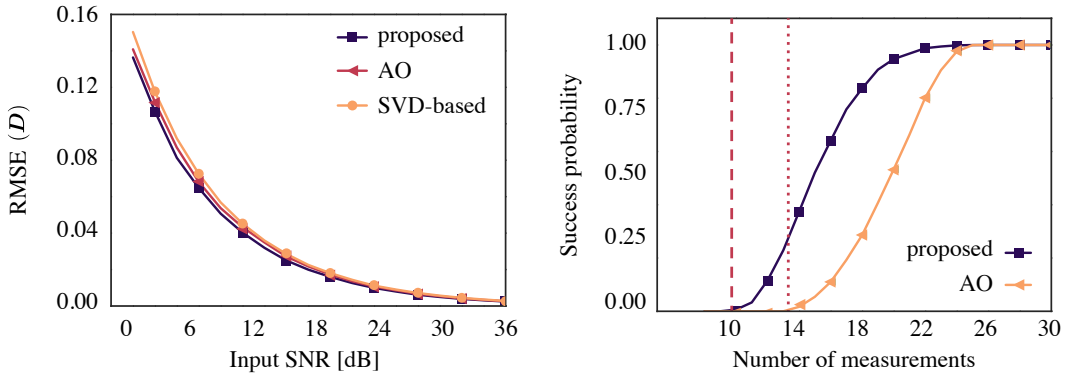


Figure 3.7: a) Comparison of the proposed method with the SVD-based approach and the alternating optimization for different values of input SNR and a complete PPDM. b) Likelihood of the algorithms to work for an randomly created mask $\boldsymbol{W}$ with a given number of measurements. Each experiment is an average of 1000 realizations.

Our experiments are conducted for three different configurations of missing entries in $\boldsymbol{D}$: 1) $\boldsymbol{D}$ is complete, 2) there are missing entries, but at least one row of $\boldsymbol{D}$ is complete, 3) there are missing entries and no complete rows.

In Section 3.5.1 and Section 3.5.2 we examine the performance of our algorithm against the most relevant existing methods for the first two cases. We perform numerical simulations on randomly generated setups: the matrix $\boldsymbol{N}$ can be any projection matrix with unit-norm columns, while $\boldsymbol{R}$ and $\boldsymbol{q}$ are unconstrained. The goal of Section 3.5.3 is to apply the proposed algorithm to the joint estimation of rooms and trajectories, and illustrate the accuracy of our algorithm for different noise levels.

3.5.1 Complete matrices

We first turn to the complete case and compare our algorithm with the SVD-based estimator proposed by Thrun [167], and the alternating optimization (AO) algorithm by Kuang et al. [101]. The SVD-based algorithm only handles this case, while the AO method can work in the presence of missing entries but with at least one complete row.

As mentioned in Section 3.4.2, the proposed algorithm, as well as the two above-mentioned solutions, does not always converge to a global minimum. Besides, the estimates obtained by AO and the proposed algorithm depend on the initial values of $\boldsymbol{R}$ and $\boldsymbol{\varphi}$. As the convergence of both methods is within a few steps (in the order of milliseconds for about 30 measurements), we can afford to repeat the estimation several times with different initializations to avoid local minima. In the experiment below, we run the iterative algorithms ten times and keep the solution with the lowest cost. We empirically observed that AO has a stronger tendency than our algorithm to get trapped in local minima.

We assume to have $N = 6$ points and $K = 5$ lines in 2D. The measurement matrix $\boldsymbol{D}$ is corrupted with Gaussian noise such that the input signal-to-noise ratio (SNR) ranges from 0 to 36 dB. The dependency of the RMSE($\boldsymbol{D}$) on the input SNR is shown in Fig. 3.7a. Our algorithm outperforms both the SVD-based approach and AO, even though the SVD-based approach was

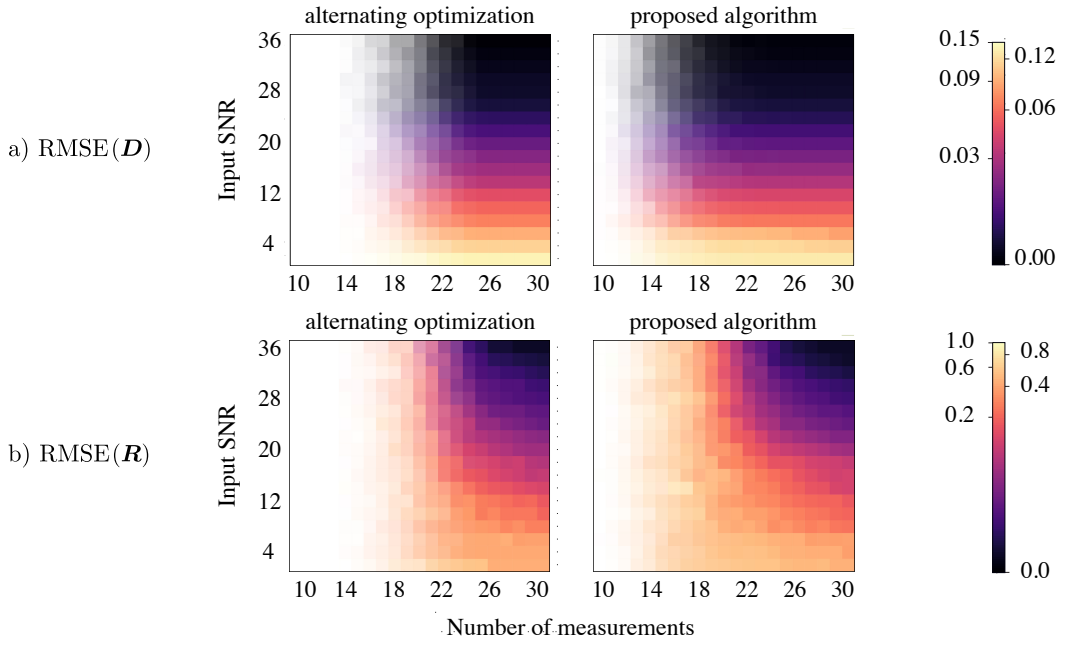


Figure 3.8: Performance of the proposed method and the alternating optimization for different input SNR and number of missing entries. Reported errors are averaged over 300 realizations. The proposed algorithm achieves lower RMSE($\boldsymbol{D}$) and RMSE($\boldsymbol{R}$) for all combinations of input SNR and missing entries.

tailored exclusively for complete matrices.

3.5.2 Missing entries

Number of entries. Next, we compare the algorithms when some entries are missing. For a successful reconstruction, the arrangement of the measurements in $\widetilde{\boldsymbol{D}}$ has to satisfy several conditions. As previously mentioned, AO needs a complete row, while our difference-based algorithm does not work with less than two entries per column. Moreover, both algorithms require a *connected* $\widetilde{\boldsymbol{D}}$. We say that a matrix is connected when there exists a path between any two entries $\widetilde{d}_{nk}$ and $\widetilde{d}_{ml}$, such that every entry in the path shares a common column or a row with the previous entry. More precisely, AO requires a connected matrix obtained from $\widetilde{\boldsymbol{D}}$ after removing its complete row.

For a given number of measurements M, we repeatedly create random masks $\boldsymbol{W}$ with M non-zero entries and verify whether the matrices $\widetilde{\boldsymbol{D}}$ satisfy the above conditions. This gives us an estimate of the likelihood for the algorithms to work with M entries, as illustrated in Fig. 3.7b. In addition, we observe that the smallest possible M for our algorithm is $2K$ when $N \leq K+1$ and $K+N-1$ otherwise, or equivalently, at least two entries per column. Similarly, the threshold for AO is $2K+N-2$, since with less measurements it is impossible to construct a connected matrix after removing a complete row from $\widetilde{\boldsymbol{D}}$. These thresholds separate success from failure modes for our algorithm and AO, as indicated in Fig. 3.7b.

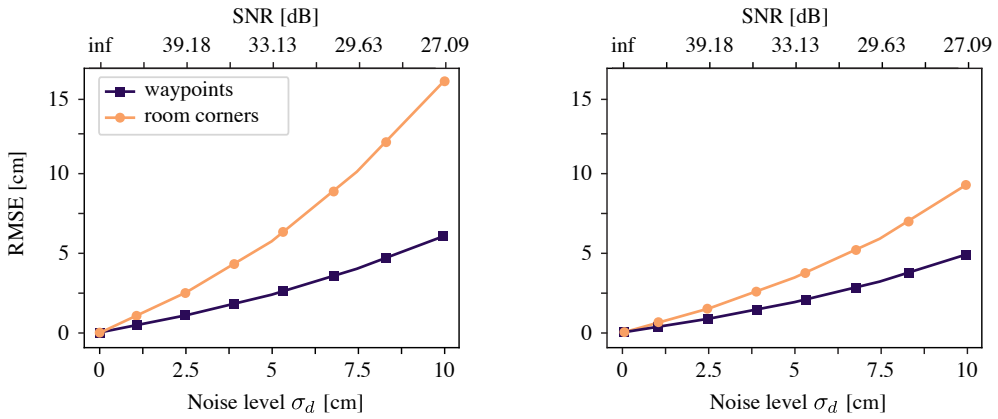


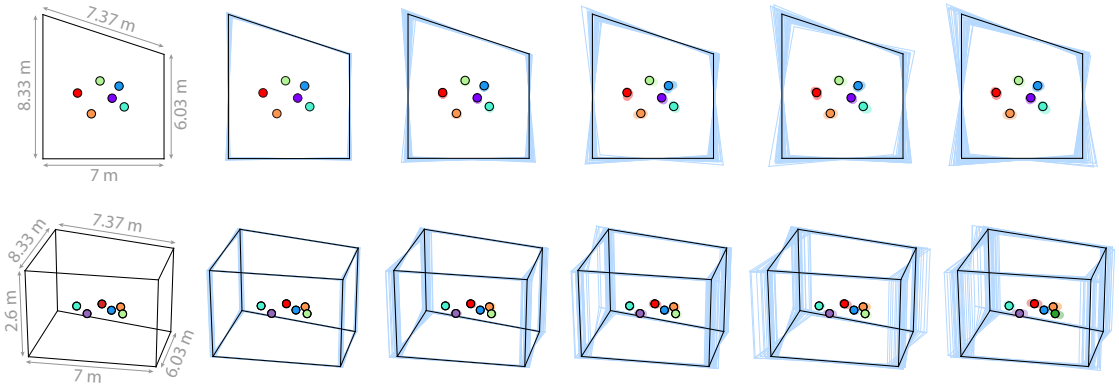
Figure 3.9: Dependence of the reconstruction errors on the noise level σ_d added to the entries of a PPDM. The configurations consist of $N = 6$ waypoints and $K = 4$ or $K = 6$ walls in 2D and 3D, respectively. For each σ_d, we ran 5000 experiments.

Performance comparison. In Fig. 3.8, we compare the performance of the two algorithms for varying levels of noise and number of measurements. Again, our method achieves lower RMSE($\boldsymbol{D}$) and RMSE($\boldsymbol{R}$) for all combinations of input SNR and missing entries. Fig. 3.8a also suggests that the reconstruction is stable and the RMSE($\boldsymbol{D}$) increases with the amount of noise in the same manner as in Fig. 3.7. Interestingly, we notice that the RMSE($\boldsymbol{D}$) values are not affected by the number of missing entries; indeed, they remain fairly constant along the x-axis. This is not the case for the RMSE($\boldsymbol{R}$), which grows significantly as the number of measurements decreases. In other words, even though it is always possible to find a low-rank matrix $\widehat{\boldsymbol{D}}$ that is consistent with the observed measurements, it is not guaranteed that its factorization is close in any way to the original point configuration; this is especially true as $\widetilde{\boldsymbol{D}}$ becomes sparser. Moreover, we observe that both algorithms can sometimes yield an estimate that is far from the original points, causing outliers in RMSE($\boldsymbol{R}$). In Fig. 3.8b we discarded the experiments resulting in an RMSE($\boldsymbol{R}$) larger than a certain threshold value ζ.

The success of the reconstruction is depicted by transparency, where a high failure rate translates to a more transparent value. In Fig. 3.8a, the reconstruction is considered successful when $\widetilde{\boldsymbol{D}}$ is connected, as explained earlier and depicted in Fig. 3.7b. To assess the success in Fig. 3.8b, we also take into account the percentage of outliers (errors larger than ζ). We observe that our approach surpasses AO with respect to RMSE($\boldsymbol{D}$) and RMSE($\boldsymbol{R}$) both in performance and the amount of missing entries.

3.5.3 Room reconstruction and trajectory estimation

To evaluate and illustrate the effect of noise on the success of the room reconstruction, we performed a number of numerical simulations in 2D and 3D. We assume $N = 6$ waypoints and $K = 4$ walls in 2D, and $N = 6$ waypoints and $K = 6$ walls in 3D. To increase the probability that K walls enclose a convex polyhedral room, instead of generating arbitrary planes, we set-up the experiment in the following way: we construct a rectangle in 2D and a rectangular prism



a) $\sigma_d = 0$ b) $\sigma_d = 0.01$ c) $\sigma_d = 0.02$ d) $\sigma_d = 0.03$ e) $\sigma_d = 0.04$ f) $\sigma_d = 0.05$

Figure 3.10: Several realizations of reconstructed rooms and waypoints for different levels of noise σ_d in meters. The walls of the original room are depicted in black, and the ground truth waypoints have black strokes. The walls of the reconstructed rooms are transparent blue, while the reconstructed waypoints are transparent. *Top*: A 2D trapezoidal room. *Bottom*: A 3D room with the trapezoid floor plan (same dimensions as in 2D), and the walls perpendicular to the floor and the ceiling.

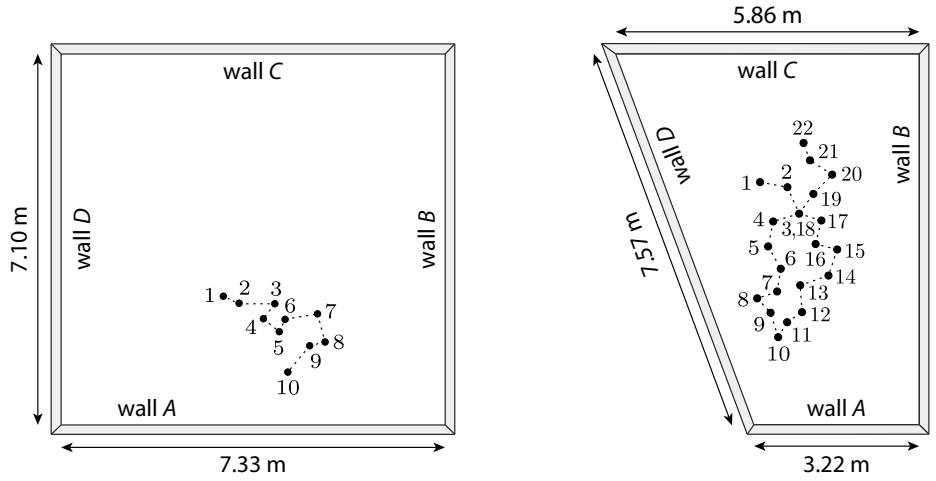
(shoebox) in 3D, whose sizes reflect realistic environments (the area of a room is about 30 m^2, and the height is 2.6 m). We generate the waypoints inside the room uniformly at random. Then, we perturb the wall normals and the locations of the waypoints to randomize each experiment while preserving the requirement that one can hear all first-order echoes at every waypoint, i.e., that the room is convex.

We compute the PPDM of every generated room-trajectory configuration and corrupt its elements with i.i.d. Gaussian noise $\mathcal{N}(0, \sigma_d^2)$. In Fig. 3.9 we show the dependence of the algorithm's performance on σ_d, where σ_d ranges from 0 to 10 cm.

In Fig. 3.10, we visualize reconstruction errors for fixed room-trajectory configurations in 2D and 3D. To replicate the environment in which we conduct real experiments, we assume that the walls are perpendicular to the floor and the ceiling. Moreover, the floor plan has a shape of a trapezoid identical to the 2D case, and the height of the room is 2.6 m. This room belongs to a class of rooms found in Section 2.5.5, so it is not uniquely specified by the distance measurements. To solve the ambiguity, we consider that all the waypoints have the same z-coordinate, that is, the height of the device does not change along the trajectory. Fig. 3.10 illustrates several realizations of reconstructed rooms and waypoints for different noise levels σ_d. As a reference, we also indicate dimensions of the room and plot the original (noiseless) setup.

3.6 Real experiments

We validate Algorithm 3.1 and Algorithm 3.2 experimentally in a classroom on our campus. In the following, we introduce the setup, describe the acquisition of ground truth data and demonstrate the performance of our algorithms on real room impulse responses.



a) Rectangular room b) Trapezoidal room

Figure 3.11: Illustration of the rooms and trajectories from the real experiments. Numbers $n = 1, 2, 3, \ldots$ inside each room index the order of the waypoints $\boldsymbol{r}_n$.

3.6.1 Setup

We conducted experiments in two rooms, depicted in Fig. 3.11a and Fig. 3.11b. Room 1 was rectangular. The wall denoted by A is a drywall, the wall C is covered with a whiteboard, while the walls B and D are made of glass. Room 2 was trapezoidal: we built a wall across room 1 made of classroom desks to replace wall D.

In both cases, we used a small cylindrical wireless speaker Veho 360 for room response measurements. Its height is 6.5 cm and a diameter is 5.5 cm. The room responses were recorded with an omnidirectional microphone located just above the loudspeaker. Such a collocated setup was placed roughly at half of the room height, so that the first order echoes from the floor and the ceiling overlapped. In order to measure high quality RIRs, we calibrated the setup in an anechoic chamber by using the same (nearly) collocated setup and by playing the same exponential sine sweep as in the room. This response was then used to recover the room impulse responses by deconvolving it from the signals recorded in the room. We applied the deconvolution algorithm developed by Scheibler et al. [151].

The Veho 360 has dual drivers facing opposite directions. To simulate a desired omnidirectional behaviour, at each waypoint we measured the room response twice: we played an exponential sine sweep 10 s long with frequencies ranging from 1 Hz to 24 kHz and a sampling frequency of $f_s = 48$ kHz, and repeated the measurement at the same location after rotating the loudspeaker by 90°. We summed up the recordings from both orientations in order to discard the directivity information.

3.6.2 Ground truth measurement

To obtain reliable ground truth data, we placed two reference points on each wall and used a laser distance meter to measure pairwise distances between all waypoints $\boldsymbol{R}$ and eight reference points $\boldsymbol{g}_i \in \mathbb{R}^2$, $\boldsymbol{G} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_8\}$. We also measured distances between the reference points

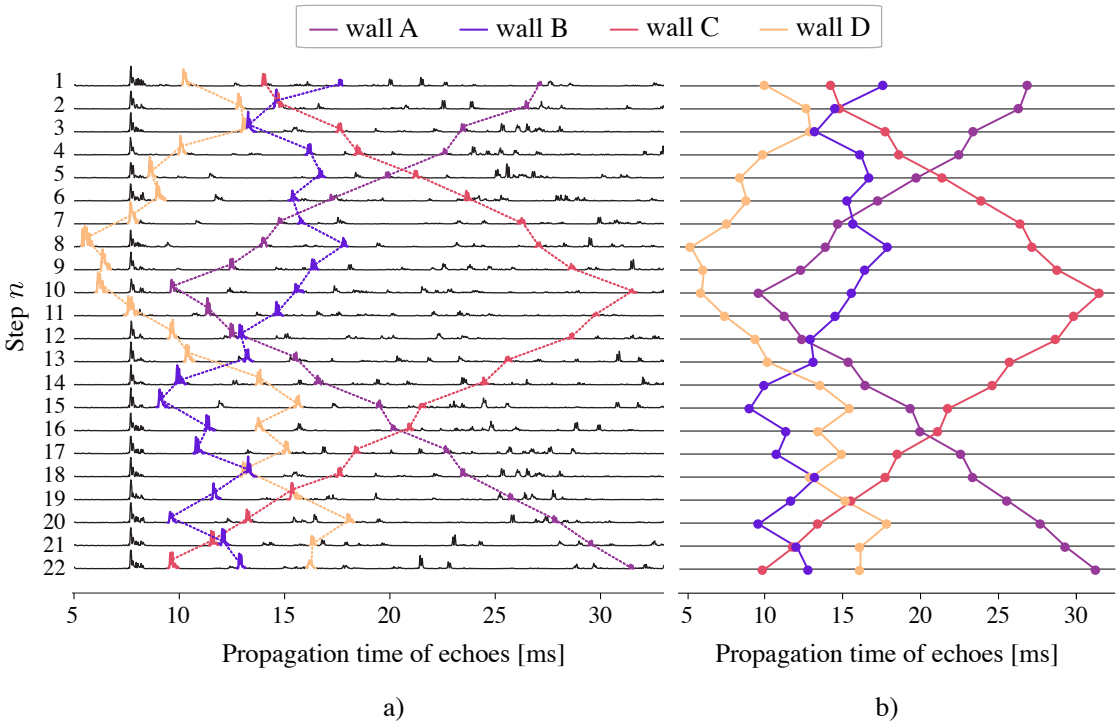


Figure 3.12: Measured RIRs. Information about the propagation times of the first-order echoes is present and accurate in the recorded RIRs, but it is available only after resolving the echo detection and echo labeling problems. a) Recorded RIRs at $N = 22$ locations in the trapezoidal room. b) Expected propagation times of the first-order echoes computed from the ground truth measurements of the trajectory and the room. In a) we colored the first-order echoes, i.e., the peaks whose propagation times are the closest to the ground truth data from b), and we used the same colors to match the echoes and the walls. Observe that in every recording we can find a peak in a very small neighborhood of the given ground truth propagation time.

and the two closest walls, which enabled us to estimate the geometry of a room and find initial locations of the reference points $\boldsymbol{G}$. The initial guess for the ground truth waypoints $\boldsymbol{R}$ was computed by trilateration: with three randomly selected reference points as anchors and the measured distances between these reference points and $\boldsymbol{r}_n$, for every n.

To improve the initial estimates, we used an alternating coordinate descent method proposed by Dokmanic et al. [48]. It minimizes the well-known *s-stress* cost function [164], i.e., it aims to find the estimates of the points $\boldsymbol{R}$ and $\boldsymbol{G}$, such that their squared distances best match a set of squared measured distances. Since our trilateration algorithm produces a good approximation of $\boldsymbol{R}$ and $\boldsymbol{G}$, the alternating coordinate descent converges fast.

3.6.3 Results

We tested the proposed algorithms in the rectangular and trapezoidal rooms from Fig. 3.11. To provide insight into their performance, we illustrate step-by-step the intermediate results for the trapezoidal room.

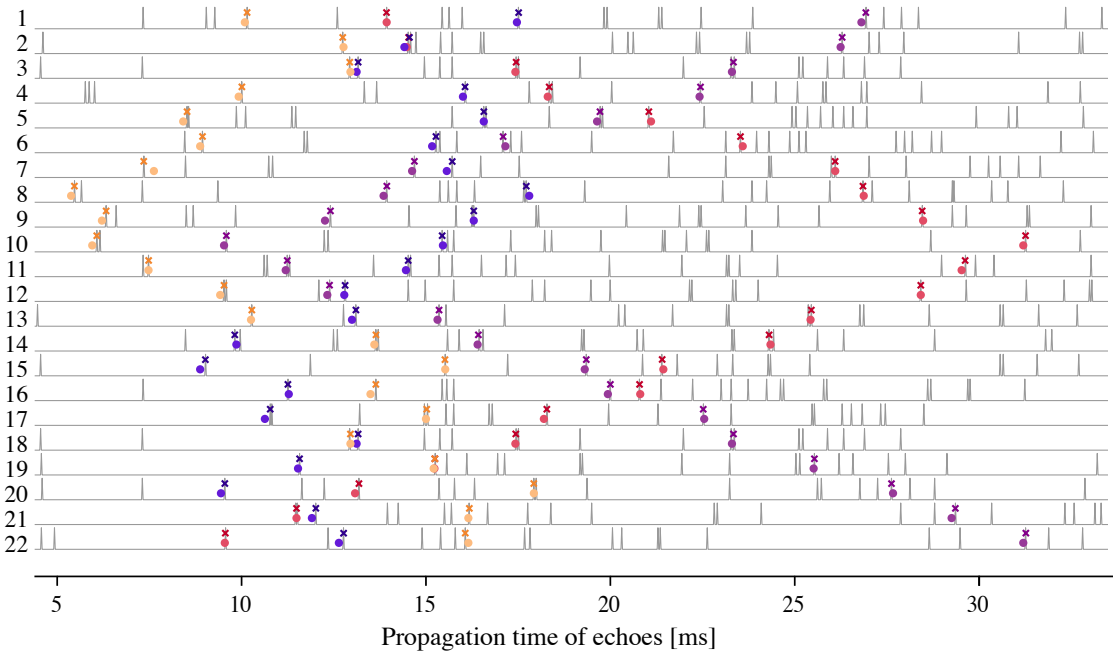


Figure 3.13: The gray signals are obtained from RIRs by removing the reflection from the floor and ceiling, and by extracting the 30 most prominent peaks. Their propagation times are given in $\boldsymbol{\tau}_n$ and they are input to Algorithm 3.1, which detects the first-order echoes, estimates their propagation times $\{\widetilde{\tau}_n\}_{n=1}^N$, and labels them with the corresponding walls. The first-order echoes are marked with $\times$, while their association to different walls is shown in different colors. The ground truth propagation times are indicated by colored dots.

Fig. 3.12a shows the room impulse responses measured at $N = 22$ waypoints. RIRs are depicted in black, while the colored peaks pinpoint the first-order echoes. In every recording we can find all peaks that correspond to the first-order echoes, but in some, the peaks are far and their amplitudes are low (e.g. reflections from the wall C from the step $n = 8$ to $n = 12$). Propagation times of the first-order echoes computed from the ground truth measurements are shown in Fig. 3.12b. These ground truth values coincide well with the recordings; hence, with robust echo detection and echo labeling algorithms we can transform RIRs into point-to-plane distances that are close to the noiseless, ground truth values.

Noiseless trajectory. The result of applying Algorithm 3.1 on the real measurements of RIRs with $\boldsymbol{R}$ being equal to the ground truth trajectory is visualized in Fig. 3.13. The gray signals are the outcome of the processing methods in Section 3.3.1 and Section 3.3.2; they consist of the most prominent peaks extracted from the measured RIRs. The colored dots identify the ground truth propagation times from Fig. 3.12b, while the propagation times $\{\widetilde{\tau}_n\}_{n=1}^N$ of the detected first-order echoes computed by (3.7) are marked with $\times$. Same colors of the peaks at different waypoints link echoes from the same walls. We can observe that the estimated propagation times are well aligned with the ground truth data, while the labeling is exact.

In Fig 3.14a we plot the output of Algorithm 3.1. The four angles for which the error is the smallest almost perfectly overlap with the ground truth angles of the walls; the average error is

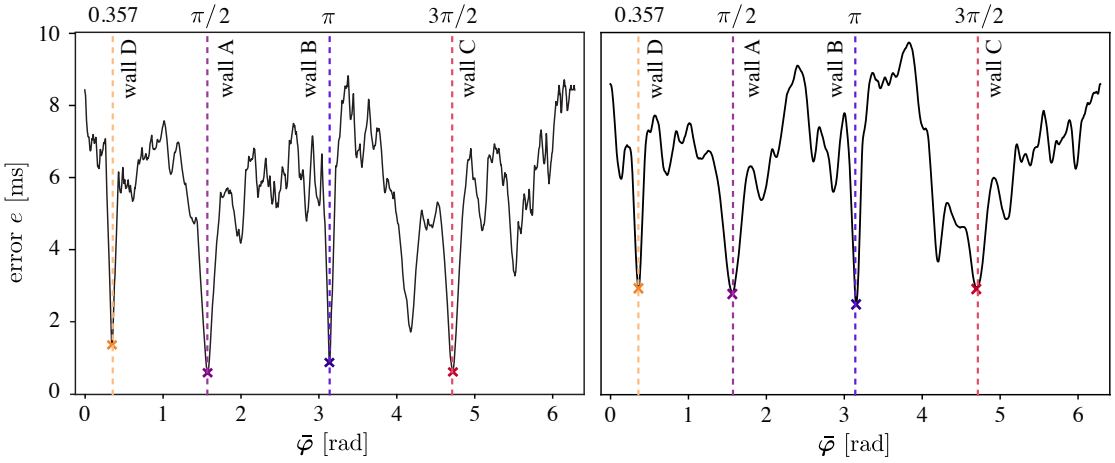


Figure 3.14: Output of Algorithm 3.1 for the a) noiseless and b) noisy estimate of $\boldsymbol{R}$. The ground truth angles of the wall normals are indicated with dashed lines. The angles for which the error is among the four smallest local minima are marked with $\times$. a) The output of Algorithm 3.1 for a noiseless trajectory $\boldsymbol{R}$ and discrete angles $\bar{\varphi}_\ell = \Omega_s \ell$, with $\Omega_s = 0.18°$ and $\ell = 1, \ldots, 2000$. The ℓ^2-norm of the error between the ground truth and the estimated angles is 0.756°, i.e., 0.189° per angle. b) The average of the errors estimated by Algorithm 3.1 for $0.2P$ trajectories with the smallest $\|\boldsymbol{e}^{[p]}\|$ among all P trajectories. The ℓ^2-norm of the error between the estimated and the true angles is 1.651°, i.e., 0.413° per angle.

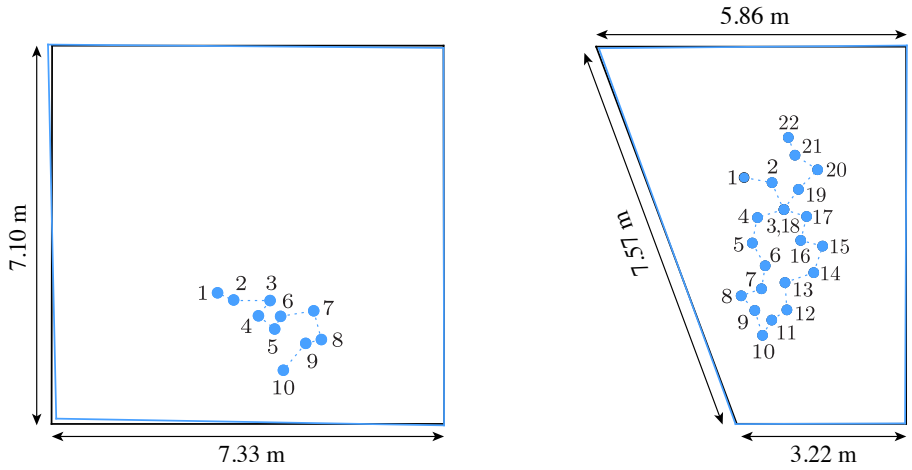


Figure 3.15: Reconstruction of the room-trajectory configuration from the real measurements. The rooms with black walls indicate the ground truth, while the blue walls and waypoints represent the estimates. The blue dotted lines depict the trajectory.

0.189° per angle.

We used Algorithm 3.2 to further refine the reconstruction of the room, but instead of randomly initializing waypoints and walls, we set $\boldsymbol{R}^0 = \boldsymbol{R}$ and $\boldsymbol{\varphi}^0 = \widehat{\boldsymbol{\varphi}}$. Moreover, after every iteration i, we ensured that the estimate of the trajectory $\widehat{\boldsymbol{R}}^i$ does not diverge from the ground truth trajectory by setting $\widehat{\boldsymbol{R}}^i = \boldsymbol{R}$. Fig. 3.15 illustrates the final estimates of the two rooms. The average distances between two waypoints are 46.85 cm and 44.44 cm in the rectangular and trapezoidal room, respectively. We observe that the estimation errors of both rooms are within a few centimeters: in the rectangular room, the RMSE of the room's corners is 5.9 cm, while in the trapezoidal room, it is 2.5 cm.

Noisy estimate of the trajectory. Finally, we illustrate the application of the extended Algorithm 3.1 for the noisy measurement of the trajectory. We used a tape and a protractor to measure the distance and the angle between every two consecutive waypoints, from which we computed noisy motion vectors $\{\widetilde{\boldsymbol{v}}_n\}_{n=2}^N$. By fixing the first waypoint to the origin $\widehat{\boldsymbol{r}}_1 = \boldsymbol{0}$, we obtained the trajectory $\widehat{\boldsymbol{R}}$ from $\widehat{\boldsymbol{r}}_n = \widehat{\boldsymbol{r}}_{n-1} + \widetilde{\boldsymbol{v}}_n$. We assumed $\sigma_v = 4$ cm, since the value is large enough to incorporate the noise of our measurement equipment. Next, we generated P realizations $\widetilde{\boldsymbol{R}}^{[p]}$ and used Algorithm 3.1 to find the corresponding errors, as described in Section 3.3.3.2. In our experiment, we chose $a = 0.2$; from Fig. 3.14b, we can observe that by averaging the errors of $0.2P$ trajectories with the smallest $\|\boldsymbol{e}^{[p]}\|$ we get correct estimates of the wall normals. Similarly to the noiseless case, the four smallest local minima almost perfectly overlap with the true angles. The ℓ^2-norm of the error between the estimated and the true angles is 1.65126°, i.e., 0.4128° per angle.

3.6.4 Discussion

Our experiments show that the proposed algorithms precisely reconstruct room geometries and trajectories when we can detect all first-order echoes in all room impulse responses. Here we discuss two drawbacks to the proposed idea. They have no effect on the algorithm when the measurements are noiseless, but can become critical when the amount of noise or the dimensions of a room exceed a certain threshold.

Disadvantage of the sequential approach. One problematic situation for our sequential approach is already exposed in Observation 3.5. A large level of noise on the trajectory can contribute to misclassification of some second-order echoes as first-order echoes. The iterative algorithm in the second step does not handle outliers; it jointly minimizes over K walls of a room. If the peaks associated to one wall do not correspond to its first-order echoes, the reconstructed room will not have a correct shape even for the remaining $K-1$ walls. Therefore, the error from the first step propagates through the next steps of the algorithm without any chance of being detected and fixed.

Missing echoes. In the measured RIRs it is common that some echoes have the same propagation times such that their peaks overlap, or that the robot is too far from the wall, so that the amplitude of the echo falls below the noise threshold. The overlap of the echoes is not a problem in the proposed approach since we can select the same peak to be a first-order echo several times and assign it to multiple walls. On the contrary, even though Algorithm 3.2 can handle missing entries, Algorithm 3.1 and its extended version cannot; in every RIR they identify peaks that are

the closest to the first-order echoes of a given trajectory. As such, they always provide a complete set of propagation times $\{\widetilde{\tau}_n\}_{n=1}^N$ of the detected first-order echoes for every $\bar{\varphi}_\ell$, $\ell = 1, \ldots, L$. These weaknesses of the algorithm motivate us to approach the problem from a probabilistic perspective in Chapter 4 and gradually build a room after every new RIR measurement.

3.7 Conclusion

In this chapter, we addressed the practical SLAM problem that motivated the study of point-to-plane distance matrices in the previous chapter. As earlier, we assumed a mobile device equipped with a single omnidirectional loudspeaker and a single omnidirectional microphone that measures room impulse responses at several locations inside a room, and tries to recover the unknown room geometry, as well as to localize itself inside of it.

We proposed an end-to-end pipeline of algorithms that solves the room reconstruction problem from a few measurements of room impulse responses obtained by a mobile device in real environments. Point-to-plane distances play an important role in the intermediate step of the pipeline, which has two main stages: 1) algorithms that receive room impulse responses as input, process them, measure the propagation times of the detected and labeled first-order echoes, and output an incomplete and noisy PPDM, and 2) an iterative algorithm for low-rank matrix factorization, which receives a PPDM as an input and outputs the reconstructed room geometry along with the trajectory of the device.

We showed through extensive numerical simulations that our iterative method in the second stage of the pipeline outperforms existing solutions, while at the same time allowing a larger number of missing entries in the measurement matrices. To show the robustness and reliability of the first part of the pipeline, we undertook real experiments in two distinct rooms. We demonstrated that the algorithms accurately reconstruct the rooms and trajectories, even in the presence of noise and unlabeled data.

3.A Summary of the MUSIC algorithm

The variant of the MUSIC algorithm [114] that we used to detect the K most prominent peaks in a RIR requires building a Toeplitz matrix from the discrete Fourier Transform $F[m]$ of the RIR $f[\ell]$:

$$F[m] = \sum_{\ell=0}^{L} f[\ell] e^{-\frac{2\pi j}{L} m\ell}.$$

The columns of the Toeplitz matrix are constructed by translating the data samples $\{F[m]\}$ as:

$$\boldsymbol{T} = \begin{bmatrix} F[K] & F[K-1] & \dots & F[0] \\ F[K+1] & F[K] & \dots & F[1] \\ \vdots & \vdots & \ddots & \vdots \\ F[L] & F[L-1] & \dots & F[L-K] \end{bmatrix}^{H} \begin{bmatrix} F[K] & F[K-1] & \dots & F[0] \\ F[K+1] & F[K] & \dots & F[1] \\ \vdots & \vdots & \ddots & \vdots \\ F[L] & F[L-1] & \dots & F[L-K] \end{bmatrix},$$

where ^{H} denotes a conjugate transpose of the matrix. We next compute the singular value decomposition (SVD) of $\boldsymbol{T}$ to separate the components relative to the K largest singular values from the others:

$$\boldsymbol{T} = \boldsymbol{U}_s \boldsymbol{\Sigma}_s \boldsymbol{V}_s^{\top} + \boldsymbol{U}_n \boldsymbol{\Sigma}_n \boldsymbol{V}_n^{\top},$$

where the subscript $_s$ indicates that the range space of $\boldsymbol{U}_s$ is the signal space, while the subscript $_n$ indicates that the range space of $\boldsymbol{U}_n$ is the noise space. The orthogonal projector onto the noise subspace is constructed as $\boldsymbol{P}_n = \boldsymbol{U}_n \boldsymbol{U}_n^{\top}$.

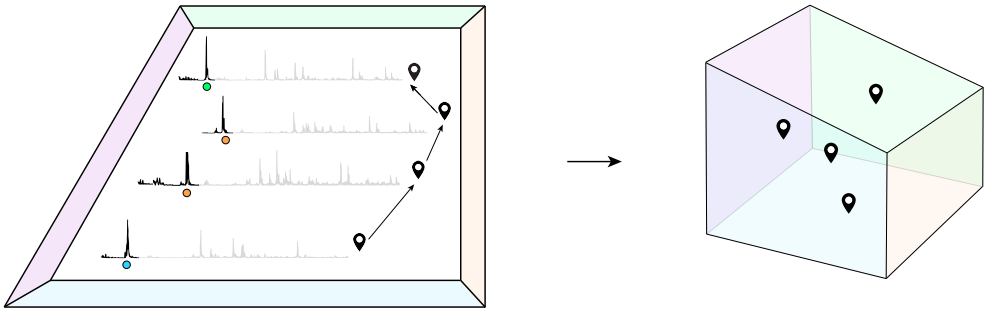
Finally, we find the desired importance factor $\alpha(t)$ of all arrival times $t \in [0, \frac{2\pi}{\Delta_\omega}]$ from:

$$\alpha(t) = \frac{1}{\|\boldsymbol{P}_n e^{j\boldsymbol{\omega} t}\|},$$

where $\boldsymbol{\omega}$ is a vector with K consecutive frequencies on a grid of spacing Δ_ω. The locations of the K values of $\alpha(t)$ with the largest magnitude correspond to the locations of the K most prominent peaks in the RIR $f[\ell]$.

Chapter 4

EchoSLAM: Simultaneous Localization and Mapping with Acoustic Echoes*



In theory, there is no difference between theory and practice.
But in practice, there is.

remark overheard at a computer science conference
Jan L. A. van de Snepscheut

*The material in this chapter is the result of joint work of the author (MK) with Yassine Ahaggach (YA). Author contributions: MK designed research, devised and implemented all algorithms, performed numerical simulations, and wrote the chapter. MK created all figures except Fig. 4.12b, which was created by YA. Sections 4.3 and 4.4 are written based on [94], which was written by MK. Section 4.7.1 is written based on [1], which was written by YA. YA designed and built the robot, and assisted MK with real experiments.
The author would like to thank Gilles Baechler for his detailed and valuable comments on earlier versions of this chapter. His advice and thoroughness significantly contributed to the chapter, and thanks to his sharp eye many "the"s got converted to "a"s, and many "a"s mysteriously disappeared from where they should not have been in the first place.

4.1 Introduction

In this chapter we formulate a probabilistic approach to joint reconstruction of rooms and trajectories from room impulse responses (RIRs) recorded by a robot equipped with a single omnidirectional source and receiver. We name the proposed algorithm EchoSLAM. In comparison with the two-step method from the previous chapter, this new approach offers a more robust solution as it relies only on the first-order echo from the nearest wall at every waypoint. Another important distinction from the previous method concerns the estimates of trajectories. In Chapter 3, we use this information only in the first step of the algorithm to discriminate the first-order echoes from other peaks in a RIR and to label them with the correct walls. In the new approach, we jointly update the belief of the trajectory and the positions of the walls.

4.1.1 Related work

Different flavors of simultaneous localization and mapping (SLAM) involve different sensing modalities, but our interest is primarily SLAM from reflections of sound or radio waves from walls, as well as solutions based on multiple sensor modalities [30, 55, 104, 125]. The existing literature can be broadly categorized into three groups.

In the first group, a mobile robot is equipped with a microphone array or multimodal sensors, and it aims to localize multiple interfering sound sources inside a reverberant environment [57, 58, 77, 87, 139]. These methods often require high-end equipment [55] or precise calibration of microphone positions in the array, sometimes combined with beamforming techniques [87, 139].

The second group involves more recent works that rely on simpler mobile devices equipped with either a source or a receiver [45, 85, 86, 107, 109, 112, 123]. To recover rooms, the devices leverage multipath wave propagation from static anchors, and the fact that the echoes correspond to range measurements from *virtual* anchors. Combined with Bayesian techniques, such as belief propagation that jointly performs data association and estimates the state of the mobile agent and the environment, this leads to robust SLAM [107]. Beyond timing, valuable information is contained in the amplitude statistics of the multipath components [108]. Shih et al. present a setup in which the mobile agent is a commodity smartphone acting as a receiver, while the source is fixed at an unknown location; their approach handles unlabeled, missing and spurious echoes in realistic environments [158]. However, in very large spaces the number of missing echoes exceeds the operational capabilities of the method. Another notable drawback is that the reconstruction accuracy depends not only on the receiver's trajectory, but also on the position of the static anchors.

The third group, which we address here, comprises setups with no fixed beacons inside rooms, and a mobile robot capable of only rudimentary sensing (equipped with a source and one or two receivers). This group has received the least attention even though it is appealing from a cost and technological perspective. It has been shown that the distances between the smartphone and the nearby walls revealed from the short-range scans by a sound source and two microphones on a smartphone can be used to reconstruct the geometry of walls [143, 181]. However, the method proposed by Zhou et al. requires the user to walk a full loop closely to the walls, hold a smartphone in a specific position and perform a particular measurement gesture that is prone to error [181]. The system presented by Pradhan et al. resolves these drawbacks and achieves much smaller reconstruction error [143]. In both papers however, the algorithms build the map of the room based on the estimated user's locations without correcting them. As the locations are estimated by dead-reckoning, they are not exact and the error propagates to the next steps.

Moreover, the microphones of a smartphone are directional, so the algorithms require information about the phone's orientation at every measurement. Steckel et al. [162] and Schouten et al. [155] present a low-cost radar sensor (a biomimetic sonar) with one emitter and two receivers that relies on the echolocation-related transfer function (ERTF) to realize a robotic platform capable of autonomous navigation through an unknown environment. Schouten et al. propose an ERTF that combines the radiation pattern of the emitter and the directivity pattern of the receiver to derive the arrival directions of echoes [155]. Combined with the estimated distances to reflectors, the system can precisely reconstruct 3D rooms and localize the robot. This is unlike our proposed algorithm, in which we assume to have one emitter and one receiver, and where omnidirectionality precludes any a priori knowledge about the angles of arrival of echoes. On the one hand, this renders the problem harder, but on the other hand, the solution is of lower cost.

4.1.2 Main contributions

In this chapter we present the EchoSLAM algorithm that overcomes many limitations of the abovementioned methods. In particular,

- it only assumes a minimal and low-cost sensing equipment: one omnidirectional microphone and one omnidirectional speaker mounted on a robot,
- it does not require static speakers or anchors,
- it does not pose constraints on room geometries,
- it does not require the number of walls to be known,
- it does not expect a complete set of first- or higher-order echoes in RIRs; it relies only on one (the first) echo from every RIR,
- it jointly updates estimated locations of the robot and estimated geometry of a room with every new measurement,
- it provides an efficient and robust labeling algorithm,
- it achieves almost real-time execution time.

4.1.3 Outline

To lay down the main ideas of EchoSLAM, we build up gradually from a simpler model. We first assume that the measurements are not corrupted by noise and we propose a deterministic solution based on elementary trigonometry in Section 4.3. The underlying idea is that every wall is tangent to the circles with centers in the waypoints and radii being equal to the propagation times of the first-order echoes scaled by one-half of the speed of sound. Given the waypoints and first-order echoes, finding the positions of the walls is therefore straightforward; to localize a wall, we only need to find a common tangent to a given set of circles.

In Section 4.4 we generalize the problem setup by assuming a noisy trajectory, and we introduce a simple probabilistic approach to estimate the walls. Its main benefit is that we can show that the algorithm converges to the correct solution when the robot explores the space randomly.

In real-world scenarios, however, we do not have noiseless information about the robot's waypoints, nor do we have the noiseless measurements of the first-order echoes. To address

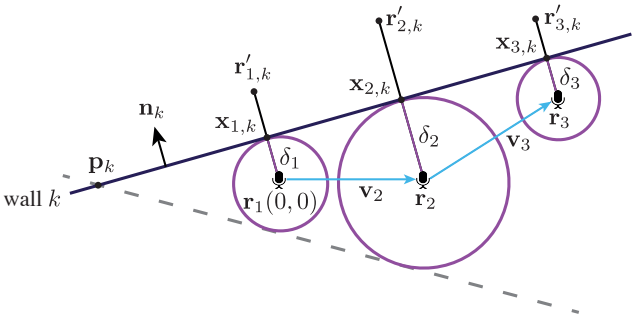


Figure 4.1: Setup with collocated loudspeaker and microphone mounted on a robot. Distances from the wall are denoted by δ_n and shown in purple, while the motion vectors $\boldsymbol{v}_n$ are in blue. The position of the wall k is unique assuming noiseless measurements and nonlinear trajectories.

these challenges, we explicitly model the noise in the measured RIRs and the robot's motion in Section 4.5. Inspired by FastSLAM [127], we use a particle filter approach to represent the posterior over the robot's waypoints and room geometry. We empirically observe that this more sophisticated algorithm works well in the presence of noise. To demonstrate its accuracy and efficiency in real environments, we test the algorithm on a modular three-wheel robot of our own production and confirm the excellent performance of the algorithm.

4.2 Problem setup

As in the previous chapters, we have a robot with a collocated microphone and loudspeaker that explores a room along a trajectory following a set of waypoints $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N$. At every waypoint $\boldsymbol{r}_n$, we measure a room impulse response from which we can extract the first peak that appears after the direct sound. Its propagation time τ_n corresponds to the round-trip time of the signal produced by the loudspeaker, reflected from the nearest wall, and registered by the microphone. From τ_n, we compute the distance between the waypoint and its nearest wall, $\delta_n = c_s\tau_n/2$. Given a point-to-plane distance matrix $\boldsymbol{D}$ from (2.4), we can define δ_n as:

$$\delta_n = \min_{k\in\{1,\ldots,K\}} D_{nk} \quad \text{for every } n = 1, \ldots, N.$$

In addition to δ_n, we assume that we can also measure the robot motion vectors, $\boldsymbol{v}_n$, at every n. The case of collocated microphone and loudspeaker is illustrated in Fig. 4.1.

Since the source and the microphone are collocated, it is not possible to discriminate between translated, rotated and reflected variants of the room about the robot. We resolve this ambiguity by fixing the initial robot's position $\boldsymbol{r}_1$ at the origin and the orientation of the first step to 0.

4.2.1 Motion model

To compute the robot's locations over time, we use the control input. For many real robots (including our platform, see Section 4.7.1), this yields very accurate position estimates. The control uses the relative motion information, which in our case consists of a rotation angle $\widetilde{\phi}_n$

and a translation $\widetilde{\ell}_n$. We use a tilde ($\widetilde{\cdot}$) to distinguish between the ground truth motion of the robot, $\boldsymbol{v}_n(\ell_n, \phi_n)$, and our belief of its motion (i.e., input control commands), $\widetilde{\boldsymbol{v}}_n(\widetilde{\ell}_n, \widetilde{\phi}_n)$.

Given the robot's position $\boldsymbol{r}_{n-1}$ and the control $\widetilde{\boldsymbol{v}}_n$ at step $n-1$, we can predict the next robot's location $\boldsymbol{r}_n$,

$$\boldsymbol{r}_n = \boldsymbol{r}_{n-1} + \widetilde{\ell}_n \begin{bmatrix} \cos\widetilde{\phi}_n & \sin\widetilde{\phi}_n \end{bmatrix}^\top,$$

where we assume that both parameters $\widetilde{\ell}_n$ and $\widetilde{\phi}_n$ are corrupted by independent Gaussian noise,

$$\widetilde{\ell}_n \sim \mathcal{N}(\ell_n, \sigma_\ell^2), \quad \widetilde{\phi}_n \sim \mathcal{N}(\phi_n, \sigma_\phi^2). \tag{4.1}$$

4.2.2 Measurement model

The robot moves inside the room and acquires information about its environment by recording the room impulse responses at different locations along its trajectory. From every recorded RIR, we first remove the direct sound and echoes from the ceiling and the floor. Most of the rooms have ceilings parallel to the floor and the height of the robot does not change, so the reflections from the ceiling and the floor arrive with the same time delay in all RIRs, which makes their detection and elimination easy. Then, we extract only the first peak; it corresponds to the first-order echo from the nearest wall and reveals the distance of the robot from the nearest wall. Again, as we use a single omnidirectional loudspeaker to play the sound and a single omnidirectional microphone to record the RIR, it is not a priori possible to obtain any information about the orientation of the wall from which the echo reflected.

We define a probabilistic measurement model for those range measurements based on experimental results with a real robot. To incorporate the uncertainties of our omnidirectional sonar, we model the measurements with a conditional probability distribution $p(\delta_n \mid \boldsymbol{r}_n, \mathcal{P})$, where $\boldsymbol{r}_n$ is the location of the robot and $\mathcal{P} = \{\mathcal{P}_k\}_{k=1}^K$ is the set of walls. We assume that the measurements have Gaussian distributions, $\delta_n \sim \mathcal{N}(d_{nk}, \sigma_\delta^2)$, where the mean d_{nk} is equal to the true distance between $\boldsymbol{r}_n$ and the nearest wall in $\mathcal{P}$, and the standard deviation σ_δ is constant for all $n = 1, \ldots, N$:

$$p(\delta_n \mid \boldsymbol{r}_n, \mathcal{P}) = \frac{1}{\sigma_\delta\sqrt{2\pi}} \exp\left(-\frac{(\delta_n - d_{nk})^2}{2\sigma_\delta^2} \right). \tag{4.2}$$

4.3 Deterministic case: noiseless motion, noiseless distances

We first assume that both the trajectory and the measured distances are noiseless. In other words, $\widetilde{\ell}_n = \ell_n$, $\widetilde{\phi}_n = \phi_n$ and $\delta_n = d_{nk}$, or, $\sigma_\ell = \sigma_\phi = \sigma_\delta = 0$.

We focus on the wall k of the room and explain how to localize it with reference to Fig. 4.1. Without loss of generality, we assume that the wall k is the nearest wall for all selected waypoints $\{\boldsymbol{r}_n\}_{n=1}^N$. Given the noiseless motion vectors, we calculate the robot's position at any step n as $\boldsymbol{r}_n = \sum_{i=2}^n \boldsymbol{v}_i$.

PROPOSITION 4.1 We can uniquely determine the wall k from three noiseless measurements obtained at non-collinear waypoints as a line passing through the points $\boldsymbol{x}_{nk}$:

$$\boldsymbol{x}_{nk} = \boldsymbol{r}_n + \frac{\delta_n^2}{\|\boldsymbol{u}_{nk}\|^2}\boldsymbol{u}_{nk} \pm \frac{\delta_n\sqrt{\|\boldsymbol{u}_{nk}\|^2 - \delta_n^2}}{\|\boldsymbol{u}_{nk}\|^2} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \boldsymbol{u}_{nk}, \tag{4.3}$$

where

$$\boldsymbol{p}_k = \frac{\delta_1}{\delta_1 - \delta_2}\boldsymbol{r}_2 - \frac{\delta_2}{\delta_1 - \delta_2}\boldsymbol{r}_1$$

is an intersection point of the outer tangents of two circles with centers $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ and radii δ_1 and δ_2, and

$$\boldsymbol{u}_{nk} = \boldsymbol{p}_k - \boldsymbol{r}_n.$$

PROOF. The vector between the image source and the robot's waypoint is given by

$$\boldsymbol{y}_{nk} = \boldsymbol{r}'_{nk} - \boldsymbol{r}_n = 2\big\langle \boldsymbol{p}_k - \boldsymbol{r}_n, \boldsymbol{n}_k \big\rangle \boldsymbol{n}_k$$

for all n. The direction of $\boldsymbol{y}_{nk}$ is perpendicular to the wall and the length of $\boldsymbol{y}_{nk}$ equals twice the distance between the robot and the wall. The only line that satisfies both conditions for all n is the common tangent of the circles with centroids at $\boldsymbol{r}_n$ and radii δ_n given by (4.3). ■

As Fig. 4.1 shows, having only two measurements gives two solutions for the wall—there are two common tangents of two circles such that the centroids are on the same side of the tangents. They are shown as a black line and a dashed gray line in the figure. Assuming that the robot does not move along a line, three measurements are sufficient to get a unique solution.

4.4 Probabilistic case: noisy motion, noiseless distances

Now, we slightly relax our assumption from the previous section and suppose that the distance measurements are noiseless $\sigma_\delta = 0$, while the motion vectors between consecutive waypoints are corrupted by Gaussian noise as in (4.1). We introduce a simple estimator to gain insight into the wall localization problem.

We model the wall as a line with unknown slope $\tan\theta_k$ and distance from the origin q_k. As in the previous section, we assume that the wall k is the closest to all selected waypoints $\{\boldsymbol{r}_n\}_{n=1}^N$. As this includes $\boldsymbol{r}_1 = \boldsymbol{0}$, the distance of the wall from the origin is known and equal to $q_k = \delta_1$.

In the procedure described below, we propose to estimate θ_k for every pair of consecutive measurements δ_{n-1} and δ_n, and the noisy motion vector $\widetilde{\boldsymbol{v}}_n(\widetilde{\ell}_n, \widetilde{\phi}_n)$. We can further refine this estimate by simply averaging independent estimators after every robot's step, which by the law of large numbers converges to the true wall.

PROPOSITION 4.2 Assume that $\ell_n = \ell$ for every n. We define the estimator of θ_k for each measurement n as:

$$\begin{aligned}\hat{\theta}_k^n &= \phi_n + \arcsin\left(\frac{\delta_n - \delta_{n-1}}{\ell}\right)\\ &= \phi_n + \arcsin\left(\frac{\widetilde{\ell}_n \sin(\theta_k - \widetilde{\phi}_n)}{\ell}\right),\end{aligned}$$

and the final estimate of θ_k after N measurements as:

$$\hat{\theta}_k^{1:N} = \frac{1}{N}\sum_{n=1}^{N} \hat{\theta}_k^n.$$

Then, $\hat{\theta}_k^{1:N}$ is unbiased when ϕ_n is uniformly distributed on the circle.

PROOF. The ratio $\widetilde{\ell}_n/\ell$ is distributed as $\mathcal{N}(1, \sigma_\ell^2/\ell^2)$ for all n, while $\theta_k - \widetilde{\phi}_n \sim \mathcal{N}(\theta_k - \phi_n, \sigma_\phi^2)$. Then, one can verify that the bias of $\hat{\theta}_k^n$ depends only on the parameter ϕ_n. Therefore, we rewrite it as $\hat{\theta}_k^n = \theta_k + f(\phi_n)$, where $f(\cdot)$ is periodic with zero mean over the period. We observe that

$$\begin{aligned}\mathbb{E}(\hat{\theta}_k^{1:N}) &= \mathbb{E}\Big(\frac{1}{N}\sum_{n=1}^{N}(\theta_k + f(\phi_n)\Big) \\ &= \theta_k + \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}(f(\phi_n)),\end{aligned}$$

and the uniform distribution of ϕ_n on the interval $[0, 2\pi]$ provides that $\mathbb{E}(f(\phi_n)) = 0$ for every n, so that $\mathbb{E}(\hat{\theta}_k^{1:N}) = \theta_k$. ■

The physical meaning is as follows: the estimator is positively biased if the robot walks towards the wall $\mathbb{E}(\hat{\theta}_k^{1:N}) - \theta_k \geq 0$, and negatively biased if the robot walks away from the wall $\mathbb{E}(\hat{\theta}_k^{1:N}) - \theta_k \leq 0$. These biases cancel out for a robot that picks its direction at random. As we assume that the robot performs a random walk, the values of ϕ_n are uniformly distributed on the circle, and one can verify that the function has zero mean. Thus $\mathbb{E}(f(\phi_n)) = 0$. By the law of the large numbers, the sequence of estimates $\hat{\theta}_k^n$ converges to the real value θ_k.

4.5 EchoSLAM: noisy motion, noisy distances

Now we consider realistic conditions in which the measurements of the robot's positions and the room impulse responses are noisy. We propose a probabilistic algorithm called EchoSLAM to estimate the posterior of the trajectory and walls,

$$p(\{\boldsymbol{r}_n\}_{n=1}^N, \mathcal{P} \mid \{\delta_n\}_{n=1}^N, \{\boldsymbol{v}_n\}_{n=2}^N),$$

given a sequence of motion vectors $\{\boldsymbol{v}_n\}_{n=2}^N$ and the distance measurements $\{\delta_n\}_{n=1}^N$.

4.5.1 Map: a probabilistic description of the walls

The map of the environment that we want to recover is a 2D room $\mathcal{P}$ defined as a set of K walls, $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K\}$. It belongs to the category of *feature-based* maps, where features are the walls that we want to localize. Every wall $\mathcal{P}_k$ is a line uniquely determined by two parameters: the angle of its incline $\theta_k \in [-\pi/2, \pi/2)$, and the signed distance from the origin q_k, where the sign of q_k indicates if the intercept of the line is positive or negative.

In our probabilistic approach, building the map of the environment requires to estimate the probability of every wall defined with the tuple (θ_k, q_k) to generate the measurement δ_n for a given $\boldsymbol{r}_n$. We denote the corresponding probability distribution by $f_\Theta^{(n)}(\theta_k, q_k)$.

To find $f_\Theta^{(n)}(\theta_k, q_k)$, let us use the Cartesian form of lines

$$y = \tan\theta_k x + q_k\sqrt{1 + \tan^2\theta_k}.$$

The distance of the line from the origin is $|q_k|$, while its distance from $\boldsymbol{r}_n = [r_{n,x}, r_{n,y}]^\top$ is:

$$d_{nk} = |\sin\theta_k r_{n,x} - \cos\theta_k r_{n,y} + q_k| \,. \tag{4.4}$$

By assuming the measurement model from Section 4.2.2, we can substitute the value of d_{nk} from (4.4) into (4.2), and obtain

$$f_\Theta^{(n)}(\theta_k, q_k) = \\ \frac{\eta}{\sigma_\delta\sqrt{2\pi}} \begin{cases} \exp\left(\frac{-1}{2\sigma_\delta^2}\left(q_k + \sin\theta_k r_{n,x} - \cos\theta_k r_{n,y} - \delta_n\right)^2\right) & \text{for } \sin\theta_k r_{n,x} - \cos\theta_k r_{n,y} + q_k \geq 0, \\ \exp\left(\frac{-1}{2\sigma_\delta^2}\left(q_k + \sin\theta_k r_{n,x} - \cos\theta_k r_{n,y} + \delta_n\right)^2\right) & \text{for } \sin\theta_k r_{n,x} - \cos\theta_k r_{n,y} + q_k < 0, \end{cases} \tag{4.5}$$

where η is a normalization factor.[1]

For the values of the noise level σ_δ present in the real experiments, we can simplify the computations by well-approximating (4.5) as:

$$f_\Theta^{(n)}(\theta_k, q_k) = \frac{\eta}{\sigma_\delta\sqrt{2\pi}} \exp\left(\frac{-\left(q_k + \sin\theta_k r_{n,x} - \cos\theta_k r_{n,y} - \delta_n\right)^2}{2\sigma_\delta^2}\right) + \\ \frac{\eta}{\sigma_\delta\sqrt{2\pi}} \exp\left(\frac{-\left(q_k + \sin\theta_k r_{n,x} - \cos\theta_k r_{n,y} - \delta_n\right)^2}{2\sigma_\delta^2}\right).$$

Given the waypoint $\boldsymbol{r}_n$ and the measurement δ_n, $f_\Theta^{(n)}(\theta_k, q_k)$ is a probability distribution of lines to be the nearest wall to $\boldsymbol{r}_n$. The two terms in the sum (4.5) are due to positive and negative arguments of the absolute value in (4.4).

Fig. 4.2 illustrates $f_\Theta^{(n)}(\theta_k, q_k)$ for two different waypoints $\boldsymbol{r}_n$. Clearly, when $\boldsymbol{r}_n$ is in the origin, the most likely walls are at the distance δ_n from the origin, $q_k = \pm\delta_n$ for any θ_k. For $\boldsymbol{r}_n = [2, 0]^\top$ the distance of the most likely walls from the origin is either

$$q_k = -\sin\theta_k r_{n,x} + \cos\theta_k r_{n,y} + \delta_n = -2\sin\theta_k + 0.6 \text{ (the upper curve), or}$$
$$q_k = -\sin\theta_k r_{n,x} + \cos\theta_k r_{n,y} - \delta_n = -2\sin\theta_k - 0.6 \text{ (the lower curve).}$$

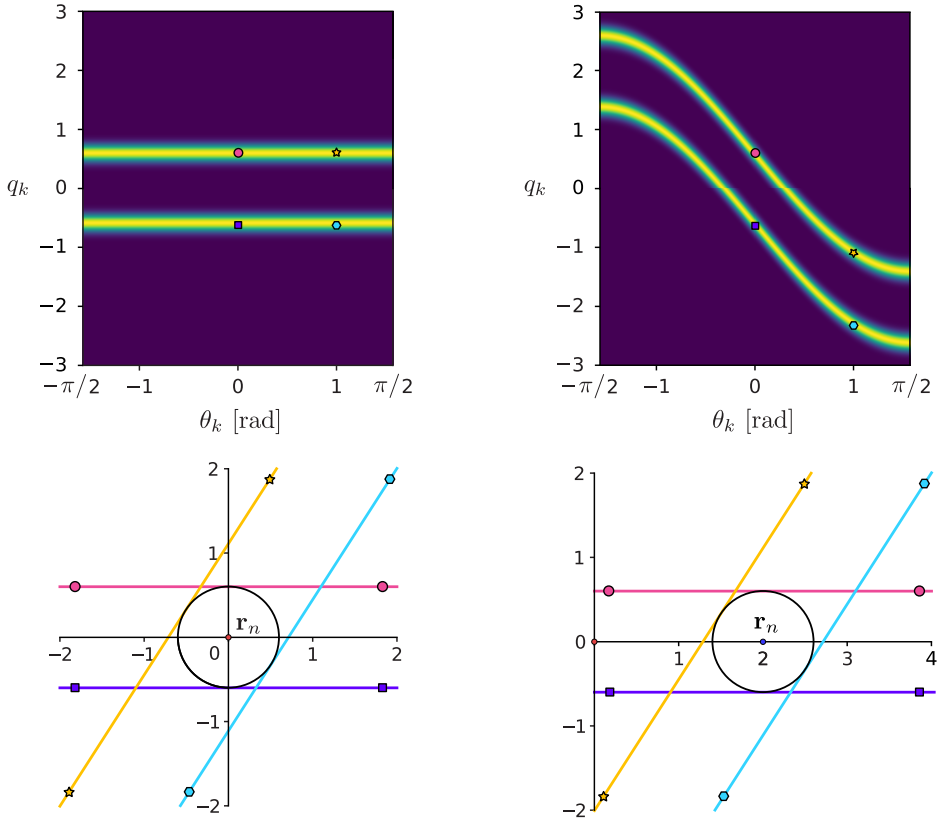
In the noiseless case, these two equations represent the distance of the tangents to the circle with center $\boldsymbol{r}_n = [2, 0]^\top$ and radius $\delta_n = 0.6$ from the origin. In the noisy case, the width of the curves is proportional to the amount of the measurement noise.

To reconstruct the map, we need to compute $f_\Theta^{(1:N)}(\theta_k, q_k) = p(\mathcal{P}_k \mid \{\boldsymbol{r}_i\}_{i=1}^N, \{\delta_i\}_{i=1}^N)$, the distribution of a wall k at step N given all past robot's positions $\{\boldsymbol{r}_i\}_{i=1}^N$ and measurements $\{\delta_i\}_{i=1}^N$. With the closed-form expression for $f_\Theta^{(n)}(\theta_k, q_k)$, it is not hard to do this recursively: we obtain $f_\Theta^{(1:n)}(\theta_k, q_k)$ from $f_\Theta^{(1:n-1)}(\theta_k, q_k)$ by incorporating a new measurement δ_n. The details are given in Section 4.5.4.

4.5.2 Particle filter approach to SLAM

The idea of our algorithm comes from the work of Montemerlo et al. [127], in which they propose to use particle filters to represent the posterior over robot's locations and landmarks. Similarly,

[1] Throughout the chapter we reuse the same symbol η to indicate that the function needs to be normalized to 1 to be a probability density function; hence, the normalization factor η in different equations can have different values.



a) $\mathbf{r}_n = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\delta_n = 0.6$, $\sigma_\delta = 0.1$ b) $\mathbf{r}_n = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$, $\delta_n = 0.6$, $\sigma_\delta = 0.1$

Figure 4.2: *Above*: Probability distributions $f_\Theta^{(n)}(\theta_k, q_k)$ of walls (θ_k, q_k) that generate the measurement δ_n for two different values of $\boldsymbol{r}_n$. If $\boldsymbol{r}_n = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$, $f_\Theta^{(n)}(\theta_k, q_k \geq 0) = -f_\Theta^{(n)}(\theta_k, q_k < 0)$ for every (θ_k, q_k), and $f_\Theta^{(n)}(\theta_k = \alpha_1, q_k) = f_\Theta^{(n)}(\theta_k = \alpha_2, q_k)$ for any α_1 and α_2. *Below*: An example of mapping four lines from a Cartesian coordinate system to their locations in $f_\Theta^{(n)}(\theta_k, q_k)$. The lines are marked with the corresponding colors and symbols. They are chosen such that the pink and purple lines are the two most likely walls for $\theta = 0$, whereas yellow and blue are the most likely for $\theta_k = 1$.

we use particle filters to model the posterior over robot's locations and walls: in our case, each particle contains the estimated robot locations and a set of distributions for every wall in the room.

Particle methods are a set of powerful simulation-based methods which rely on samples to approximate probability density distributions. The key idea is to represent the posterior by a set of random state samples drawn from that posterior. They are a subset of Sequential Monte Carlo methods and they generalize the traditional Kalman filtering; they can be applied to non-linear models with non-Gaussian errors, and they are computationally tractable for higher dimensional problems [6, 50].

Advantages of particle filters. We list several advantages of the particle approach to SLAM over other methods:

1. The FastSLAM algorithm is the first to recognize and exploit the fact that knowledge of the robot's trajectory renders the individual landmark measurements independent [130]. This observation allows to decompose the SLAM problem into a robot localization problem and K independent landmark estimation problems. More formally, we factor the posterior over the entire trajectory $\{\boldsymbol{r}_n\}_{n=1}^N$ and the room geometry $\mathcal{P} = \{\mathcal{P}_k\}_{k=1}^K$ as:

$$p(\{\boldsymbol{r}_n\}_{n=1}^N, \mathcal{P} \mid \{\delta_n\}_{n=1}^N, \{\boldsymbol{v}_n\}_{n=2}^N) = p(\{\boldsymbol{r}_n\}_{n=1}^N \mid \{\delta_n\}_{n=1}^N, \{\boldsymbol{v}_n\}_{n=2}^N) \prod_{k=1}^K p(\mathcal{P}_k \mid \{\boldsymbol{r}_n\}_{n=1}^N, \{\delta_n\}_{n=1}^N).$$

2. Particle filters can cope with non-linear robot motion models as they approximate distributions by a finite number of drawn samples. Therefore, they do not require any assumptions on the density functions and they can represent multimodal beliefs.

3. The data association, i.e., identifying a landmark that is observed in a given step and associating it to the measurement, is made on a per-particle basis, so the filter can maintain multiple data associations simultaneously. If we sample over all data associations, we can obtain an approximation of its full posterior. This is one of the key advantages of FastSLAM over other methods which track only a single data association at any point in time [168]. In the algorithm proposed in this chapter, we take this idea even further.

Distinctions from FastSLAM. Before we get into the details of the algorithm, we introduce the particularities of our problem setup in comparison to the original FastSLAM formulation.

1. Posterior of landmark estimates: Every particle in the FastSLAM algorithm contains a set of Kalman filters for every landmark in the map. They are defined with mean and covariance. The landmark estimates are updated by an Extended Kalman Filter (EKF).

   In our problem, the landmarks are walls. This introduces two important distinctions from the standard solutions: 1) As a set of circles can have two common tangents, one on each side, one particle can have two different posterior distributions associated to the same landmark (wall). In practice, this occurs when the robot follows a linear trajectory, or close to linear in the noisy case. For every new measurement we need to compute two separate distributions (one for the lines for which the waypoints are on their left side, and another one for the lines for which the waypoints are on their right side[2]) and update the wall estimates independently for each side. 2) The dimensionality of our measurements is lower than that of the walls, so we cannot rely on EKF to update the posterior distributions of walls. Instead, we derive exact expressions for the distributions after every new measurement, and approximate them by fitting Gaussian functions.

2. Data association: In the typical particle filter approach, every particle contains a sequence of labels which indicate the order of the measured landmarks. When a new measurement is added, only one landmark is associated to it, for example by maximizing the probability that it has generated the measurement. There exist numerous other methods [127].

[2]We can tell if a point is on the left or the right side of the line because the slopes of lines are determined by $\tan\theta_k$ for $\theta_k \in [-\pi/2, \pi/2)$. Therefore, we have directed lines with the direction vector $[\cos\theta_k, \sin\theta_k]^\top$.

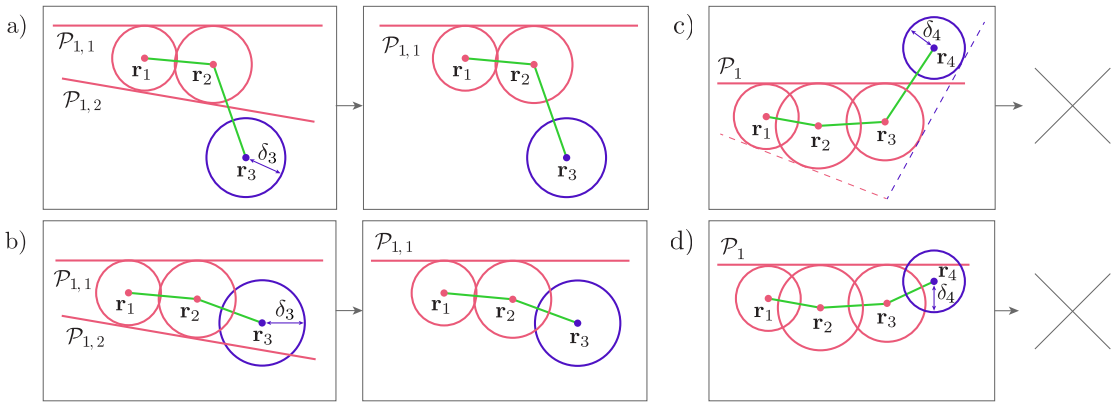


Figure 4.3: In a) and b) $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are associated to the first wall, while $\boldsymbol{r}_3$ is associated to the second wall. There are two common tangents $\mathcal{P}_{1,1}$ and $\mathcal{P}_{1,2}$ around $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ that could be the wall $\mathcal{P}_1$; they are illustrated in red. In a) $\boldsymbol{r}_3$ is located on the other side of $\mathcal{P}_{1,2}$, while in b) the distance of $\boldsymbol{r}_3$ from $\mathcal{P}_{1,2}$ is smaller than δ_3 (the distance of $\boldsymbol{r}_3$ from the wall $\mathcal{P}_2$). In both cases, $\mathcal{P}_{1,2}$ cannot be part of the solution set, so we remove it. In c) and d) $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$ are associated to the first wall, while $\boldsymbol{r}_4$ is associated to the second wall. There is only one common tangent to $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$ which defines the wall $\mathcal{P}_1$. In c) $\boldsymbol{r}_4$ is located on the other side of $\mathcal{P}_1$, while in d) the distance of $\boldsymbol{r}_3$ from $\mathcal{P}_1$ is smaller than δ_4 (the distance of $\boldsymbol{r}_3$ from $\mathcal{P}_2$). In both cases, the wall $\mathcal{P}_1$ cannot exist along with the waypoints $\{\boldsymbol{r}_n\}_{n=1}^4$, so we discard the solution (we set the weight of the particle to zero). There can be multiple reasons for this inconsistency: for example, we wrongly associate $\mathcal{P}_1$ to $\boldsymbol{r}_3$. Labeling that matches $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ with the first wall, and $\boldsymbol{r}_3$ and $\boldsymbol{r}_4$ with the second wall results in the valid case (the walls plotted with dashed lines).

In our problem, the data association refers to echo labeling, and the number of landmarks is very small compared to visual SLAM, where thousands of landmarks are observed in every frame. We therefore do not have to associate only one wall to the measured echo; we can construct several data associations for every given particle and track all the solutions.

3. Additional restrictions: There are several problem-related constraints that can reduce the set of solutions or make the particles impossible: 1) All waypoints have to be on the same side of all walls. 2) There is no wall closer to a waypoint than the wall associated to that waypoint. If one of these constraints is not satisfied, we remove the wall that violates the constraint. If this causes some waypoints to not have an associated wall anymore, we reject the particle. We illustrate these cases in Fig. 4.3.

Particles in EchoSLAM. At the core of our EchoSLAM algorithm are the P particles,

$$Y_n = \{Y_n^{[1]}, Y_n^{[2]}, \ldots, Y_n^{[P]}\}.$$

They are samples of a posterior distribution over the entire trajectory $\{\boldsymbol{r}_i\}_{i=1}^n$ and the room geometry $\mathcal{P} = \{\mathcal{P}_k\}_{k=1}^K$ up to step n. In other words, every particle represents a hypothesis of the actual state of the system at step n.

At the current step n, each particle $Y_n^{[p]}$ contains the estimated robot locations $\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n$, parameters of one or two distributions for every wall in a room, a sequence of labels of the

observed walls $\boldsymbol{c}_n^{[p]} = [c_1^{[p]}, c_2^{[p]}, \ldots, c_n^{[p]}]$, and a weight $w_n^{[p]}$. Each hypothesis has an associated *weight* or *importance factor*, which represents its likelihood to happen. Therefore, particles with large weights are near the modes of the posterior distribution, while those with small weight are near the tails [165].

Recall that the possibility of having two different probability distributions for the same wall k arises from the fact that the waypoints can be (nearly) colinear, so the lines on both sides of the trajectory have non-zero probabilities to be a wall. If the trajectory is nonlinear, the two probability distributions differ. We model the wall distributions as Gaussians with means $\mu_{k_1,n}^{[p]}$ and $\mu_{k_2,n}^{[p]} \in \mathbb{R}^2$, and covariance matrices $\Sigma_{k_1,n}^{[p]}$ and $\Sigma_{k_2,n}^{[p]} \in \mathbb{R}^{2\times 2}$. To shorten the notation, we use $\mu_{k,n}^{[p]} = \{\mu_{k_1,n}^{[p]}, \mu_{k_2,n}^{[p]}\}$ and $\Sigma_{k,n}^{[p]} = \{\Sigma_{k_1,n}^{[p]}, \Sigma_{k_2,n}^{[p]}\}$. When all lines on one side become improbable (i.e., their probability falls below a certain threshold), we discard the corresponding mean and covariance, and keep the parameters of the other distribution, $\mu_{k,n}^{[p]} = \mu_{k_j,n}^{[p]}$ and $\Sigma_{k,n}^{[p]} = \Sigma_{k_j,n}^{[p]}$, where $j = 1$ or $j = 2$. The entire structure of a particle is shown below:

$$Y_n^{[p]} = \left(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n, \mu_{1,n}^{[p]}, \Sigma_{1,n}^{[p]}, \ldots, \mu_{K,n}^{[p]}, \Sigma_{K,n}^{[p]}, \boldsymbol{c}_n^{[p]}, w_n^{[p]}\right).$$

To initialize the algorithm, we assume that all particles at $n = 1$ are equally likely, $w_1^{[p]} = 1/P$, and we label the wall that corresponds to the first measurement as 1, $\boldsymbol{c}_1^{[p]} = [1]$ for every $p = 1, \ldots, P$. We place all particles at the global coordinate system's origin, $\boldsymbol{r}_1 = [0, 0]^\top$, which fixes the translation of the estimated trajectories. To fix the rotation and reflection, we assume that the robot's local coordinate system is such that its initial orientation is aligned with the x-axis of the global coordinate system.

After every new measurement of the room impulse response, we obtain a new control $\boldsymbol{v}_n$ and measurement δ_n associated with the wall c_n. To incorporate those measurements and update the posterior over robot's waypoints and room geometry, we propose the EchoSLAM algorithm which consists of three steps that we repeat at every n:

1. prediction,
2. measurement update,
3. computation of the importance weight followed by resampling.

In the prediction step, we estimate our belief of the robot's new location using the robot's kinematics by generating particles based on the motion model. After we obtain the distance measurement from the nearest wall, for every particle and its estimate of the wall, we compute the probability of that measurement; it is proportional to the displacement of the robot's true location compared to our belief of where the robot is, and this is represented by the importance factor. In the measurement update step, we then use the measurement to update our belief of where the walls are. In the last step, we draw with replacement from the set of existing particles a set of new particles, where the probability of drawing a particle corresponds to its normalized importance weight.

We have already developed a fundamental analytical framework for the three steps in the previous sections. In Sections 4.5.3, 4.5.4 and 4.5.5 we discuss them in more detail and put the algorithm to work. We assume that the labeling is known, i.e., we know the index of the wall from which the first echo originates; in Section 4.5.6 we devise an algorithm which allows us to relax this assumption.

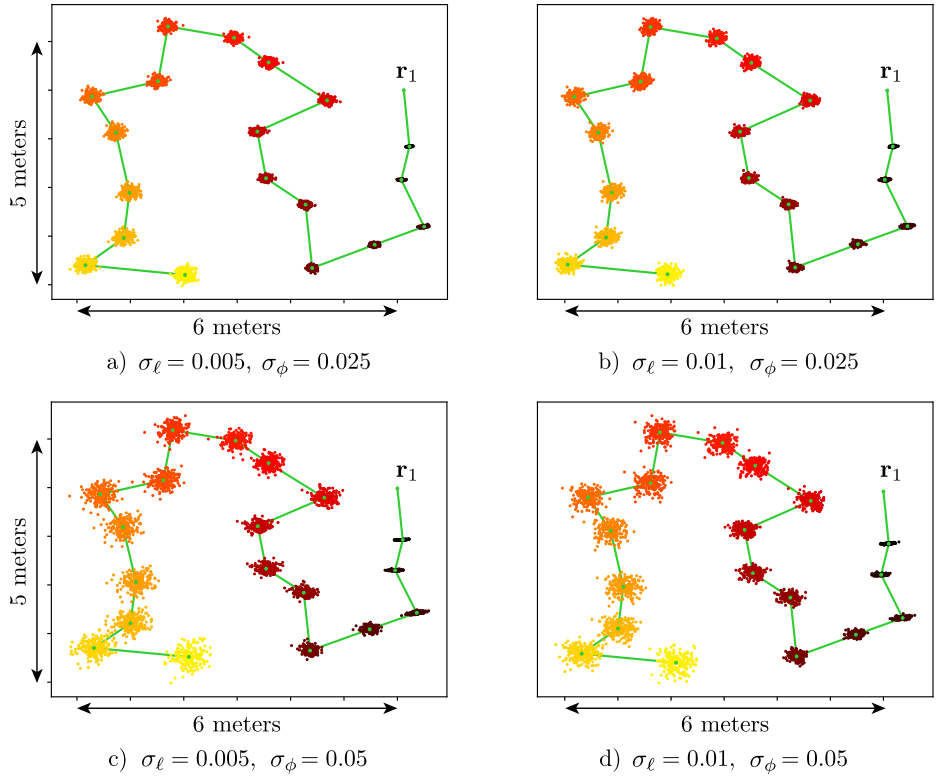


Figure 4.4: Sampling from the motion model for different noise parameters. The start location is denoted by $\boldsymbol{r}_1$, and the green line represents the robot input control actions. The 200 particles in every step approximate the robot's belief at different points in time. The values of σ_ℓ are in meters and σ_ϕ in radians.

4.5.3 Prediction

For every step n, we create an empty set of particles Y_n. To generate new locations $\boldsymbol{r}_n^{[p]}$, we sample from $p(\boldsymbol{r}_n \mid \boldsymbol{r}_{n-1}^{[p]}, \boldsymbol{v}_n)$ as follows: For a particle p in Y_{n-1}, we obtain $\hat{\ell}_n$ and $\hat{\phi}_n$ from $\hat{\ell}_n \sim \mathcal{N}(\widetilde{\ell}_n, \sigma_\ell^2)$ and $\hat{\phi}_n \sim \mathcal{N}(\widetilde{\phi}_n, \sigma_\phi^2)$, where $\widetilde{\ell}_n$ and $\widetilde{\phi}_n$ are introduced in Section 4.2.1. Then, according to the motion model from Section 4.2.1, we compute $\boldsymbol{r}_n^{[p]} = \boldsymbol{r}_{n-1}^{[p]} + \hat{\ell}_n[\cos\hat{\phi}_n, \sin\hat{\phi}_n]^\top$, and add $\boldsymbol{r}_n^{[p]}$ along with the previous waypoints $\{\boldsymbol{r}_i^{[p]}\}_{i=1}^{n-1}$ to Y_n. We repeat this for every $p = 1, \ldots, P$.

The noise levels σ_ℓ^2 and σ_ϕ^2 are in general unknown to us. In real experiments, we estimated these values by assessing the robot's precision with repetitive forward/backward motions and rotation. In simulations, we initialize σ_ℓ^2 and σ_ϕ^2 with the values estimated from these real measurements, and incrementally add more noise to test the robustness of our algorithm.

Fig. 4.4 illustrates particles obtained from the motion model with the described method. It is clear that the uncertainty grows as a non-sensing robot moves; to reduce the noise we have to exploit the sensor measurements.

4.5.4 Measurement update

Before we process a new measurement δ_n, we expand the sequence of labels $\boldsymbol{c}_{n-1}^{[p]}$ with the index of the wall that is the closest to $\boldsymbol{r}_n$. We denote it by c_n and construct a new sequence of labels by appending c_n at the end of $\boldsymbol{c}_{n-1}^{[p]}$. Then, for every particle p we add $\boldsymbol{c}_n^{[p]}$ to Y_n.

If the wall k is not the closest to the robot at step n, i.e., $k \neq c_n$, its posterior remains unchanged,

$$\left(\mu_{k,n}^{[p]}, \Sigma_{k,n}^{[p]}\right) = \left(\mu_{k,n-1}^{[p]}, \Sigma_{k,n-1}^{[p]}\right).$$

Otherwise, we incorporate the newest measurement δ_n by updating the posterior of the wall k. The update formula depends on the number of previous steps at which we have already observed an echo from the wall k. We differentiate three cases: 1) If it is the first measurement from the wall k, the procedure is trivial: we skip the measurement update and resampling (no-op), and continue with motion prediction for $n+1$. 2) If it is the second measurement, we explain in Section 4.5.4.1 how to compute the wall distributions and show that the resulting distributions can be approximated by Gaussians. 3) The case with three and more measurements is discussed in Section 4.5.4.2.

4.5.4.1 Two measurements

Let $\boldsymbol{r}_{m_k}$ and $\boldsymbol{r}_{s_k}$ be two waypoints associated to the same wall k, $c_{m_k} = c_{s_k} = k$. Indices m_k and s_k indicate the steps in which we observe the wall k for the first and the second time, in order. As the knowledge of the trajectory renders the individual wall estimates independent, we are allowed to translate and rotate waypoints associated to one wall separately from waypoints associated to another wall. Thus, without loss of generality, we can assume that we translate and rotate $\boldsymbol{r}_{m_k}$ and $\boldsymbol{r}_{s_k}$, such that for every wall k they lie at the origin and on the x-axis, respectively, while preserving their relative distance. In the following, we consider therefore that $\boldsymbol{r}_{m_k} = [0,0]^\top$ and $\boldsymbol{r}_{s_k} = [r_x, 0]^\top$, and rely on this assumption in all measurement update formulas. After the last step of the algorithm, we need to translate and rotate the estimated wall parameters $\hat{q}_k$ and $\hat{\theta}_k$ to undo the initial translation and rotation of $\boldsymbol{r}_{m_k}$ and $\boldsymbol{r}_{s_k}$ and align them with their original coordinates.

Next, recall that we use the sign of q_k to indicate if a wall intersects the y-axis above or below the origin. Together with that convention, the assumption that $\boldsymbol{r}_{m_k} = [0,0]^\top$ and $\boldsymbol{r}_{s_k} = [r_x, 0]^\top$ guarantees that the wall distributions $f_\Theta^{(1:s_k)}(\theta_k, q_k)$ after two measurements are bimodal: one mode is for the lines that are above the origin ($q_k \geq 0$), and one for the lines that are below ($q_k < 0$), see Fig. 4.5a.

To see that, observe that in the noiseless case the nearest walls to $\boldsymbol{r}_{m_k}$ and $\boldsymbol{r}_{s_k}$ are the outer common tangents to circles with centers $\boldsymbol{r}_{m_k}$ and $\boldsymbol{r}_{s_k}$ and radii δ_{m_k} and δ_{s_k} (we assume that the circles belong to non-degenerate cases, i.e., $\|\boldsymbol{r}_{s_k} - \boldsymbol{r}_{m_k}\| > \delta_{m_k} + \delta_{s_k}$ or $|\delta_{m_k} - \delta_{s_k}| \leq \|\boldsymbol{r}_{s_k} - \boldsymbol{r}_{m_k}\| < \delta_{m_k} + \delta_{s_k}$). In the presence of noise, more lines become potential walls, and the resulting distributions are centered around the outer tangents—the only possible walls in the noiseless case.

A property that the wall distribution is bimodal is essential for the algorithm as it allows us to approximate distributions with Gaussians and store only their means and covariance matrices in the particle filter. An example of a wall distribution for two waypoints that do not lie on the x-axis is illustrated in Fig. 4.5b; it is multimodal with sharp transitions around $\pm\pi/2$. Our assumption that $r_y = 0$ permits such cases to happen.

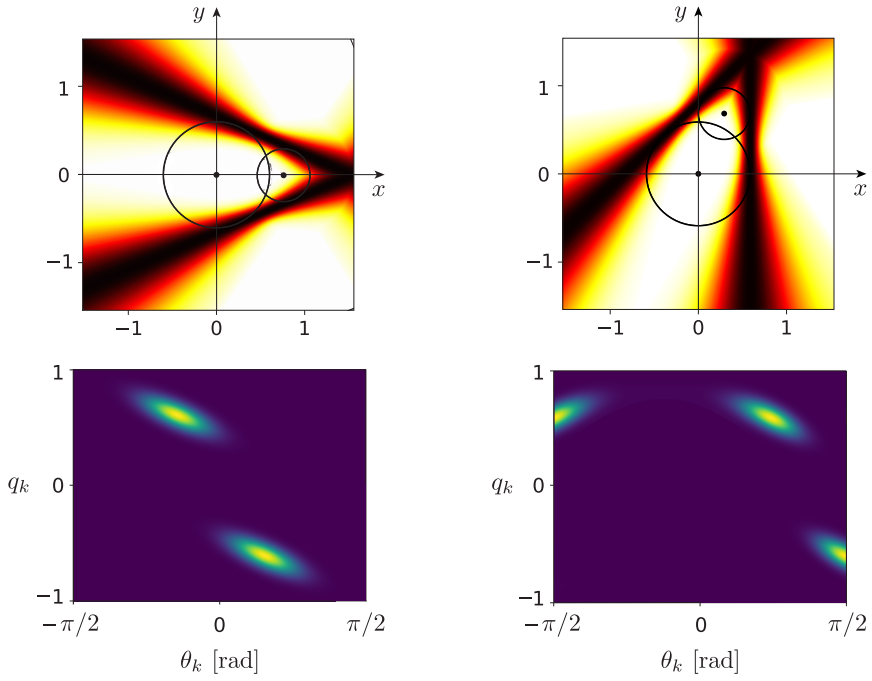


Figure 4.5: a) For $\boldsymbol{r}_1 = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$, $\boldsymbol{r}_2 = \begin{bmatrix} 0.762 & 0 \end{bmatrix}^\top$, $\delta_1 = 0.6$, $\delta_2 = 0.3$ and $\sigma_\delta = 0.1$, we illustrate possible walls with the most likely ones in black. Its distribution $f_\Theta^{(1:2)}(\theta_k, q_k)$ is bimodal, where each mode is centered around the outer tangent of two circles. b) Here, $\boldsymbol{r}_1 = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$, $\boldsymbol{r}_2 = \begin{bmatrix} 0.3 & 0.7 \end{bmatrix}^\top$, $\delta_1 = 0.6$, $\delta_2 = 0.3$ and $\sigma_\delta = 0.1$. The corresponding distribution $f_\Theta^{(1:2)}(\theta_k, q_k)$ has sharp transitions around $\pm\pi/2$, and it is multimodal. The reason is that some of the lines for which $\boldsymbol{r}_1$ is on the left side have positive and some negative intercept, as they emerge from different outer tangents by adding measurement noise.

Given two measurements δ_{m_k} and δ_{s_k}, the probability distribution of the wall k is a normalized product $(f_\Theta^{(m_k)} \cdot f_\Theta^{(s_k)})(\theta_k, q_k)$:

$$\begin{aligned} & f_\Theta^{(1:s_k)}(\theta_k, q_k) \\ & \quad = (f_\Theta^{(m_k)} \cdot f_\Theta^{(s_k)})(\theta_k, q_k) \\ & \quad = \frac{\eta}{2\pi\sigma_\delta^2}\left[\exp\left(-\frac{(q_k - \delta_{m_k})^2}{2\sigma_\delta^2}\right) + \exp\left(-\frac{(q_k + \delta_{m_k})^2}{2\sigma_\delta^2}\right)\right] \\ & \qquad \left[\exp\left(-\frac{(q_k + r_x \sin\theta_k - \delta_{s_k})^2}{2\sigma_\delta^2}\right) + \exp\left(-\frac{(q_k + r_x \sin\theta_k + \delta_{s_k})^2}{2\sigma_\delta^2}\right)\right], \end{aligned} \tag{4.6}$$

where $f_\Theta^{(m_k)}(\theta_k, q_k)$ and $f_\Theta^{(s_k)}(\theta_k, q_k)$ are computed from (4.5).

Observe that $f_\Theta^{(1:s_k)}(\theta_k, q_k)$ is a distribution with four modes, which is not in accordance with our claim that the wall distributions have at most two modes. The reason is that two modes correspond to the outer tangents of two circles with centers $\boldsymbol{r}_{m_k}$ and $\boldsymbol{r}_{s_k}$, and radii δ_{m_k} and

δ_{s_k}, while the other two modes correspond to the inner tangents of the same two circles. As all waypoints have to be inside the room, lines that cross a trajectory between two waypoints cannot be valid walls, so we set $f_\Theta^{(1:s_k)}(\theta_k, q_k)$ to 0 for the lines (θ_k, q_k) associated to the inner tangents.

To formally check if $\boldsymbol{r}_{m_k}$ and $\boldsymbol{r}_{s_k}$ are on the same side of a line through $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, directed as $\boldsymbol{x}_2 - \boldsymbol{x}_1$, we need to compute the sign of the cross products $(\boldsymbol{x}_2 - \boldsymbol{x}_1) \times (\boldsymbol{r}_k - \boldsymbol{x}_1)$ for $k = m_k$ and $k = s_k$. In our case, we fix the directions of the walls by setting

$$\boldsymbol{x}_1 = \begin{bmatrix} 0 & \frac{q_k}{\cos\theta_k} \end{bmatrix}^\top \quad \text{and} \quad \boldsymbol{x}_2 = \begin{bmatrix} x & x\tan\theta_k + \frac{q_k}{\cos\theta_k} \end{bmatrix}^\top \quad \text{with } x > 0,$$

so we obtain

$$\operatorname{sgn}(\boldsymbol{r}_{m_k}) = \operatorname{sgn}(q_k) \quad \text{and} \quad \operatorname{sgn}(\boldsymbol{r}_{s_k}) = \operatorname{sgn}\Big(r_x \tan\theta_k + \frac{q_k}{\cos\theta_k}\Big).$$

For every wall (θ_k, q_k) for which $\boldsymbol{r}_{m_k}$ and $\boldsymbol{r}_{s_k}$ are on the opposite sides of the wall, we set its probability to 0:

$$f_\Theta^{(1:s_k)}(\theta_k, q_k) = \begin{cases} 0 & \operatorname{sgn}(\boldsymbol{r}_{m_k}) \neq \operatorname{sgn}(\boldsymbol{r}_{s_k}), \\ f_\Theta^{(1:s_k)}(\theta_k, q_k) & \text{otherwise.} \end{cases} \tag{4.7}$$

In conclusion of the above analysis, the normalized distribution $f_\Theta^{(1:s_k)}(\theta_k, q_k)$ computed by (4.6) and corrected by (4.7) reveals probabilities of lines to be a wall k, given two distances δ_{m_k} and δ_{s_k} measured at the waypoints $\mathbf{0}$ and $[r_x, 0]^\top$.

Gaussian approximation. We argue that $f_\Theta^{(1:s_k)}(\theta_k, q_k)$ can be well approximated with a Gaussian mixture. To back up our claim, we show that it is practically impossible to distinguish between the original distribution and its approximation, unless the noise is unrealistically large. The main advantage of using Gaussian approximations instead of the original $f_\Theta^{(1:s_k)}(\theta_k, q_k)$ is that it allows us to store only their means and covariance matrices in the particle filter, which greatly reduces the space and time complexity.

To shorten the notation in this section, we set $m_k = 1$ and $s_k = 2$. We show that the probability distribution of a wall $f_\Phi^{(1:2)}(\phi_k, q_k)$, obtained from $f_\Theta^{(1:2)}(\theta_k, q_k)$ by the change of variable $\phi_k = \sin\theta_k$, is a Gaussian mixture with two components. We apply $\phi_k = \sin\theta_k$ and rewrite (4.6) as

$$\begin{aligned} f_\Phi^{(1:2)}(\phi_k, q_k) = \frac{\eta}{2\pi\sigma_\delta^2} \Bigg[& \\ \exp\left(-\frac{(q_k-\delta_1)^2 + (q_k + r_x\phi_k - \delta_2)^2}{2\sigma_\delta^2}\right) + \exp\left(-\frac{(q_k-\delta_1)^2 + (q_k + r_x\phi_k + \delta_2)^2}{2\sigma_\delta^2}\right) & \\ + \exp\left(-\frac{(q_k+\delta_1)^2 + (q_k + r_x\phi_k - \delta_2)^2}{2\sigma_\delta^2}\right) + \exp\left(-\frac{(q_k+\delta_1)^2 + (q_k + r_x\phi_k + \delta_2)^2}{2\sigma_\delta^2}\right) \Bigg]. & \end{aligned} \tag{4.8}$$

The arguments of the exponential functions in the first two terms of (4.8) can be split into partial fractions as:

$$\frac{(q_k-\delta_1)^2 + (q_k + r_x\phi_k \pm \delta_2)^2}{2\sigma_\delta^2} = \frac{(q_k-\delta_1)^2}{\sigma_\delta^2} + \frac{\big(\phi_k + \frac{\delta_1 \pm \delta_2}{r_x}\big)^2}{\frac{2}{r_x^2}\sigma_\delta^2} + \frac{(q_k-\delta_1)\big(\phi_k + \frac{\delta_1 \pm \delta_2}{r_x}\big)}{\frac{1}{r_x}\sigma_\delta^2}. \tag{4.9}$$

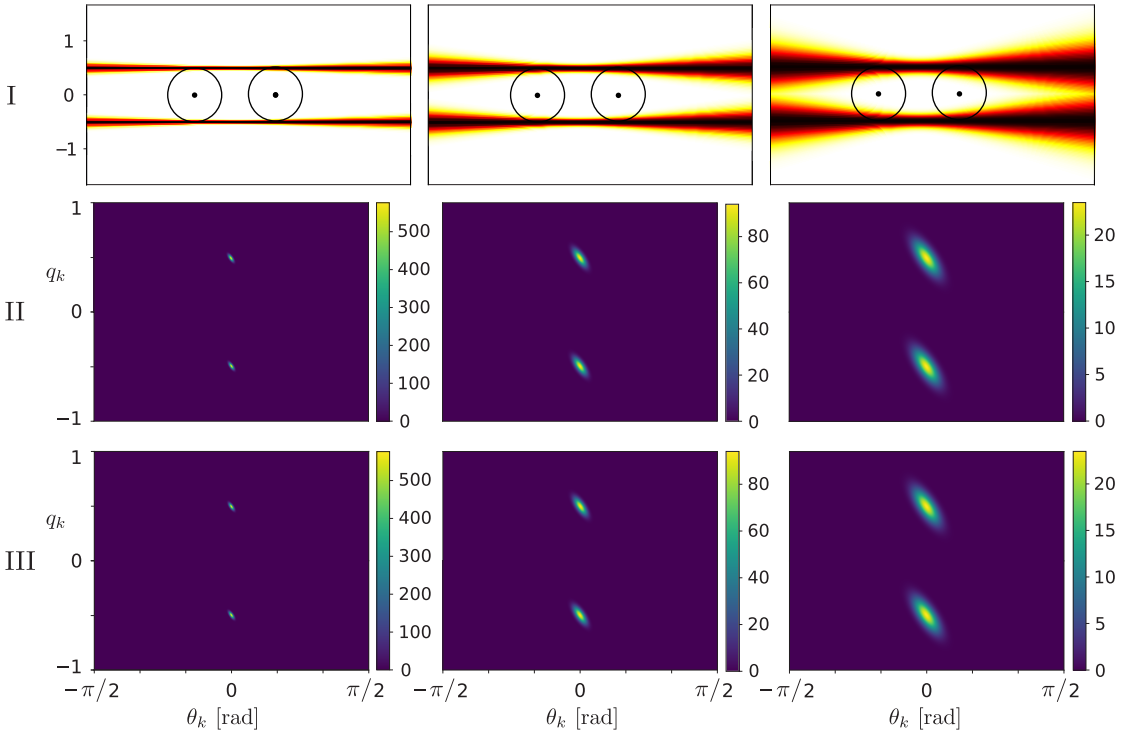


Figure 4.6: I: For a given pair of waypoints $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, and measurements $\delta_1 = \delta_2$, we illustrate the probabilities of lines to be the nearest wall to $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$. Darker colors indicate higher values. PDFs $f_\Theta^{(1:2)}(\theta_k, q_k)$ are shown in II and their approximations by Gaussian distributions in III. As the most likely lines are centered around $\theta_k = 0$, the approximation is more accurate than for any other value of $\theta_k \in (0, \pi)$.

Since a bivariate normal probability density function (PDF) is defined as

$$g(\phi_k, q_k) = \frac{1}{2\pi\sigma_\Phi\sigma_Q\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}\left(\frac{(\phi_k-\mu_\Phi)^2}{\sigma_\Phi^2} + \frac{(q_k-\mu_Q)^2}{\sigma_Q^2} - \frac{2\rho(\Phi-\mu_\Phi)(q_k-\mu_Q)}{\sigma_\Phi\sigma_Q}\right)\right],$$

by making the following substitutions in (4.9)

$$\mu_\Phi = \delta_1, \quad \sigma_\Phi = \sigma_\delta, \quad \mu_Q = \frac{\delta_1 \mp \delta_2}{r_x}, \quad \sigma_Q = -\frac{\sqrt{2}}{r_x}\sigma_\delta, \quad \rho = \frac{\sqrt{2}}{2},$$

we prove that the first two terms of (4.8) are bivariate Gaussian functions. The proof is analogous for the last two terms.

As observed earlier in this section, two of these terms correspond to the outer tangents of the two circles, while the other two terms correspond to the inner tangents, so we can discard them. We conclude that the probability distribution of a wall $f_\Phi^{(1:2)}(\phi_k, q_k)$ is a Gaussian mixture with two components: one component is the PDF of lines that are above the origin, and the other component is the PDF of lines below the origin. Adding a third measurement from the same wall often resolves this ambiguity.

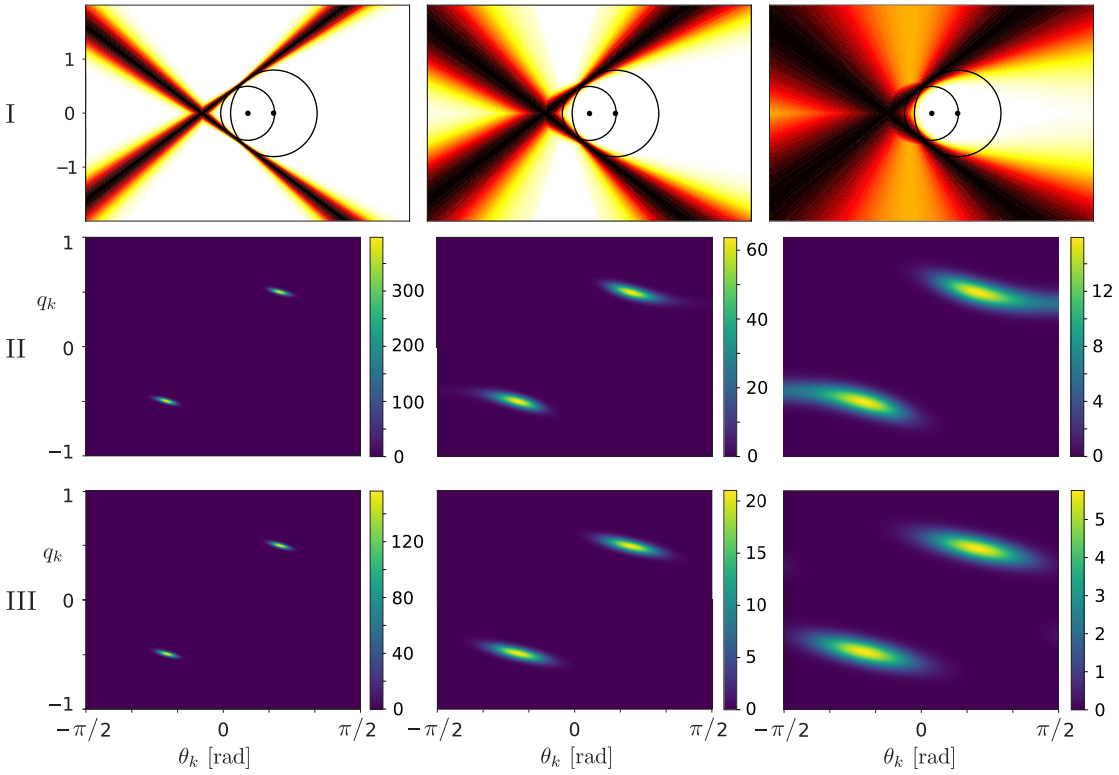


Figure 4.7: Plots represent probabilities of lines to be a wall, as in Fig. 4.6. The difference is that $\delta_2 > \delta_1$, which induces subtle discrepancy between the original distributions and their Gaussian approximations. However, this becomes noticeable only for larger values of σ_δ.

By reverting to the original variables, this result implies that $f_\Theta^{(1:2)}(\theta_k, q_k)$ is well approximated with a Gaussian shape for small values of θ_k, as we can apply the small-angle approximation $\phi_k = \sin\theta_k \approx \theta_k$. As θ_k grows and approaches $\pi/2$, the approximation becomes less accurate. Two examples are illustrated in Fig. 4.6 and Fig. 4.7. The second row of figures presents the original distributions $f_\Theta^{(1:2)}(\theta_k, q_k)$, while the third row shows their Gaussian approximations. As the angles of the most probable lines in Fig. 4.6 are close to zero ($\delta_1 = \delta_2$), the original distributions are accurately approximated by Gaussians. In Fig. 4.7 we can notice the difference between the original distributions and their approximations, in particular for larger values of the noise level σ_δ. Based on our experimental evaluation of the robot accuracy, σ_δ does not exceed the value of 5 cm, which is used in the implementation of our algorithm. Therefore, the Gaussian approximation does not degrade the performance of our algorithm for different values of θ_k, assuming practical values of $\sigma_\delta \leq 5$ cm.

4.5.4.2 Adding new measurements

Unlike FastSLAM, we cannot take advantage of the Gaussian approximation of $p(\mathcal{P}_k \mid \{\boldsymbol{r}_i\}_{i=1}^{n-1}, \{\delta_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1})$ to efficiently compute $p(\mathcal{P}_k \mid \{\boldsymbol{r}_i\}_{i=1}^{n}, \{\delta_i\}_{i=1}^{n}, \{c_i\}_{i=1}^{n})$. The reason is that

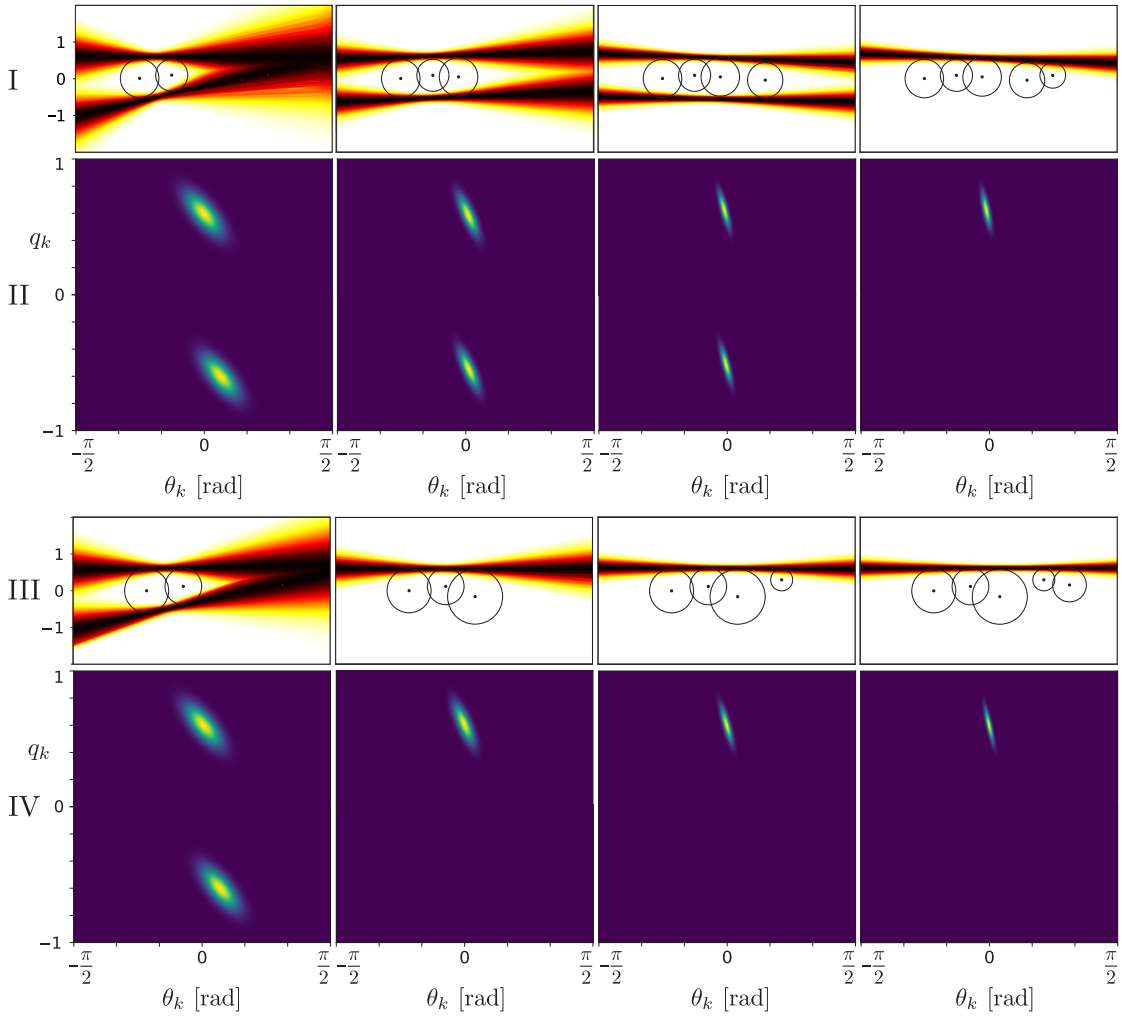


Figure 4.8: Plots I and III illustrate probabilities of lines to be a wall for steps from $n = 2$ to $n = 5$, where darker colors correspond to higher values. II and IV depict PDFs $f_\Theta^{(1:n)}(\theta_k, q_k)$. The waypoints $\{\boldsymbol{r}_n\}_{n=1}^4$ in I are almost linear, resulting in two non-zero PDFs, one for each side of the trajectory. At $n = 5$, the waypoint $\boldsymbol{r}_5$ and the measurement δ_5 are such that the probability of all lines with $q_k < 0$ becomes 0. In III this happens at $n = 3$, so the wall cannot have a negative intercept. Also, we can observe that every additional measurement makes the PDF of lines with $q_k \geq 0$ narrower.

when a measurement is of lower dimensionality than the coordinates of a landmark, it is impossible to use the standard EKF measurement update [168]. As this is the case for our distance measurement δ_n and the wall coordinates (θ_k, q_k), we cannot profit from the computational benefits of EKFs; instead, we compute the posterior $f_\Theta^{(1:n-1)}(\theta_k, q_k)$ from its Gaussian parameters stored in the particle filter and combine it with $f_\Theta^{(n)}(\theta_k, q_k)$ from (4.5) to obtain a new posterior

$f_\Theta^{(1:n)}(\theta_k, q_k)$. The details are provided below.

In the previous section we derived the distribution $f_\Theta^{(1:n-1)}(\theta_k, q_k)$ for two measurements from the same wall. It is bimodal and centered around the outer tangents. As each mode is associated to lines that are on one side of the waypoints, adding new measurements renders one of them less likely than the other, until no walls on that side are possible (unless the robot moves on a line).

Here we propose an algorithm to update $f_\Theta^{(1:n-1)}(\theta_k, q_k)$ with a new measurement. As the first two waypoints associated to the same wall k lie in the origin and on the x-axis, respectively, $f_\Theta^{(n)}(\theta_k, q_k)$ needs to be computed for a translated and rotated $\boldsymbol{r}_n$, such that its relative position with respect to the first two waypoints does not change.

We translate $\boldsymbol{r}_n$ by $-\boldsymbol{r}_{m_k}$ and rotate the new point $\boldsymbol{r}_n - \boldsymbol{r}_{m_k}$ by $-\arctan \frac{\boldsymbol{r}_{s_k,y} - \boldsymbol{r}_{m_k,y}}{\boldsymbol{r}_{s_k,x} - \boldsymbol{r}_{m_k,x}}$ around the origin. We denote it by $\boldsymbol{r}'_n$. Then, for the given measurement δ_n and $\boldsymbol{r}'_n$, we compute the wall distribution $f_\Theta^{(n)}(\theta_k, q_k)$ from (4.5). We incorporate the new measurement by multiplying the estimated distribution with the previous posterior $f_\Theta^{(1:n-1)}(\theta_k, q_k)$,

$$f_\Theta^{(1:n)}(\theta_k, q_k) = \eta f_\Theta^{(1:n-1)}(\theta_k, q_k) f_\Theta^{(n)}(\theta_k, q_k). \tag{4.10}$$

We finally approximate $f_\Theta^{(1:n)}(\theta_k, q_k)$ by a normal distribution, and add its mean and covariance matrix to the particle filter Y_n. Examples of wall distributions after five steps are shown in Fig. 4.8.

4.5.5 Resampling

Resampling is a common technique in particle filtering to correct for a mismatch between the proposal and the target distribution. The proposal distribution is the one at which particles are generated,

$$\begin{aligned} &p(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n \mid \{\delta_i\}_{i=1}^{n-1}, \{\boldsymbol{v}_i\}_{i=2}^n, \{c_i\}_{i=1}^{n-1}) \\ &\qquad = p(\boldsymbol{r}_n^{[p]} \mid \boldsymbol{r}_{n-1}^{[p]}, \boldsymbol{v}_n) p(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^{n-1} \mid \{\delta_i\}_{i=1}^{n-1}, \{\boldsymbol{v}_i\}_{i=2}^{n-1}, \{c_i\}_{i=1}^{n-1}), \end{aligned}$$

while the target distribution takes into account the newest measurement δ_n,

$$p(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n \mid \{\delta_i\}_{i=1}^n, \{\boldsymbol{v}_i\}_{i=2}^n, \{c_i\}_{i=1}^n).$$

The weight or the importance factor $w_n^{[p]}$ is the quotient of the target and the proposal distribution and it corresponds to the probability of the measurement δ_n given the particle $\boldsymbol{r}_n^{[p]}$:

$$\begin{aligned} w_n^{[p]} &\approx \frac{p(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n \mid \{\delta_i\}_{i=1}^n, \{\boldsymbol{v}_i\}_{i=2}^n, \{c_i\}_{i=1}^n)}{p(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n \mid \{\delta_i\}_{i=1}^{n-1}, \{\boldsymbol{v}_i\}_{i=2}^n, \{c_i\}_{i=1}^{n-1})} \\ &\approx \frac{p(\delta_n \mid \{\boldsymbol{r}_i^{[p]}\}_{i=1}^n, \{\delta_i\}_{i=1}^{n-1}, \{\boldsymbol{v}_i^{[p]}\}_{i=2}^n, \{c_i\}_{i=1}^n) p(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n \mid \{\delta_i\}_{i=1}^{n-1}, \{\boldsymbol{v}_i^{[p]}\}_{i=2}^n, \{c_i\}_{i=1}^n)}{p(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n \mid \{\delta_i\}_{i=1}^{n-1}, \{\boldsymbol{v}_i\}_{i=2}^n, \{c_i\}_{i=1}^{n-1})} \\ &\approx \frac{p(\delta_n \mid \boldsymbol{r}_n^{[p]}, c_n) p(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n \mid \{\delta_i\}_{i=1}^{n-1}, \{\boldsymbol{v}_i^{[p]}\}_{i=2}^n, \{c_i\}_{i=1}^{n-1})}{p(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n \mid \{\delta_i\}_{i=1}^{n-1}, \{\boldsymbol{v}_i\}_{i=2}^n, \{c_i\}_{i=1}^{n-1})} \\ &= \eta p(\delta_n \mid \boldsymbol{r}_n^{[p]}, c_n). \end{aligned}$$

To calculate $w_n^{[p]}$, we need to further transform $p(\delta_n \mid \boldsymbol{r}_n^{[p]}, c_n)$, and obtain:

$$w_n^{[p]} = \eta \sum_{\mathcal{P}_{c_n}} p(\delta_n \mid \mathcal{P}_{c_n}, \boldsymbol{r}_n^{[p]}, c_n) p(\mathcal{P}_{c_n} \mid \{\boldsymbol{r}_i^{[p]}\}_{i=1}^{n-1}, \{\delta_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1}), \tag{4.11}$$

where $p(\mathcal{P}_{c_n} \mid \{\boldsymbol{r}_i^{[p]}\}_{i=1}^{n-1}, \{\delta_i\}_{i=1}^{n-1}\{c_i\}_{i=1}^{n-1})$ is known from the previous step,

$$p(\mathcal{P}_{c_n} \mid \{\boldsymbol{r}_i^{[p]}\}_{i=1}^{n-1}, \{\delta_i\}_{i=1}^{n-1}\{c_i\}_{i=1}^{n-1}) \sim \mathcal{N}(\mu_{c_n,n-1}^{[p]}, \Sigma_{c_n,n-1}^{[p]}),$$

while $p(\delta_n \mid \mathcal{P}_{c_n}, \boldsymbol{r}_n^{[p]}, c_n)$ can be computed from (4.2) with $d_{nk} = \text{dist}(\boldsymbol{r}_n^{[p]}, \mathcal{P}_{c_n})$. By drawing with replacement P particles from Y_n according to the weights $w_n^{[p]}$, we approximate the desired target distribution.

After the resampling step, the robot is ready to receive new control commands, move, and record a RIR, while our EchoSLAM algorithm accordingly iterates over the prediction, measurement update and resampling steps. The algorithm stops after the robot has visited N locations and measured N RIRs. The final estimates of the walls and robot's waypoints are then simply taken from the particle in Y_N with the largest weight, denoted by $\hat{p}$. The robot's waypoints are $\{\boldsymbol{r}_n^{[\hat{p}]}\}_{n=1}^N$, while the parameters of the walls are obtained from $[\hat{\theta}_k, \hat{q}_k] = \mu_{k,N}^{[\hat{p}]}$. Recall however that $\hat{\theta}_k$ and $\hat{q}_k$ correspond to the walls estimated for the rotated and translated waypoints of the trajectory, as discussed in Section 4.5.4.1. To obtain the correct values, we need to rotate $\hat{\theta}_k$ such that the first two waypoints associated to k recover their original orientation. Observe that by rotating waypoints around the origin for some angle β_k, the corresponding distribution $f_\Theta^{(1:N)}(\theta_k, q_k)$ only circulary shifts around θ_k for the same angle β_k. In our case, $\beta_k = \arctan \frac{\boldsymbol{r}_{s_k,y} - \boldsymbol{r}_{m_k,y}}{\boldsymbol{r}_{s_k,x} - \boldsymbol{r}_{m_k,x}}$. To translate the rotated waypoints and align them with their original coordinates, we need to adjust the estimated value of $\hat{q}_k$; this is done by simple trigonometry.

This procedure concludes our discussion of the EchoSLAM algorithm with known labeling: we have found the trajectory of a robot and the walls of a room that are the most likely to generate the measured distances. Its summary is provided in Algorithm 4.1.

4.5.6 Unknown echo labeling

Given a recording of a room impulse response, we can easily extract the first echo and measure its propagation time to obtain the distance of the robot from the nearest wall. But for the EchoSLAM algorithm to work, we also need to know from which wall the detected echo originates. The problem of determining the correct mapping of measurements to landmarks is commonly referred to as the data association or correspondence problem; we also use the more specific term of *echo labeling*.

Echo labeling. Echo labeling is performed on a per-particle basis as different particles can observe different walls at the same step n. The main difference between our setup and standard SLAM problems is that the number of walls is very small in comparison to the number of landmarks typically observed in the map. Instead of choosing only one (the most likely) label for every particle at every n, we propose a branch and prune algorithm that builds a tree of possible label sequences for every particle.

Let us consider the tree corresponding to the particle $Y_n^{[p]}$. The number of walls at step n of the algorithm is denoted by K_n, $K_n = \max \boldsymbol{c}_n^{[p]}$. As long as we can provide K_{max} such that $K_{max} \geq K$, the exact number of walls K can be unknown. Every node of depth n in the tree

Algorithm 4.1 The main building block of EchoSLAM with the given labeling

Input: δ_n, c_n, $\boldsymbol{v}_n$, Y_{n-1}

$Y_n = \emptyset$
for $p = 1$ to P **do**
  $\boldsymbol{r}_n^{[p]} \sim p(\boldsymbol{r}_n \mid \boldsymbol{r}_{n-1}^{[p]}, \boldsymbol{v}_n)$
  create $\boldsymbol{c}_n^{[p]}$ by appending c_n to $\boldsymbol{c}_{n-1}^{[p]}$
  for $k = 1$ to K, such that $k \neq c_n$ **do**
    $\left(\mu_{k,n}^{[p]}, \Sigma_{k,n}^{[p]}\right) = \left(\mu_{k,n-1}^{[p]}, \Sigma_{k,n-1}^{[p]}\right)$
  end for
  $k = c_n$
  if wall k observed for the first time **then**
    no-op
  else if wall k observed for the second time **then**
    translate $\boldsymbol{r}_{m_k}$ and $\boldsymbol{r}_{s_k}$ by $-\boldsymbol{r}_{m_k}$
    rotate $\boldsymbol{r}_{s_k} - \boldsymbol{r}_{m_k}$ by $-\arctan \frac{\boldsymbol{r}_{s_k,y} - \boldsymbol{r}_{m_k,y}}{\boldsymbol{r}_{s_k,x} - \boldsymbol{r}_{m_k,x}}$
    initialize the posterior $f_\Theta^{(1:s_k)}(\theta_k, q_k)$ with (4.6)
    compute $\mu_{k,n}^{[p]}$ and $\Sigma_{k,n}^{[p]}$ of its Gaussian approximation
  else
    translate $\boldsymbol{r}_n$ by $-\boldsymbol{r}_{m_k}$
    rotate $\boldsymbol{r}_n - \boldsymbol{r}_{m_k}$ by $-\arctan \frac{\boldsymbol{r}_{s_k,y} - \boldsymbol{r}_{m_k,y}}{\boldsymbol{r}_{s_k,x} - \boldsymbol{r}_{m_k,x}}$ to obtain $\boldsymbol{r}'_n$
    compute $f_\Theta^{(n)}(\theta_k, q_k)$ by (4.5) where $\boldsymbol{r}_n$ is replaced with $\boldsymbol{r}'_n$
    update the posterior $f_\Theta^{(1:n)}(\theta_k, q_k)$ by (4.10)
    compute $\mu_{k,n}^{[p]}$ and $\Sigma_{k,n}^{[p]}$ of its Gaussian approximation
  end if
  if $\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n$ on the same side of all walls and $\mathrm{dist}(\boldsymbol{r}_n^{[p]}, \mathcal{P}_{c_n}) \leq \mathrm{dist}(\boldsymbol{r}_n^{[p]}, \mathcal{P}_j)$ for all $j = 1, \ldots, K$, $j \neq c_n$ **then**
    compute $w_n^{[p]}$ by (4.11)
  else
    $w_n^{[p]} = 0$
  end if
  $Y_n^{[p]} = \left(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n, \mu_{1,n}^{[p]}, \Sigma_{1,n}^{[p]}, \ldots, \mu_{K,n}^{[p]}, \Sigma_{K,n}^{[p]}, \boldsymbol{c}_n^{[p]}, w_n^{[p]}\right)$
  add $Y_n^{[p]}$ to Y_n
end for
$Y'_n = \emptyset$
repeat
  draw $Y_n^{[p]}$ with probability $w_n^{[p]}$ and add it to Y'_n
until $\mathrm{length}(Y'_n) = P$
set the weight of all particles in Y'_n to $1/P$
return Y'_n

represents a sequence of labels that could have given rise to the measured distances $\delta_1, \delta_2, \ldots, \delta_n$, based on the probabilistic criteria described below. The reason why we create a tree of labels instead of selecting only one path at every n is that for a given sequence $\boldsymbol{c}_{n-1}$, we can sometimes

associate several different labels to δ_n, where all of them yield a valid solution at n. Expanding $\boldsymbol{c}_{n-1}$ with different labels is akin to creating several child nodes of the node $\boldsymbol{c}_{n-1}$. It is clear that we do not have to create a child node for every label in every node; to decide which labels are feasible for some chosen node at level $n-1$, we do the following. First, among all existing wall estimates $k = 1, \ldots, K_{n-1}$ at step $n-1$, we find the one that maximizes the probability $p(\delta_n \mid \mathcal{P}_k, \boldsymbol{r}_n^{[p]}, k)$ of measuring δ_n, and we denote it by c_n. If there are two distributions associated to one wall, we select the one with the higher probability $j = \arg\max_{j \in \{1,2\}} p(\delta_n \mid \mathcal{P}_{k_j}, \boldsymbol{r}_n^{[p]}, k_j)$. Then, by using $\boldsymbol{c}_n^{[p]} := [\boldsymbol{c}_{n-1}^{[p]}, k]$ to denote the operation of adding a label k at the end of the current sequence of labels $\boldsymbol{c}_{n-1}^{[p]}$, we compute $\boldsymbol{c}_n^{[p]}$ given $\boldsymbol{c}_n^{[p-1]}$ based on the following threshold-based criteria:

- If $p(\delta_n \mid \mathcal{P}_{c_n}, \boldsymbol{r}_n^{[p]}, c_n) > \tau_m$, we create a single child node of the current node and label it with $\boldsymbol{c}_n^{[p]} := [\boldsymbol{c}_{n-1}^{[p]}, c_n]$.
- If $p(\delta_n \mid \mathcal{P}_k, \boldsymbol{r}_n^{[p]}, k) \leq \tau_m$ for every $k = 1, \ldots, K_{n-1}$, we might have a measurement from a yet-unobserved wall. Hence, we increase the number of walls by one $K_n = K_{n-1} + 1$, create a single child node of the current node, and label it with $\boldsymbol{c}_n^{[p]} := [\boldsymbol{c}_{n-1}^{[p]}, K_n]$.
- If $p(\delta_n \mid \mathcal{P}_k, \boldsymbol{r}_n^{[p]}, k) \leq \tau_m$ for every $k = 1, \ldots, K_{n-1}$, we create a child node for every k that appears exactly once in $\boldsymbol{c}_{n-1}^{[p]}$, and label it with $\boldsymbol{c}_n^{[p]} := [\boldsymbol{c}_{n-1}^{[p]}, k]$.
- For every $1 \leq k \leq K_{n-1}$ for which $\tau_\ell < p(\delta_n \mid \mathcal{P}_k, \boldsymbol{r}_n^{[p]}, k) \leq \tau_m$, we create a child node and label it with $\boldsymbol{c}_n^{[p]} := [\boldsymbol{c}_{n-1}^{[p]}, k]$.

We set a high value for τ_m to ensure that we match only one wall k with the measurement δ_n when the confidence about their correspondence is high. However, as the measurement noise affects estimated wall positions $\mathcal{P}_k$ and the estimated robot location $\boldsymbol{r}_n^{[p]}$, the value of $p(\delta_n \mid \mathcal{P}_{c_n}, \boldsymbol{r}_n^{[p]}, c_n)$ can fall below the threshold τ_m even for the correct labeling. To reduce the possibility of losing a valid solution because of being too restrictive in pruning the branches, for every wall k that satisfies $\tau_\ell < p(\delta_n \mid \mathcal{P}_k, \boldsymbol{r}_n^{[p]}, k) \leq \tau_m$, we also create a branch. Moreover, we have to create branches for all walls that have been observed only once, as we have no information about them. We create new labels only when all existing walls are unlikely to be the correct matches.

We store the sequences of labels of all child nodes in a list $\boldsymbol{C}_n^{[p]}$, and we branch by choosing the most likely child node. If that choice does not lead to a feasible solution, we step back up the tree and choose the next most likely sequence of labels. The solution tree is therefore traversed depth-first. We give priority to the label k with the highest value of $p(\delta_n \mid \mathcal{P}_k, \boldsymbol{r}_n^{[p]}, k)$, and furthermore, we give priorities to the labels associated to the existing walls over a new label. In terms of implementation, we represent a tree as a list, so we simply append the sequences of labels with higher priority at the end. When branching, we always select the last added sequence from the list.

The algorithm constructs a tree for every particle p independently: it returns a list of possible sequences of labels at step n, and the current number of walls K_n. We summarize these steps in Algorithm 4.2 and illustrate it with an example in Fig. 4.9.

EchoSLAM with unknown labeling. To provide a complete solution to the EchoSLAM algorithm with unknown echo labeling, we need to merge Algorithm 4.2 with Algorithm 4.1. For

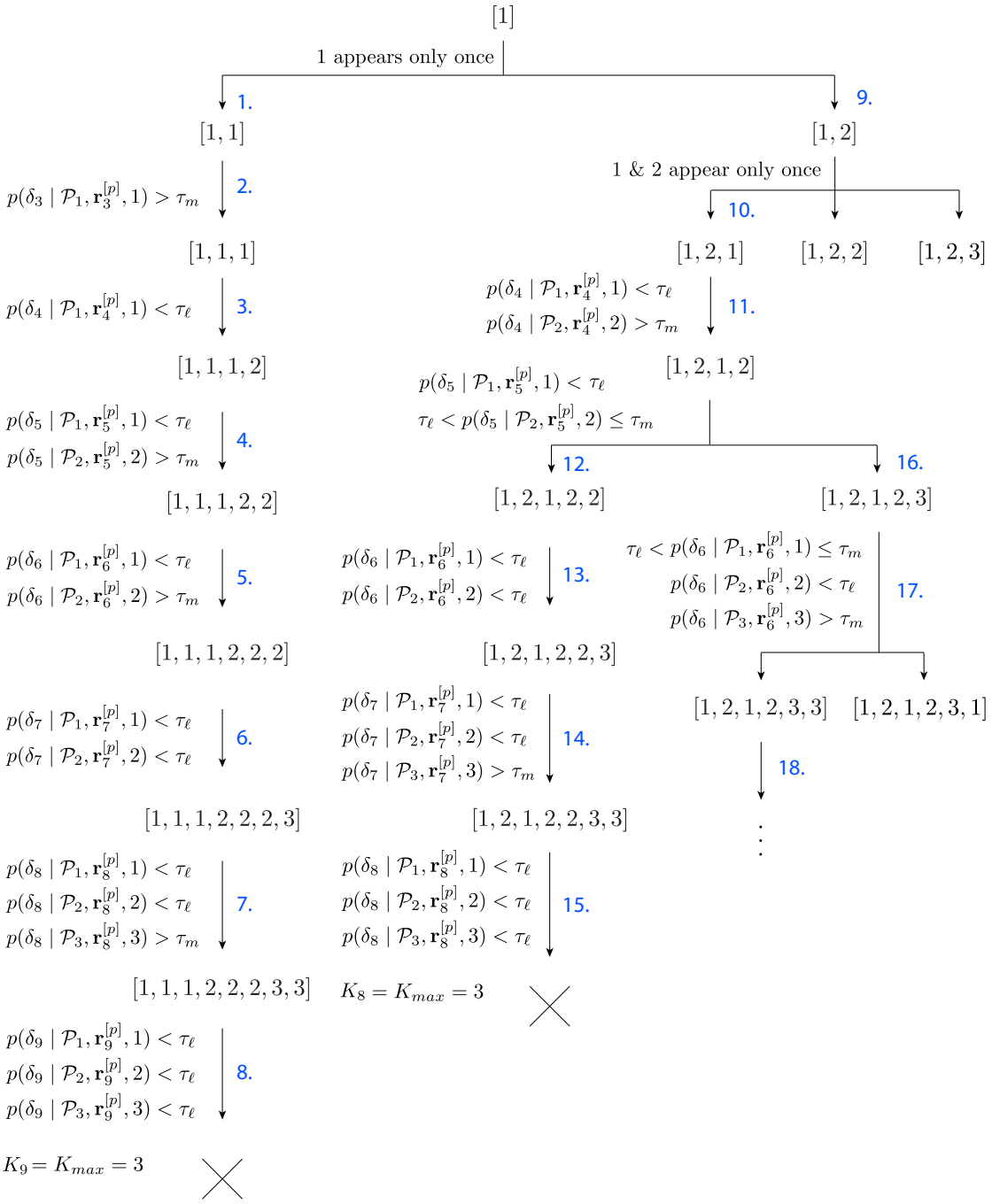


Figure 4.9: An example of a tree for an echo labeling problem with $K_{max} = 3$. The blue numbers indicate the order of created nodes. The branching conditions that are satisfied in every node are listed next to the arrows.

Algorithm 4.2 Echo labeling

K_{max} is a global variable

Input: $K_{n-1}, Y_{n-1}^{[p]}, \boldsymbol{r}_n^{[p]}, \delta_n$

$p_{max} = 0$
for $k = 1$ to K_{n-1} **do**
  if $Y_{n-1}^{[p]}$ contains two wall distributions for k **then**
    $p(\delta_n \mid \mathcal{P}_k, \boldsymbol{r}_n^{[p]}, k) = \max\{p(\delta_n \mid \mathcal{P}_{k_1}, \boldsymbol{r}_n^{[p]}, k_1),\ p(\delta_n \mid \mathcal{P}_{k_2}, \boldsymbol{r}_n^{[p]}, k_2)\}$
  end if
  if $p(\delta_n \mid \mathcal{P}_k, \boldsymbol{r}_n^{[p]}, k) > p_{max}$ **then**
    $p_{max} = p(\delta_n \mid \mathcal{P}_k, \boldsymbol{r}_n^{[p]}, k)$
    $c_n = k$
  end if
end for
if $p_{max} > \tau_m$ **then**
  create $\boldsymbol{c}_n^{[p]}$ by appending c_n to $\boldsymbol{c}_{n-1}^{[p]}$
  return $[\boldsymbol{c}_n^{[p]}], K_{n-1}$
else
  $\boldsymbol{C}_n^{[p]} = \emptyset$
  $K_n = K_{n-1} + 1$
  create $\boldsymbol{c}_n^{[p]}$ by appending K_n to $\boldsymbol{c}_{n-1}^{[p]}$
  add $\boldsymbol{c}_n^{[p]}$ to $\boldsymbol{C}_n^{[p]}$
  for $k = 1$ to K_{n-1} **do**
    if $p(\delta_n \mid \mathcal{P}_k, \boldsymbol{r}_n^{[p]}, k) > \tau_\ell$ or $\left(\sum_{c_i \in \boldsymbol{c}_{n-1}^{[p]}} \mathbf{1}_{(c_i == k)}\right) = 1$ **then**
      create $\boldsymbol{c}_n^{[p]}$ by appending k to $\boldsymbol{c}_{n-1}^{[p]}$
      add $\boldsymbol{c}_n^{[p]}$ to $\boldsymbol{C}_n^{[p]}$
    end if
  end for
  return $\boldsymbol{C}_n^{[p]}, K_n$
end if

some chosen node $\boldsymbol{c}_{n-1}$, Algorithm 4.2 returns a sequence of labels $\boldsymbol{C}_n^{[p]}$ of all child nodes that are feasible for the given measurement δ_n and particle $Y_{n-1}^{[p]}$. By putting the sequences of all particles $p = 1, \ldots, P$ without repetition in a new list, we obtain $\boldsymbol{C}_n$. Then we introduce a *global labeling tree* Υ, which represents a union of the labeling trees of all particles. Given δ_n and $\{Y_{n-1}^{[p]}\}_{p=1}^{P}$, the node corresponding to $\boldsymbol{c}_{n-1}$ in Υ has one child node for every sequence in $\boldsymbol{C}_n$. At $n = 1$, Υ has only one node and its associated sequence of labels is equal to [1].

We propose to create copies of particle sets in every node. More precisely, when we create a child node, we create a copy of the particle set of its parent, and extend the old sequence of labels with a new label. During the measurement update and resampling, we update the parameters of the walls and the trajectory for every child independently. By doing so, we store the states of the estimates in every node, which allows us to traverse the tree depth-first and return to nodes at lower depths. We use $\boldsymbol{c}_n$ as a superscript for Y_n to uniquely identify particle sets of the same depth but of different label sequences, $Y_n^{\boldsymbol{c}_n}$. The complete EchoSLAM with unknown echo labeling is stated in Algorithm 4.3, with the main functionality in Algorithm 4.4.

Algorithm 4.3 The main building block of EchoSLAM with unknown labeling

Labeling tree Υ and particle sets $Y_n^{\boldsymbol{c}_n}$ are global variables

Input: δ_n, $\boldsymbol{v}_n$, $\boldsymbol{c}_{n-1}$

1: $Y_{n-1} := Y_{n-1}^{\boldsymbol{c}_{n-1}}$ (*)
2: $\boldsymbol{C}_n = \emptyset$
3: **for** $p = 1$ to P **do**
4: $\quad \boldsymbol{r}_n^{[p]} \sim p(\boldsymbol{r}_n \mid \boldsymbol{r}_{n-1}^{[p]}, \boldsymbol{v}_n)$
5: $\quad$ invoke Algorithm 4.2 and assign the returned values to $\boldsymbol{C}_n^{[p]}$ and K_n
6: $\quad$ **for** $\boldsymbol{c}_i \in \boldsymbol{C}_n^{[p]}$ **do**
7: $\quad\quad$ **if** $\boldsymbol{c}_i \notin \boldsymbol{C}_n$ **then**
8: $\quad\quad\quad$ add $\boldsymbol{c}_i$ to $\boldsymbol{C}_n$
9: $\quad\quad\quad$ $Y_n^{\boldsymbol{c}_i} = \emptyset$
10: $\quad\quad\quad$ **for** $p = 1$ to P **do**
11: $\quad\quad\quad\quad$ copy $Y_{n-1}^{[p]}$ to $Y_n^{\boldsymbol{c}_i,[p]}$, and add new $\boldsymbol{r}_n^{[p]}$ and $\boldsymbol{c}_i$:

$$Y_n^{\boldsymbol{c}_i,[p]} = \left(\{\boldsymbol{r}_i^{[p]}\}_{i=1}^n, \mu_{1,n-1}^{[p]}, \Sigma_{1,n-1}^{[p]}, \ldots, \mu_{K,n-1}^{[p]}, \Sigma_{K,n-1}^{[p]}, \boldsymbol{c}_i, w_{n-1}^{[p]}\right)$$

12: $\quad\quad\quad\quad$ add $Y_n^{\boldsymbol{c}_i,[p]}$ to $Y_n^{\boldsymbol{c}_i}$
13: $\quad\quad\quad$ **end for**
14: $\quad\quad$ **end if**
15: $\quad$ **end for**
16: **end for**
17: append $\boldsymbol{C}_n$ to Υ
18: remove the last sequence from $\boldsymbol{C}_n$ and assign it to $\boldsymbol{c}$
19: $n := \text{length}(\boldsymbol{c})$
20: $Y_{n-1} := Y_n^{\boldsymbol{c}}$ (*)
21: $Y_n = \emptyset$
22: **for** $p = 1$ to P **do**
23: $\quad$ do lines 5 to 29 from Algorithm 4.1
24: **end for**
25: do lines 31 to 35 from Algorithm 4.1 and assign the returned particle set to $Y_n^{\boldsymbol{c}}$
26: **return** $\boldsymbol{c}$

(*) The particle filter which is being processed is now part of the global particle set, so we assign it to the local variable Y_{n-1} for the consistent notation with previous algorithms.

4.6 Simulations

To demonstrate that the proposed algorithm works well in simulations, we illustrate the results for various setups, ranging from the noiseless case and known labeling to the noisy measurements and unknown labeling.

We construct a room to be a right-angled trapezoid as in Fig. 4.10. The lengths of the walls that are bases of the trapezoid are 8 m (lower) and 6 m (upper), while the lateral sides are 6 m (left) and 6.32 m (right). The trajectory is piecewise linear, described as a sequence of motion vectors, $\{\boldsymbol{v}_n(\ell_n, \phi_n)\}_{n=2}^N$. The lengths of the steps range from 0.7 to 2.7 m.

Algorithm 4.4 The complete EchoSLAM algorithm with unknown labeling

$\boldsymbol{c}_1 = [1]$
$\Upsilon = [\boldsymbol{c}_1]$
for $p = 1$ to P **do**
  $Y_1^{\boldsymbol{c}_1,[p]} = \left(\boldsymbol{r}_1^{[p]} := [0,0]^\top, \boldsymbol{c}_1^{[p]} := \boldsymbol{c}_1, w_1^{[p]} := 1/P\right)$
end for
$Y_1^{\boldsymbol{c}_1} = \{Y_1^{\boldsymbol{c}_1,[1]}, Y_1^{\boldsymbol{c}_1,[2]}, \ldots, Y_1^{\boldsymbol{c}_1,[P]}\}$
for $n = 2$ to N **do**
  send a control $\boldsymbol{v}_n$ to the robot
  record and process a RIR, and compute δ_n from the propagation time of the first peak
  repeat
    invoke Algorithm 4.3 with δ_n, $\boldsymbol{v}_n$ and $\boldsymbol{c}_{n-1}$
  until the length of the returned sequence of labels is n
  assign the returned sequence of labels to $\boldsymbol{c}_n$
end for
find the particle $\hat{p}$ in $Y_N^{\boldsymbol{c}_N}$ with the largest weight
apply a rigid transformation to $\{\boldsymbol{r}_n^{[\hat{p}]}\}_{n=1}^N$ and $\{\mu_{k,N}^{[\hat{p}]}\}_{k=1}^K$, such that it transforms the waypoints to their original coordinates
display the estimated trajectory and the walls

Noiseless case. Fig. 4.10 illustrates the noiseless case, $\sigma_\delta = 0$, $\sigma_\ell = 0$ and $\sigma_\phi = 0$, with the known labeling. All four walls of the room are correctly estimated after 13 steps. Many repeated experiments have shown that in this simple case, the walls and the trajectory are always correctly estimated after only a few steps.

Unknown labeling and noisy distance measurements. Now we add one level of difficulty to the problem by assuming that the labeling is not known and $\sigma_\delta = 5$ cm, while keeping the motion vectors noiseless, $\sigma_\ell = 0$ and $\sigma_\phi = 0$. Our simulations show that the estimated rooms are very close to the original room: we ran 50 experiments with different trajectories in the same trapezoid room, and computed the root mean squared errors (RMSE) of the estimated corners' locations. The average error of all experiments is 7.2 cm per corner. We observe that the labels of the estimated room do not always perfectly match the ground truth labels; however, the wrong labels do not noticeably affect the accuracy of the reconstruction. The reason is that the measurements are wrongly assigned only when the robot is almost equally apart from the two closest walls. Therefore, when the trajectory is known, the reconstruction error is only due to the noisy distance measurements and it is not corrupted by mistakes in the labeling algorithm.

Unknown labeling and noisy motion and distance measurements. In the most general case, all measurements are considered to be noisy and the labeling is unknown. Fig. 4.11 shows estimated rooms and trajectories after every step of the robot in the experiment with the following parameters: $\sigma_\delta = 5$ cm, $\sigma_\ell = 1$ cm, $\sigma_\phi = 0.05$ rad, $P = 30$, $N = 20$, $K_{max} = 4$. The values assumed in our noisy setup are conservative. For a comparison, the corresponding values obtained with our real mobile robot are $\sigma_\ell = 0.27$ cm and $\sigma_\phi = 0.024$ rad (we report these values in Section 4.7.1). As the labeling is unknown, Algorithm 4.4 branches and explores different label sequences. In Fig. 4.11 we plot all particles from the particle set that terminates our algorithm,

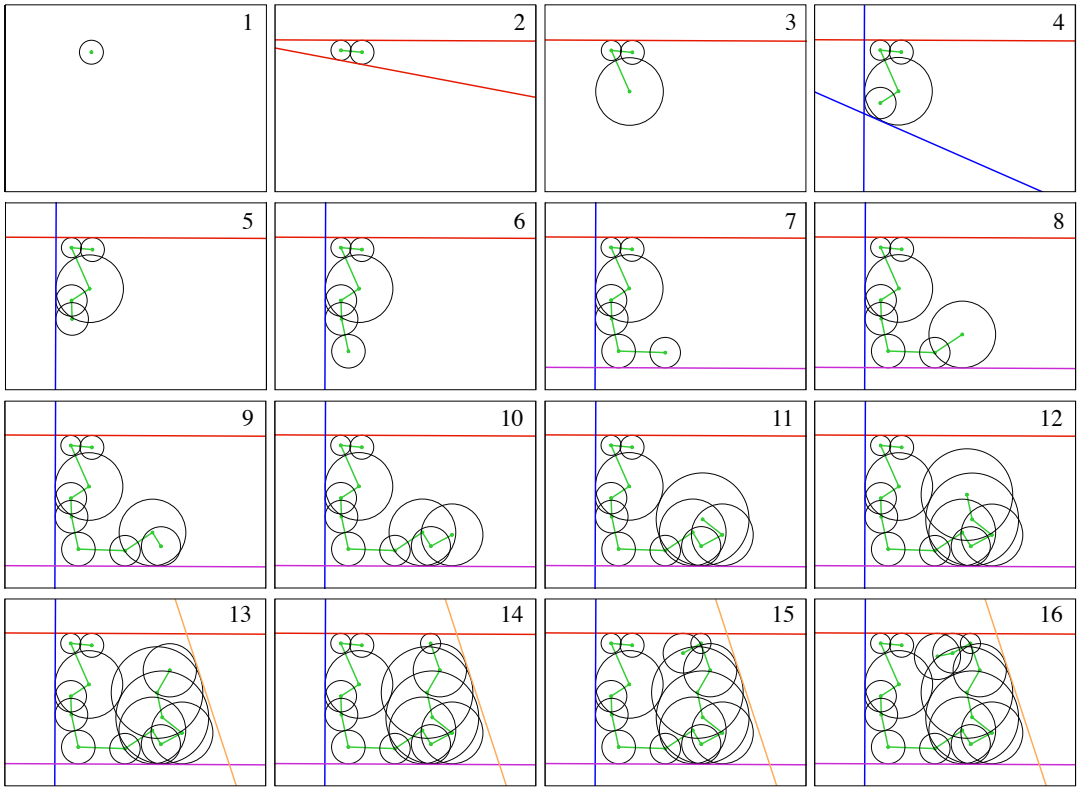


Figure 4.10: An example of one execution of the EchoSLAM algorithm with known labeling and noiseless measurements. The room and the trajectory are correctly estimated after a few steps.

i.e., the particle set for which we reach the maximum depth of the labeling tree. This means the labeling from that particle set is feasible; indeed, in the presented case the labeling is correct at all waypoints.

4.7 Experimental results

In this section we demonstrate our end-to-end algorithm in practice, which creates a 2D floor plan with the robot's trajectory from the recorded room impulse responses.

4.7.1 Hardware

We designed a modular mobile robot with the main characteristics of being accurate, easily controllable and allowing a quick retrieval of the experimental data.

Robot design. The robot's architecture is composed of three layers. The electronic and the mehanical layer perform electrical and mechanical functions that are shared among different applications (power supply, control and localization of the robot), while the experiment setup

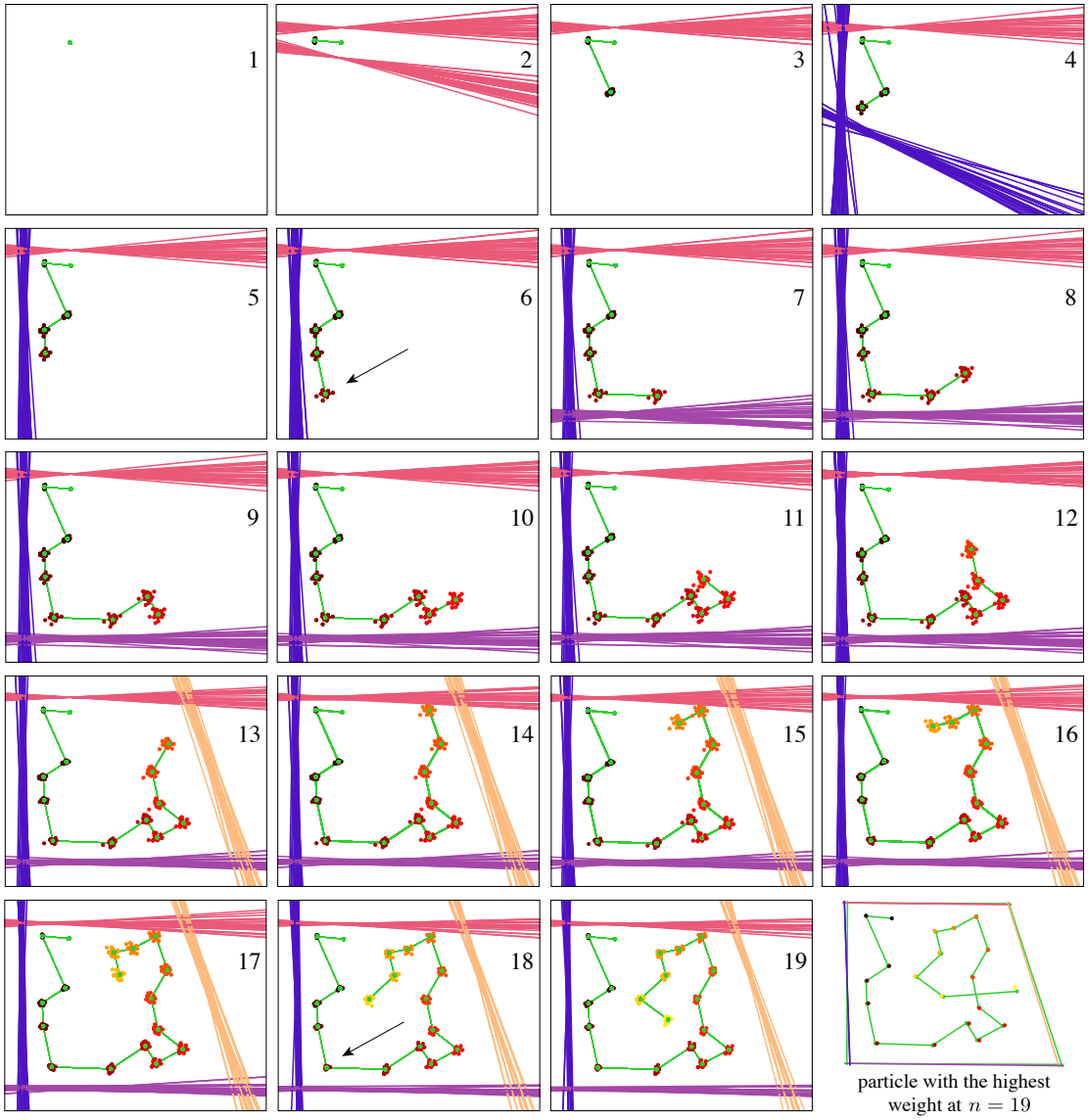
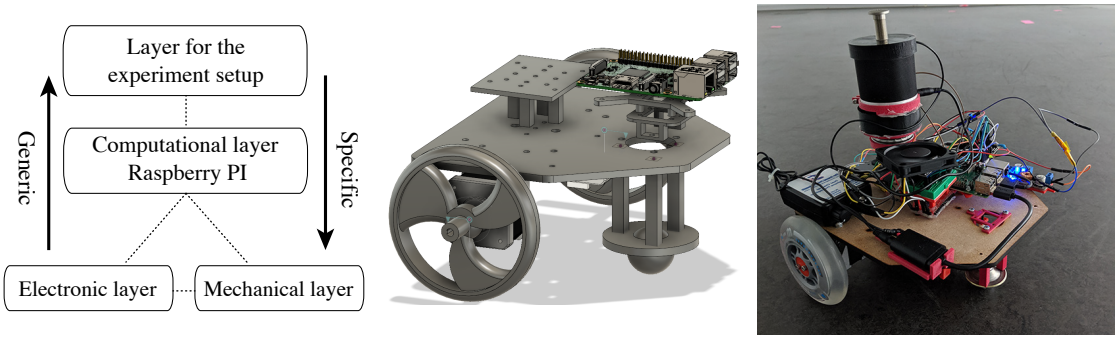


Figure 4.11: An example of one execution of the EchoSLAM algorithm with unknown labeling and noisy measurements, with $\sigma_\delta = 5$ cm, $\sigma_\ell = 1$ cm, $\sigma_\phi = 0.05$ rad. The numbers in the right upper corner enumerate the steps. The benefit of resampling is evident if we compare for example step $n = 6$ and $n = 18$; a black arrow is pointing towards the waypoints of interest. At $n = 6$ we generated new particles according to our motion model. In the following steps we incorporated new measurements and improved our estimates of the walls, which in turn helped us to improve the noisy estimates of the trajectory. At $n = 18$, the distribution of the particles generated at $n = 6$ is much narrower and centered around the ground truth waypoints. The last figure shows the particle with the highest weight, along with the overlaying ground truth room-trajectory configuration in green.



a) b) c)

Figure 4.12: a) Schematic diagram of the robot three-layer architecture. b) Overall model of the mobile robot. c) Constructed robot with mounted loudspeaker and microphone.

Electronic layer			
Sensors	**Control**	**Power supply**	
IMU	Microcontroller	Step-down voltage regulator	Battery pack
Temperature sensor	Stepper motor drivers	Battery level indicator	Relay
Long-range laser sensors		Battery menagement system	Charging port

Mehanical layer	
Differential drive	**Structural**
Scooter wheels	Base "parametric botled" plate
Caster ball	Vibration dampeners
Stepper motors	Stepper motors mounts
Scooter wheels coupling	Electronic layer mounts
	Experimentation setup mounts

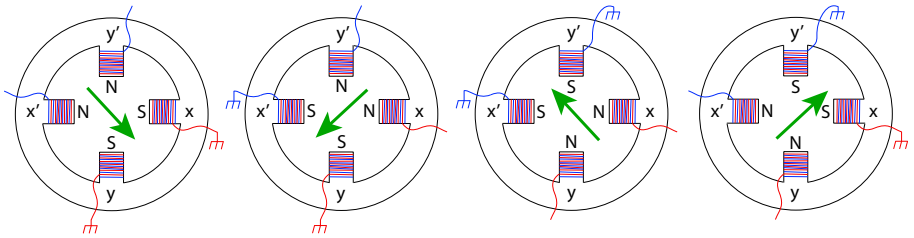
Experiment setup layer
EchoSLAM
USB sound card
Mechanical rotation platform
Servo motor
Loudspeaker
Microphone

Figure 4.13: Main components of every layer.

layer is specific to the application (microphone and loudspeaker for the EchoSLAM algorithm). The idea is to achieve a scalable, inter-operable system, where each layer can be replaced, modified and upgraded independently while maintaining the normal functionality of the robot. A schematic diagram and model of the robot architecture are shown in Fig. 4.12. Below we describe the main components and their functions, while the detailed list of layers elements can be found in Fig. 4.13.

The electrical layer is comprised of three modules:

- Power supply. The solution is based on a rechargeable battery pack with a battery management system and a level indicator. The components from the outer layer are either low consumption components (5V), which are powered through the step down voltage regulator, or high consumption components (12V), which are directly powered from the battery pack and turned on and off by the electrical relay. Such a configuration enables stable and



	x	x′	y	y′
a	0	1	0	1
b	1	0	0	1
c	1	0	1	0
d	0	1	1	0

e)

Figure 4.14: Operation principle of a stepper motor. A permanent magnet is used for a rotor, represented as a green arrow. Coils are put on four stators denoted by x, x', y and y', which control the rotation of the motor: when the electric current is poured into x' and y', the magnet is in the position illustrated in a). By applying an electric current to the coils x and y', the magnet rotates by 90° clockwise and its state is shown in b). By energizing the coils in order outlined in e), the rotor turns clockwise.

efficient power supply.

- Sensors. Long-range sensors are used for distance and proximity sensing in the obstacle avoidance module. The inertial measurement unit (IMU) is constituted of a gyroscope, an accelerometer and a magnetometer. The temperature sensor measures the ambient temperature, which alters the speed of sound.
- Control. The robot is equipped with motor drivers that run the stepper motors. The drivers are controlled by a microcontroller.

The mechanical layer is made of two modules:

- A differential drive. Our differential wheeled robot is driven by two scooter wheels coupled to stepper motors. A caster ball is used as free rolling ball to provide stability to the system and offload the motors.
- A structural module. The mechanical modularity of the robot is represented through the base. It is a plate with a fixed number of holes distributed in a parametric way. The holes allow to assemble various mounts (e.g. Raspberry PI mount, stepper motor mounts) and connect different components of the robot. Vibration dampeners are also used to reduce the vibrations of the stepper motors.

In the current version of the robot, the experiment setup layer can collect data and perform tasks for two different applications: the EchoSLAM algorithm and Wi-Fi based indoor localization. Our EchoSLAM algorithm requires a USB sound card, loudspeaker, microphone, mechanical rotating platform and a servo motor, while the Wi-Fi based localization needs an RTT access point and phone mounts.

Assessment of the robot's positioning performance. The choice of stepper motors benefits both resolution and accuracy. The working principle behind stepper motors is illustrated in Fig. 4.14. The rotor, represented as a green arrow in Fig. 4.14, aligns itself with the magnetic field generated by energizing a coil or a couple of coils in the desired direction. Thus, by energizing the coil in a continuous sequence, we can rotate the rotor in discrete steps. To accurately hold

a certain position, we can stop the sequence at any given time and keep the corresponding coil energized. Our stepper motor has 200 steps per revolution, which corresponds to 1.8° resolution. To increase the resolution and achieve smoother movement, we use microstepping. It is based on the idea that energizing several coils with different voltages at the same time creates additional intermediate stable magnetic field directions.

Three types of motions are tested to assess the accuracy and precision of the robot: translation, rotation and circular motion. We compare the executed trajectory with the given commands. As we use a microstepping factor of 16 (i.e., increase the number of steps per revolution by a factor of 16), a stepper with 200 steps per revolution and a scooter wheel of nominal diameter 100 mm, the achieved resolution is $\Delta d = \pi * 100 * \frac{1.8}{360} * \frac{1}{16} \approx 0.098$ mm.

To evaluate the translation accuracy, we asked the robot to move forward by 50 cm and backwards by 50 cm, and we repeated the experiment 20 times. With the implemented stepper motor technology, the accuracy and precision should not change for different distances travelled by the robot, in particular for the range of distances used in our EchoSLAM algorithm (50 to 200 cm). When the robot was asked to move forward by 50 cm, on average it moved 50.295 cm with a standard deviation of 0.248 cm. Similarly, when the robot was asked to move backward by 50 cm, on average it moved 50.305 cm with a standard deviation of 0.294 cm. We can conclude that the system is precise and accurate for linear translations and that the noise level used in our numerical simulations is high and conservative compared to the real experiments.

To evaluate the accuracy of the rotation and circular motion for a given radius, we repeated a number of similar tests. We varied the value of the rotational angle as well as of the radius. All runs showed that there is minimal shift of the rotation center, and resulted in an average standard deviation of 0.024 rad.

4.7.2 Algorithm in practice

The robot described in the previous section is used to reconstruct a classroom on our campus by performing Algorithm 4.4. We undertook the experiments in the same rectangular room as in Chapter 3 and recorded acoustic measurements with the same cylindrical wireless speaker, Veho 360. Ground truth waypoints are obtained by measuring the distances between the measurement locations and the reference points with a laser meter as in Section 3.6.2.

The simplicity and usability of the robot is illustrated in Fig. 4.15. Many meters of cables, several microphones, stands, adapters and sound mixer are replaced with a single robot 30 cm long and wide. The robot is easily controllable: to move it to the next location, we need to provide a pair of values (ℓ_n, ϕ_n), where ℓ_n is the length of the step, and ϕ_n is the relative angle with respect to the current orientation. Those values are then translated to commands called G-Code that can be interpreted and executed by the robot. Instead of sending a command after every movement of the robot, the experiment can be fully automated if we provide a complete trajectory $\{(\ell_n, \phi_n)\}_{n=2}^{N}$ at the beginning of the experiment. Then, for every n, the robot moves by $\boldsymbol{v}_n = [\ell_n \cos\phi_n, \ell_n \sin\phi_n]^\top$, stops, plays a sweep, records the response, increases n by 1 and repeats such behavior. We can retrieve every recording instantaneously through a web-based interface and process it in real-time. During our experiments, however, we paused a robot after every measurement to mark its ground truth location on the floor.

Room impulse responses recorded at $N = 20$ locations inside of the room are shown in Fig. 4.16. To extract the most prominent peaks, we apply a variant of the MUSIC algorithm [114], which we already used in Chapter 3. Contrary to algorithms in Chapter 3, Algorithm 4.4 requires only one first-order echo in every RIR, so their detection is flawless; observe in Fig. 4.16 that

a) b)

Figure 4.15: Comparison of the measurement setup before and after the development of the robot. a) Standard installation. b) Look, no cables!

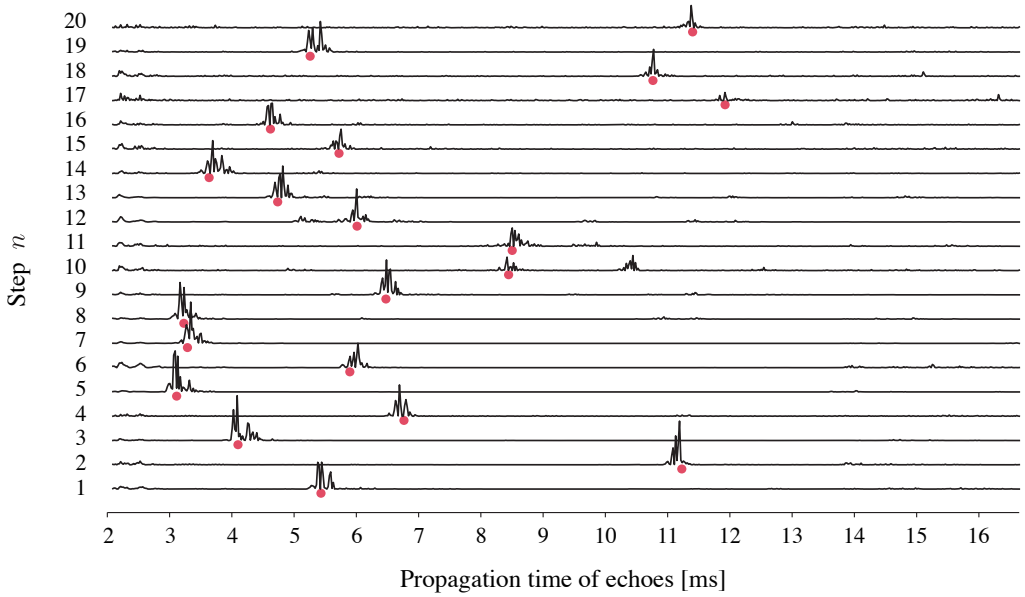
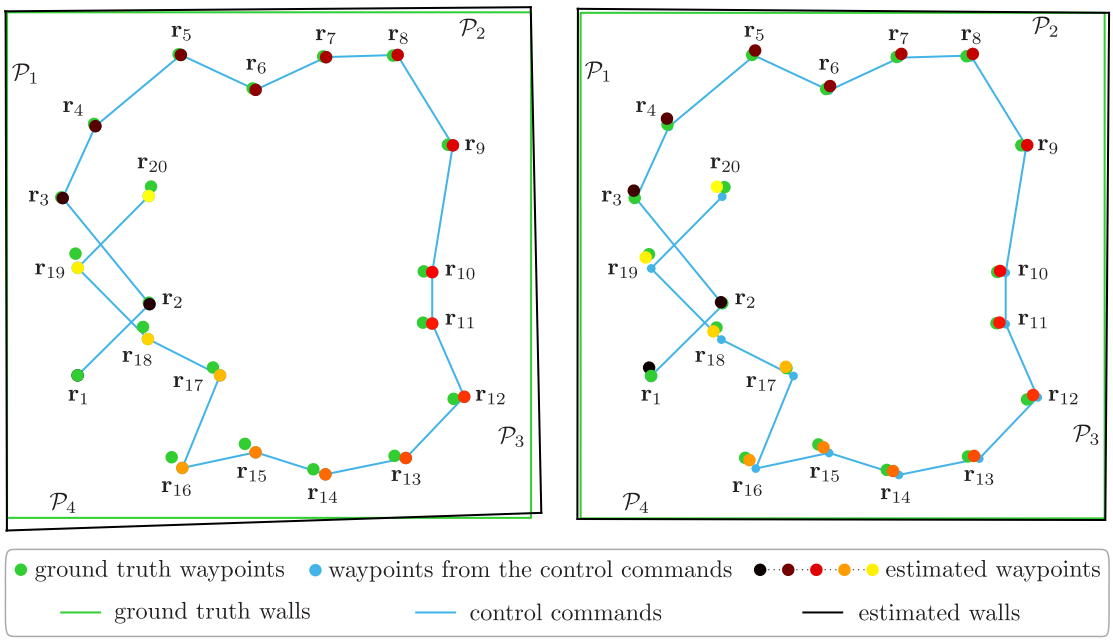


Figure 4.16: The RIRs recorded by the robot at $N = 20$ waypoints. Red dots indicate locations where we expect to see the first echo from the nearest wall computed from the ground truth. We can observe that the first peak in every recorded RIR is (almost) perfectly aligned with its expected location and there are no false positive or false negative echoes.

the first peak in every RIR is indeed the first echo from the nearest wall, and its amplitude is significantly higher than from any other peak in the RIR.

We provide the propagation times of the detected first peaks to Algorithm 4.4, along with the control commands that were executed by the robot to move in the space. Given only those measurements without the labels, Algorithm 4.4 precisely reconstructs the room and the trajectory, as illustrated in Fig. 4.17b. The average error for the corners and waypoints is 8.3 cm and 6.5



a) b)

Figure 4.17: Room reconstruction and robot localization with unknown labeling, given the measurements provided by the real robot. a) One execution of the modified EchoSLAM algorithm, in which the trajectory was not updated jointly with the walls. The trajectory was estimated directly from the controls given to the robot, and based on such a trajectory, we computed the distributions of the walls. We can notice that the estimation error accumulates over time, both for the walls and the waypoints. The average error for the corners in the room is 11.9 cm, while for the waypoints it is 6.8 cm. b) One execution of the presented EchoSLAM algorithm. In comparison with a), it clearly visualizes an advantage of jointly updating the trajectory estimates along with the walls. The average error for the corners and the waypoints is 8.3 and 6.5 cm, respectively.

cm, respectively. To demonstrate the benefit of jointly updating the room trajectory with the walls estimates, we now modify Algorithm 4.4 and pretend to believe that the robot's motion is noiseless, so that the waypoints are simply obtained from the control commands $\boldsymbol{r}_n = \sum_{i=2}^{n} \boldsymbol{v}_i$. In such a scenario, we get the error of 11.9 cm for the vertices and 6.8 cm for the waypoints; this is illustrated in Fig. 4.17a. When comparing the reconstructed rooms and waypoints in Fig. 4.17a and b, we can observe that in Fig. 4.17a the error accumulates with new steps and every wall that is observed later in the process has a larger offset from its ground truth value. On the other hand, in Fig. 4.17b we correct our belief of the trajectory with every new measurement, and recover the waypoints and the walls that are very close to the ground truth.

4.8 Conclusion

In this chapter we developed a robust real-time system to solve the SLAM problem with a single omnidirectional sonar mounted on a robot. We proposed EchoSLAM, a particle-based solution that reconstructs side walls of 3D rooms from acoustic measurements obtained at several locations inside unknown rooms.

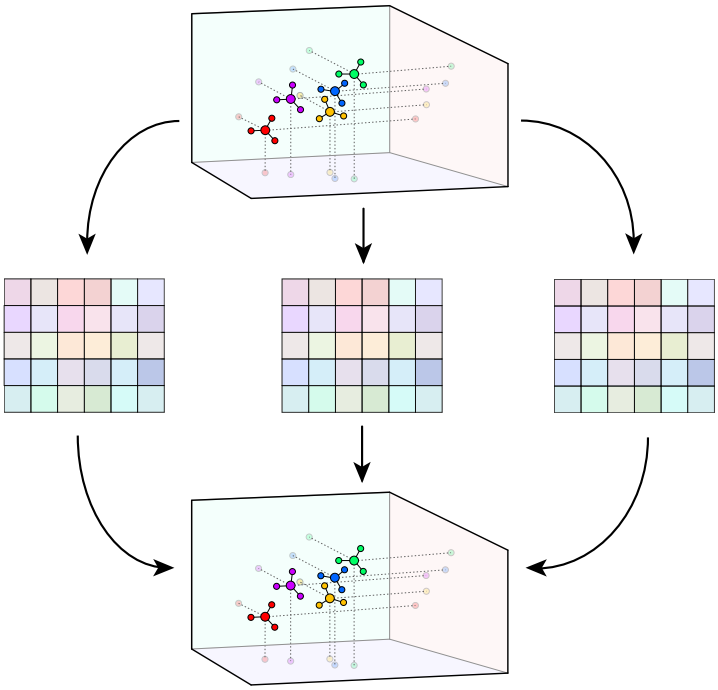
Our algorithm showed many advantages compared to the existing approaches: it can leverage only range-measurements between the robot and walls, without any angular information, and it does not require a full set of first- or higher-order echoes; to the contrary, it needs only the first echo from the nearest wall. Our experiments confirmed that the corresponding peak is always very prominent in room impulse responses, even when they are recorded with cheap off-the-shelf 360 speakers. To the best of our knowledge, there are no other devices or robots capable of localizing themselves in unknown environments from the RIRs recorded by only one omnidirectional source and receiver.

We also devised a labeling algorithm to match the detected echoes with the walls from which they reflect. The proposed echo labeling method is flexible: by changing the values of two parameters, at every measurement we can vary between running an exhaustive search over all labels and selecting only the most likely label. The former always finds the correct labeling, but is considerably slower that the latter, which has a higher risk of picking a wrong label, but runs almost in real-time. The proposed branch and prune method therefore allows users to specify the desired speed–robustness trade-off by simply configuring the two parameters.

To evaluate and use EchoSLAM in real environments, we built a three-wheel modular robot. It can precisely follow a given piecewise linear trajectory, and after every segment play a sine sweep and record the room impulse. The results of our experiment conducted in a classroom on our campus demonstrate that the room geometry and robot's measurement locations are precisely recovered from the echoes of the nearest walls with unknown labels. Future work involves more systematic evaluation of the system by performing more extensive numerical and real world experiments.

Chapter 5

A Binaural Bat Without a Sense of Direction Can Hear the Shape of Rooms



If you refuse to take account of theory, then you have forgotten that practice is often an offspring of theory.

Simply Transcribed: Quotations from Writings
Fausto Cercignani

5.1 Introduction

In all the previous chapters we considered a collocated omnidirectional source and receiver mounted together on a mobile robot that follows a trajectory inside an unknown room. In Chapter 3 and Chapter 4 we proposed algorithms to recover rooms and trajectories from the first-order echoes recorded by the robot by leveraging the kinematics measurements. However, if we have no information about the kinematics, i.e. no constraints on the trajectory, our analysis in Chapter 2 shows that propagation times of the first-order echoes cannot uniquely determine the room geometry and localize the robot.

In 2D, this happens only in some special rooms or with particular trajectories. For example, we cannot distinguish parallelograms obtained by shearing and scaling a rectangular room. Moreover, given the first-order echoes measured along a linear trajectory, we can find infinitely many rooms of various shapes with the same first-order echoes. In 3D, this is a common problem for much wider set of rooms and trajectories, as well as for any room with less than six walls.

5.1.1 Main contributions

In this chapter we show that placing additional receivers on a robot enables a unique reconstruction of almost all rooms and trajectories. We also prove that for certain rooms, a mobile setup without any fixed infrastructure inside a room is not sufficient for the unique localization: we identify rooms in which we can never uniquely localize the robot regardless of the number of its sources and receivers. As such "rooms" are either of an infinite length or height, they are not of practical interest. For the others, we determine the smallest number of receivers that uniquely specify rooms and trajectories with the first-order echoes. We show that we need at least two receivers in 2D and three in 3D, and we identify the problems that arise with fewer receivers.

5.1.2 Outline

The problem setup is introduced in Section 5.2. As the analysis and the results differ for 2D and 3D, we derive them separately. For the 2D setup in Section 5.3, we first assume to have one source and one receiver, but unlike our analysis in Chapter 2, the source and the receiver are not collocated. In Section 5.3.1 we examine all room-trajectory configurations that suffer from non-uniqueness in reconstruction with the collocated setup, and we show that a non-collocated setup resolves some of these ambiguities, while some other appear. In Section 5.3.2 we prove that by adding one receiver, we can uniquely specify room-trajectory configurations. Analogously, we divide Section 5.4 concerning the 3D setup in two parts: in Section 5.4.1 we identify cases when one source and two receivers cannot uniquely determine room-trajectory configurations, while in Section 5.4.2 we add one receiver and prove that this resolves all ambiguities in the reconstruction.

5.2 Problem setup

We assume a robot that autonomously moves inside an unknown room. An omnidirectional acoustic source and omnidirectional receiver(s) are mounted on the robot. We denote the number of receivers by $M \geq 1$ and we aim to find the smallest M that guarantees unique localization of the robot and reconstruction of the room. The only available information are the first-order echoes of the pulses emitted and recorded by the robot.

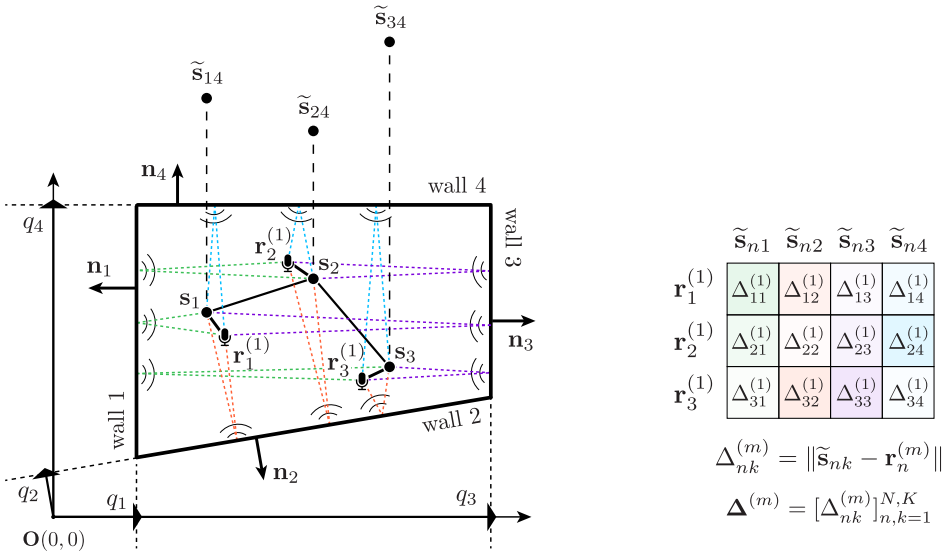


Figure 5.1: A room with $K = 4$ walls $\{\mathcal{P}_k\}_{k=1}^K$. The trajectory consists of $N = 3$ waypoints $\{\mathbf{s}_n\}_{n=1}^N$. The waypoints are the locations of the source, while the receiver's locations are marked with $\{\mathbf{r}_n^{(1)}\}_{n=1}^N$. The image sources are shown for $k = 4$. We illustrate the sound propagation path from $\mathbf{s}_n$ to $\mathbf{r}_n^{(1)}$ for every n and k, where the echoes from the same wall are indicated by the same color. On the right, we depict the corresponding measurement matrix $\boldsymbol{\Delta}^{(1)}$.

The robot's trajectory is defined by N waypoints $\{\boldsymbol{s}_n\}_{n=1}^N$, where $\boldsymbol{s}_n \in \mathbb{R}^D$ is the location of the source at the nth waypoint and D is the dimension of the space. The orientation of the robot can change at each waypoint $n = 1, \ldots, N$, and it is given by $\boldsymbol{u}_n = [\cos\alpha_n, \sin\alpha_n]^\top$ with respect to some reference direction. There are M receivers with a known relative geometry mounted on the robot. Each receiver $m = 1, \ldots, M$ is determined by its distance $\ell^{(m)}$ and the orientation $\beta^{(m)}$ from the source, with $\beta^{(1)} = 0$. Then, the position of the mth receiver at the nth waypoint $\boldsymbol{r}_n^{(m)}$ is computed from

$$\boldsymbol{r}_n^{(m)} = \boldsymbol{s}_n + \ell^{(m)}\boldsymbol{u}_n^{(m)}, \tag{5.1}$$

where $\boldsymbol{u}_n^{(m)} = [\cos(\alpha_n + \beta_m), \sin(\alpha_n + \beta_m)]^\top$.

We describe a room with K planar walls $\{\mathcal{P}_k\}_{k=1}^K$ and reuse the same notation as in the previous chapters, i.e., $\mathcal{P}_k = (\boldsymbol{n}_k, q_k)$, where $\boldsymbol{n}_k \in \mathbb{R}^D$ is the unit normal and q_k is the distance of the wall from the origin. At every waypoint, the source produces a pulse and M receivers register the direct path of the sound and its reflections from the walls. In the image source model [4, 24] we replace the reflections from the walls with signals produced by image sources; they are mirror images of the real source across the corresponding walls. For a first-order echo and the kth wall, the image source $\widetilde{\boldsymbol{s}}_{nk}$ of the real source $\boldsymbol{s}_n$ is given as $\widetilde{\boldsymbol{s}}_{nk} = \boldsymbol{s}_n + 2\big((\boldsymbol{p}_k - \boldsymbol{s}_n)^\top \boldsymbol{n}_k\big)\boldsymbol{n}_k$, where $\boldsymbol{p}_k \in \mathbb{R}^D$ is any point on the kth wall. We illustrate the notation for $m = 1$, $N = 3$ and $K = 4$ in Fig. 5.1.

The propagation time $\tau_{nk}^{(m)}$ of the sound wave from the source $\boldsymbol{s}_n$ to the receiver $\boldsymbol{r}_n^{(m)}$ reflected once from the kth wall is proportional to the distance $\Delta_{nk}^{(m)}$ between the image source $\widetilde{\boldsymbol{s}}_{nk}$ and

$\boldsymbol{r}_n^{(m)}$,

$$\tau_{nk}^{(m)} = \frac{\Delta_{nk}^{(m)}}{c_s} = \frac{\|\widetilde{\boldsymbol{s}}_{nk} - \boldsymbol{r}_n^{(m)}\|}{c_s},$$

where c_s is the speed of sound.

We can thus find the distances $\Delta_{nk}^{(m)}$ by measuring the times of arrival of the first-order echoes. We assume that we can match echoes with their reflecting walls, so that their distances have been given to us in a form of a matrix $\boldsymbol{\Delta}^{(m)} \in \mathbb{R}^{N\times K}$ with

$$\boldsymbol{\Delta}^{(m)} = [\Delta_{nk}^{(m)}]_{n=1,k=1}^{N,K}.$$

In this chapter we extend the notion of *room-trajectory* configurations $\mathcal{R}$ introduced in Chapter 2. As we lose the symmetry of the collocated setup considered in the previous chapters, we have to include the information about the robot's orientation $\boldsymbol{u}_n$; thus, a room-trajectory configuration is defined as $\mathcal{R} = \big(\{\mathcal{P}\}_{k=1}^K, \{(\boldsymbol{s}_n, \boldsymbol{u}_n)\}_{n=1}^N\big)$, and the corresponding distance measurements are given by $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^M$.

In the following, we address the question whether a collection of $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^M$ uniquely determines a room-trajectory configuration $\mathcal{R}$ that generates it and find the smallest M for which it does. As it is clear that rotated, reflected and translated versions of $\mathcal{R}$ all give the same $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^M$, we consider them to be the same configuration. We formalize the uniqueness problem below:

PROBLEM 5.1 What is the smallest M for which the probability of a unique recovery of room-trajectory configuration $\mathcal{R}\big(\{\mathcal{P}\}_{k=1}^K, \{(\boldsymbol{s}_n, \boldsymbol{u}_n)\}_{n=1}^N\big)$ from a collection of $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^M$ is equal to one for all $\{\mathcal{P}\}_{k=1}^K$ and all $\{\boldsymbol{s}_n\}_{n=1}^N$?

In our analysis the uniqueness is considered under the following assumption:

ASSUMPTION 1 The orientations $\boldsymbol{u}_n$ of the robot are picked independently at random for every $n = 1, \ldots, N$ following a non-degenerate probability density function.

We remark that certain ambiguities in the reconstruction remain even with an arbitrarily large number of sources and receivers; this means that some information is irremediably lost and cannot be recovered from the first-order echoes. This happens only in some specific rooms, which are not of practical interest (e.g. infinitely long corridors and infinitely tall prisms). As results differ for 2D and 3D, we analyze them separately.

5.3 Two-dimensional setup

We begin with the 2D case, where the walls are lines and their unit normals are parametrized as $\boldsymbol{n}_k = [\cos\varphi_k, \sin\varphi_k]^\top$, with $\varphi_k \in [0, 2\pi)$ for every k.

In Chapter 2, we proved that a single collocated source and receiver do not uniquely determine rooms and trajectories. The minimal modification of such a setup is to keep $M = 1$ and move the receiver away from the source. We consider this case in Section 5.3.1 and show that while some ambiguities are resolved, new ones arise. We therefore increase the number of receivers to $M = 2$ in Section 5.3.2 and prove that such setups uniquely specify room-trajectory configurations.

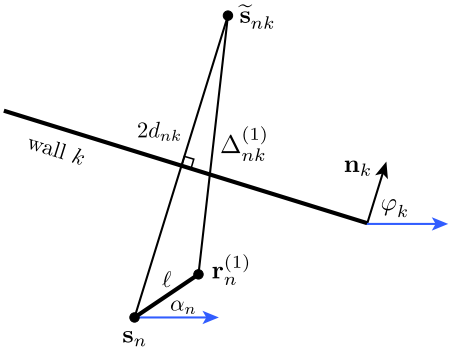


Figure 5.2: One source $\boldsymbol{s}_n$ and one non-collocated receiver $\boldsymbol{r}_n^{(1)}$ at the nth waypoint. Vector $\boldsymbol{n}_k$ is the outward-pointing wall normal and $\widetilde{\boldsymbol{s}}_{nk}$ is the image source corresponding to the first-order echo. The blue lines represent the reference direction.

5.3.1 2D: One receiver

Consider a source and receiver mounted on a robot at distance ℓ; an example with one wall is illustrated in Fig. 5.2. The measurement $\Delta_{nk}^{(1)}$ between the image source $\widetilde{\boldsymbol{s}}_{nk}$ and the receiver $\boldsymbol{r}_n^{(1)} = \boldsymbol{s}_n + \ell \boldsymbol{u}_n$ is given as

$$(\Delta_{nk}^{(1)})^2 = \|\widetilde{\boldsymbol{s}}_{nk} - \boldsymbol{r}_n^{(1)}\|^2 = 4d_{nk}^2 - 4d_{nk}\ell \boldsymbol{n}_k^\top \boldsymbol{u}_n + \ell^2, \tag{5.2}$$

where $d_{nk} = q_k - \boldsymbol{n}_k^\top \boldsymbol{s}_n$ is the distance of $\boldsymbol{s}_n$ from the kth wall.

While the separation of the source and receiver resolves one of the main ambiguities of the collocated setup—non-unique reconstruction of parallelogram rooms—it does not come without cost. The width of infinitely long corridors and trajectories in rectangular rooms are not uniquely determined by (5.2). We analyze these cases from Section 5.3.1.1 to Section 5.3.1.4.

5.3.1.1 Rectangular rooms

Rectangular rooms are uniquely specified by $\boldsymbol{\Delta}^{(1)}$, but the waypoints are not; given a rectangular room, there are four source-receiver pairs at distance ℓ with the same $\boldsymbol{\Delta}^{(1)}$.

To show this, denote $\boldsymbol{s}_n = [s_{n,x}, s_{n,y}]^\top$ and substitute the wall normals of a rectangular room,

$$\boldsymbol{n}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \boldsymbol{n}_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \boldsymbol{n}_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \boldsymbol{n}_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{5.3}$$

into (5.2). We obtain

$$\begin{aligned}
(\Delta_{n1}^{(1)})^2 &= 4(q_1 - s_{n,y})^2 - 4(q_1 - s_{n,y})\ell \sin\alpha_n + \ell^2, \\
(\Delta_{n2}^{(1)})^2 &= 4(q_2 + s_{n,x})^2 + 4(q_2 + s_{n,x})\ell \cos\alpha_n + \ell^2, \\
(\Delta_{n3}^{(1)})^2 &= 4(q_3 + s_{n,y})^2 + 4(q_3 + s_{n,y})\ell \sin\alpha_n + \ell^2, \\
(\Delta_{n4}^{(1)})^2 &= 4(q_4 - s_{n,x})^2 - 4(q_4 - s_{n,x})\ell \cos\alpha_n + \ell^2,
\end{aligned} \tag{5.4}$$

for every $n = 1, \ldots, N$.

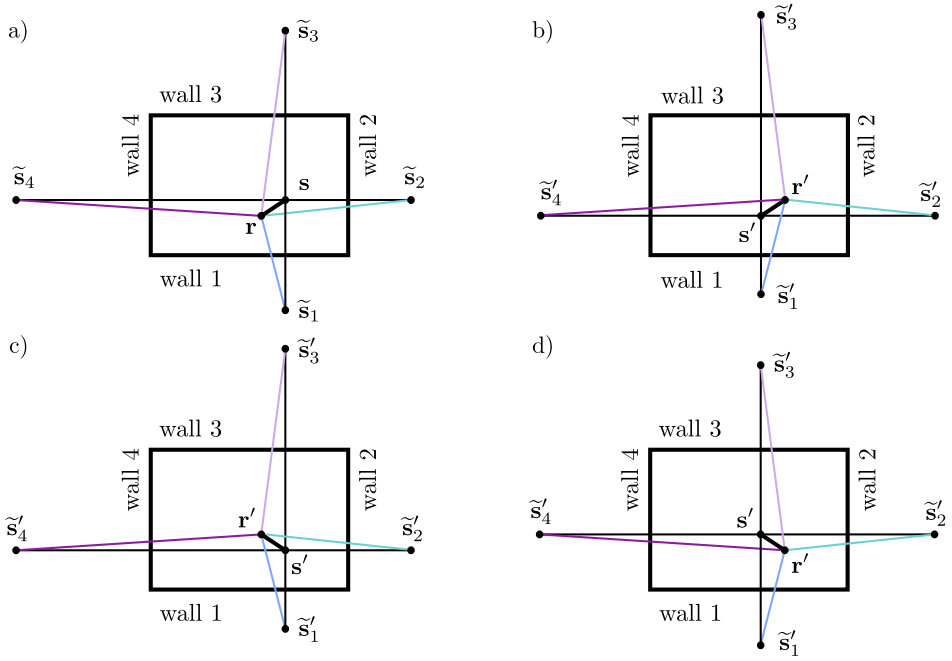


Figure 5.3: Distinct waypoints in the same rectangular room can measure the same distances $\boldsymbol{\Delta}^{(1)}_{\mathcal{R}} = \boldsymbol{\Delta}^{(1)}_{\mathcal{R}'}$, where $\mathcal{R} = \big(\{\mathcal{P}\}_{k=1}^{K}, \{(\boldsymbol{s}_n, \boldsymbol{u}_n)\}_{n=1}^{N}\big)$ is illustrated in a), while $\mathcal{R}' = \big(\{\mathcal{P}\}_{k=1}^{K}, \{(\boldsymbol{s}'_n, \boldsymbol{u}'_n)\}_{n=1}^{N}\big)$ can be any room-trajectory configuration from b) to d). Identical measurements in two setups are marked with the same color.

The system (5.4) does not change if we replace the robot's orientation and the coordinates $(\alpha_n, s_{n,x}, s_{n,y})$ with one of the following triples: $(2\pi - \alpha_n, s_{n,x}, s_{n,y} + \ell \sin\alpha_n)$, $(\pi - \alpha_n, s_{n,x} + \ell\cos\alpha_n, s_{n,y})$ or $(\pi + \alpha_n, s_{n,x} + \ell\cos\alpha_n, s_{n,y} + \ell\sin\alpha_n)$. This implies that the waypoints cannot be reconstructed uniquely from $\boldsymbol{\Delta}^{(1)}$, but the set of possible room-trajectory configurations is finite, unlike with a collocated source and receiver. An example is illustrated in Fig. 5.3.

5.3.1.2 Infinitely long corridors

If a collocated source and receiver are placed in a corridor of infinite length, the distance between a waypoint and two walls uniquely determines the width of the corridor. This is not the case for a non-collocated setup; an example of two corridors with the same $\boldsymbol{\Delta}^{(1)}$ but different widths and waypoints is given in Fig. 5.4.

We describe corridors of infinite length with the normals $\boldsymbol{n}_1$ and $\boldsymbol{n}_3$ from (5.3). Then, the two measured distances correspond to $\Delta^{(1)}_{n1}$ and $\Delta^{(1)}_{n3}$ from (5.4). Without loss of generality, we assume that the origin lies on one wall, i.e. $q_3 = 0$ and $q_1 = w$, where w is the width of the corridor. From (5.2) we obtain

$$\begin{aligned}(\Delta^{(1)}_{n1})^2 &= 4(w - s_{n,y})^2 - 4(w - s_{n,y})\ell\sin\alpha_n + \ell^2,\\ (\Delta^{(1)}_{n3})^2 &= 4s_{n,y}^2 + 4s_{n,y}\ell\sin\alpha_n + \ell^2.\end{aligned} \tag{5.5}$$

For fixed ℓ and some chosen value of w, we can compute the values of $s_{n,y}$ and α_n for the

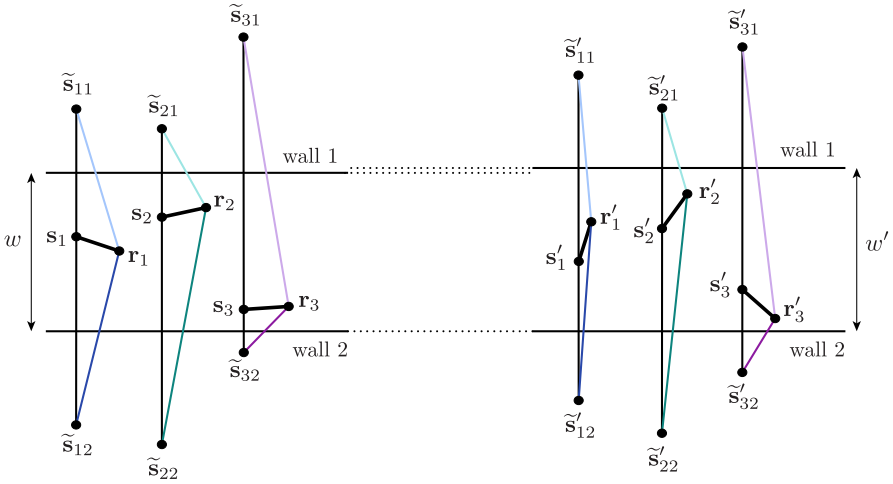


Figure 5.4: Two infinitely long corridors with different widths w and w' and same distances $\boldsymbol{\Delta}^{(1)}_{\mathcal{R}} = \boldsymbol{\Delta}^{(1)}_{\mathcal{R}'}$. Identical measurements are marked with the same color.

given measurements $\Delta^{(1)}_{n1}$ and $\Delta^{(1)}_{n3}$ from (5.5) independently for every $n = 1, \ldots, N$. We can therefore vary the value of w within a certain interval,[1] and reproduce any number of infinitely long corridors of different widths and trajectories, but the same measurements $\Delta^{(1)}_{n1}$ and $\Delta^{(1)}_{n3}$.

5.3.1.3 Parallelogram rooms

Contrary to the case of a collocated source and receiver, with a non-collocated setup we can uniquely distinguish parallelogram rooms given the first-order echoes. We prove this by contradiction. We first take a parallelogram room $\{\mathcal{P}_k\}_{k=1}^4$, set the origin at the intersection of the walls $k = 2$ and $k = 3$, so that $q_2 = q_3 = 0$, and compute $\boldsymbol{\Delta}^{(1)}_{\mathcal{R}}$ for some randomly chosen waypoints and orientations $\{(\boldsymbol{s}_n, \boldsymbol{u}_n)\}_{n=1}^N$. Then we assume that there exists a shear factor that transforms $\{\mathcal{P}_k\}_{k=1}^4$ into a distinct parallelogram $\{\mathcal{P}'_k\}_{k=1}^4$ with waypoints and orientations $\{(\boldsymbol{s}'_n, \boldsymbol{u}'_n)\}_{n=1}^N$, such that $\boldsymbol{\Delta}^{(1)}_{\mathcal{R}'} = \boldsymbol{\Delta}^{(1)}_{\mathcal{R}}$; our goal is to find $\mathcal{R}'$ for which $\mathcal{R}' \neq \mathcal{R}$.

Without loss of generality, let $\{\mathcal{P}_k\}_{k=1}^4$ enclose a rectangle (5.3), and let the normals of $\{\mathcal{P}'_k\}_{k=1}^4$ be

$$\boldsymbol{n}'_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \boldsymbol{n}'_2 = \begin{bmatrix} \cos\varphi' \\ \sin\varphi' \end{bmatrix}, \boldsymbol{n}'_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \boldsymbol{n}'_4 = -\begin{bmatrix} \cos\varphi' \\ \sin\varphi' \end{bmatrix}, \tag{5.6}$$

where $\varphi' \notin \{0, \pi\}$. The measurements in the rectangular and parallelogram rooms are computed from (5.2) by replacing the values of the wall normals with (5.3) and (5.6), respectively. When

[1] The width w cannot take arbitrary real values because the system of quadratic equations does not always have a solution.

the measurements in the two rooms are identical, we have:

$$(q_1 - s_{n,y})^2 - (q_1 - s_{n,y})\ell \sin\alpha_n = (q_1' - s_{n,y}')^2 - (q_1' - s_{n,y}')\ell \sin\alpha_n', \tag{5.7}$$

$$s_{n,y}^2 + s_{n,y}\ell \sin\alpha_n = s_{n,y}'^2 + s_{n,y}'\ell \sin\alpha_n', \tag{5.8}$$

$$s_{n,x}^2 + s_{n,x}\ell \cos\alpha_n = \bar{s}_n^2 + \bar{s}_n\ell \cos(\alpha_n' - \varphi'), \tag{5.9}$$

$$(q_4 - s_{n,x})^2 - (q_4 - s_{n,x})\ell \cos\alpha_n = (q_4' + \bar{s}_n)^2 + (q_4' + \bar{s}_n)\ell \cos(\alpha_n' - \varphi'), \tag{5.10}$$

where $\bar{s}_n = s_{n,x}' \cos\varphi' + s_{n,y}' \sin\varphi'$.

A set of distances measured at the nth waypoint adds four equations of the form (5.7)–(5.10) with three new unknowns $(s_{n,x}', s_{n,y}', \alpha_n')$ to the system. Such an overdetermined system is not necessarily inconsistent; indeed, with $q_1' = q_1$, $q_4' = q_4$, $s_{n,x} = -\bar{s}_n$ and $s_{n,y} = s_{n,y}'$ it has solutions. From (5.7) and (5.8) it follows that $\sin\alpha_n' = \sin\alpha_n$, while (5.9) and (5.10) imply that $\cos\alpha_n = -\cos(\alpha_n' - \varphi')$ for every $n = 1, \ldots, N$. Combined, we have either that

$$\varphi' \in \{0, \pi\}, \quad \alpha_n' = \alpha_n = \frac{\varphi' + \pi}{2} \quad \text{or} \quad \alpha_n' = \pi - \alpha_n = \frac{\varphi'}{2}.$$

We can therefore find a parallelogram that satisfies $\mathbf{\Delta}^{(1)}_{\mathcal{R}'} = \mathbf{\Delta}^{(1)}_{\mathcal{R}}$ if $\alpha_n = \alpha_1$ or $\alpha_n = \pi - \alpha_1$ for every n, which occurs with probability 0 under Assumption 1.

To find the solution to (5.7)–(5.10) in general, we can first determine $s_{n,y}'$ and α_n' from (5.7) and (5.8) up to horizontal and vertical reflections as in Section 5.3.1.1. We can then rewrite (5.9) and (5.10) as quadratic equations in one variable, $s_{n,x}'$, with the leading coefficients of $\cos^2\varphi'$. However, the coefficients in front of the linear term are not identical; their difference is $2q_4' \cos\varphi'$. As two quadratic equations in one variable have a solution when all coefficients are proportional, it follows that $q_4' = 0$ or $\varphi' = \pm\pi/2$.

The above results prove that there do not exist two distinct parallelogram rooms in which we can obtain the same first-order echoes under the Assumption 1. In other words, $\mathbf{\Delta}^{(1)}_{\mathcal{R}'} = \mathbf{\Delta}^{(1)}_{\mathcal{R}}$ implies $\mathcal{R}' = \mathcal{R}$. On the other hand, the presented example for which the system (5.7)–(5.10) is consistent shows that if a robot does not change its orientation, or it rotates only by 180°, then we cannot uniquely recover parallelogram rooms from the measured $\mathbf{\Delta}^{(1)}$.

5.3.1.4 Linear trajectories

Given a collocated source and receiver, we showed in Chapter 2 that for any arbitrary room with K walls and distances measured along any linear trajectory inside the room, there exists another room with the same distances obtained at different collinear waypoints.

Here we derive a result for triangular rooms, $K = 3$. We show that there exist linear trajectories along which the robot can measure the same $\mathbf{\Delta}^{(1)}$ as in some other rooms of different shapes (for the proof see Appendix 5.A). In other words, there exist equivalence classes of rooms and trajectories with respect to $\mathbf{\Delta}^{(1)}$. An example of two rooms from the same class is illustrated in Fig. 5.5. This however does not exclude the existence of linear trajectories for which the measured $\mathbf{\Delta}^{(1)}$ uniquely specifies the room-trajectory configuration. Though our claim for the non-collocated setup is not as strong as for the collocated setup, the provided result shows that collinear waypoints can cause ambiguities in the reconstruction.

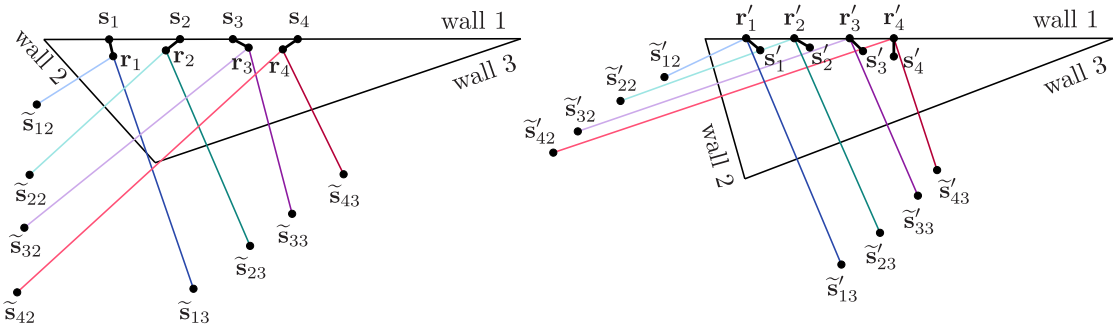


Figure 5.5: Two rooms with different geometries and same distances $\boldsymbol{\Delta}_{\mathcal{R}}^{(1)} = \boldsymbol{\Delta}_{\mathcal{R}'}^{(1)}$. The trajectory in the room on the left is linear. Identical measurements are marked with the same color.

5.3.2 2D: Two receivers

We now assume two receivers and a source mounted on a robot. As illustrated in Fig. 5.6, the source $\boldsymbol{s}_n$ is the midpoint of the segment whose endpoints are receivers $\boldsymbol{r}_n^{(1)}$ and $\boldsymbol{r}_n^{(2)}$,

$$\boldsymbol{r}_n^{(1)} = \boldsymbol{s}_n + \ell \boldsymbol{u}_n, \quad \boldsymbol{r}_n^{(2)} = \boldsymbol{s}_n - \ell \boldsymbol{u}_n, \tag{5.11}$$

for every $n = 1, \ldots, N$.

The distances $\Delta_{nk}^{(1)}$ and $\Delta_{nk}^{(2)}$ that correspond to the propagation times of the first-order echoes recorded by $\boldsymbol{r}_n^{(1)}$ and $\boldsymbol{r}_n^{(2)}$ are given by

$$(\Delta_{nk}^{(1)})^2 = \|\widetilde{\boldsymbol{s}}_{nk} - \boldsymbol{r}_n^{(1)}\|^2 = 4d_{nk}^2 - 4d_{nk}\ell \boldsymbol{n}_k^\top \boldsymbol{u}_n + \ell^2, \tag{5.12}$$

$$(\Delta_{nk}^{(2)})^2 = \|\widetilde{\boldsymbol{s}}_{nk} - \boldsymbol{r}_n^{(2)}\|^2 = 4d_{nk}^2 + 4d_{nk}\ell \boldsymbol{n}_k^\top \boldsymbol{u}_n + \ell^2. \tag{5.13}$$

Instead of studying (5.12) and (5.13) directly, we refer to their sum and difference,

$$(\Delta_{nk}^{(2)})^2 + (\Delta_{nk}^{(1)})^2 = 8d_{nk}^2 + 2\ell^2, \tag{5.14}$$

$$(\Delta_{nk}^{(2)})^2 - (\Delta_{nk}^{(1)})^2 = 8d_{nk}\ell \boldsymbol{n}_k^\top \boldsymbol{u}_n. \tag{5.15}$$

As we will see, (5.14) and (5.15) lead to forms from Chapter 2 and facilitate the analysis.

Let the columns of $\boldsymbol{S} \in \mathbb{R}^{D\times N}$ and $\boldsymbol{U} \in \mathbb{R}^{D\times N}$ contain the robot's waypoints and orientations, respectively, $\boldsymbol{S} \overset{\text{def}}{=} [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N], \boldsymbol{U} \overset{\text{def}}{=} [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N]$. Let the columns of $\boldsymbol{N} \in \mathbb{R}^{D\times K}$ be the outward looking normal vectors of walls, $\boldsymbol{N} \overset{\text{def}}{=} [\boldsymbol{n}_1, \ldots, \boldsymbol{n}_K]$, and $\boldsymbol{q} \overset{\text{def}}{=} [q_1, \ldots, q_K]^\top \in \mathbb{R}^K$ be the vector of distances between the walls and the origin. Furthermore, recall from Chapter 2 that the point-to-plane distance matrix (PPDM) $\boldsymbol{D}$ is defined as $\boldsymbol{D} = \boldsymbol{1}\boldsymbol{q}^\top - \boldsymbol{S}^\top \boldsymbol{N}$, (2.2).

With this notation in hand, and by using the apostrophe $'$ to denote the corresponding vectors and matrices in a room-trajectory configuration $\mathcal{R}'$, we can rewrite (5.14) and (5.15) in matrix form and state the following lemma:

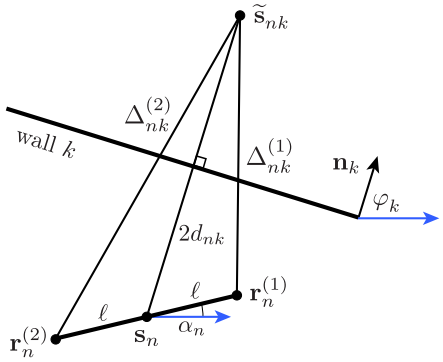


Figure 5.6: One source $\boldsymbol{s}_n$ and two receivers $\boldsymbol{r}_n^{(1)}$ and $\boldsymbol{r}_n^{(2)}$ provide the distance measurements $\Delta_{nk}^{(1)}$ and $\Delta_{nk}^{(2)}$. They are proportional to the propagation times of the pulses produced by $\boldsymbol{s}_n$, reflected from the wall k, and recorded by $\boldsymbol{r}_n^{(1)}$ and $\boldsymbol{r}_n^{(2)}$.

LEMMA 5.2 Two room-trajectory configurations $\mathcal{R}$ and $\mathcal{R}'$ in 2D generate the same set of distances $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^2$ and $\{\boldsymbol{\Delta}_{\mathcal{R}'}^{(m)}\}_{m=1}^2$ if and only if

$$\boldsymbol{D} = \boldsymbol{D}' \quad \text{and} \tag{5.16}$$

$$\overline{\boldsymbol{U}}^\top \overline{\boldsymbol{N}} = \boldsymbol{0}, \tag{5.17}$$

where

$$\overline{\boldsymbol{U}} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{U} \\ -\boldsymbol{U}' \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_1 & \dots & \boldsymbol{u}_N \\ -\boldsymbol{u}_1' & \dots & -\boldsymbol{u}_N' \end{bmatrix}, \tag{5.18}$$

$$\overline{\boldsymbol{N}} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{N} \\ \boldsymbol{N}' \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_1 & \dots & \boldsymbol{n}_K \\ \boldsymbol{n}_1' & \dots & \boldsymbol{n}_K' \end{bmatrix}. \tag{5.19}$$

PROOF. Assume that for some $\mathcal{R}$ and $\mathcal{R}'$ we have $\boldsymbol{\Delta}_{\mathcal{R}}^{(m)} = \boldsymbol{\Delta}_{\mathcal{R}'}^{(m)}$ for $m \in \{1, 2\}$. Equivalently,

$$\begin{aligned} \boldsymbol{\Delta}_{\mathcal{R}}^{(1)} + \boldsymbol{\Delta}_{\mathcal{R}}^{(2)} &= \boldsymbol{\Delta}_{\mathcal{R}'}^{(1)} + \boldsymbol{\Delta}_{\mathcal{R}'}^{(2)}, \\ \boldsymbol{\Delta}_{\mathcal{R}}^{(1)} - \boldsymbol{\Delta}_{\mathcal{R}}^{(2)} &= \boldsymbol{\Delta}_{\mathcal{R}'}^{(1)} - \boldsymbol{\Delta}_{\mathcal{R}'}^{(2)}. \end{aligned} \tag{5.20}$$

By substituting (5.14) and (5.15) into (5.20), and together with $d_{nk} \geq 0$ for every $n = 1, \dots, N$ and $k = 1, \dots, K$, we obtain (5.16) and (5.17). ■

As in Chapter 2 we already studied (5.16) in detail, the advantage of rewriting (5.12) and (5.13) as (5.14) and (5.15) is now evident. In Chapter 2 we proved that two room-trajectory configurations have the same PPDM if and only if $\overline{\boldsymbol{S}}^\top \overline{\boldsymbol{N}} = \boldsymbol{0}$, where

$$\overline{\boldsymbol{S}} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{S} \\ -\boldsymbol{S}' \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_1 & \dots & \boldsymbol{s}_N \\ -\boldsymbol{s}_1' & \dots & -\boldsymbol{s}_N' \end{bmatrix}.$$

The two conditions (5.16) and (5.17) are similar: (5.16) requires that the columns of $\overline{\boldsymbol{S}}$ are in the nullspace of $\overline{\boldsymbol{N}}^\top$, while (5.17) requires that the columns of $\overline{\boldsymbol{U}}$ are in the nullspace of $\overline{\boldsymbol{N}}^\top$.

To check the existence of configurations that are not uniquely specified by $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^2$, we first need to impose linear dependencies among the columns of $\overline{\boldsymbol{N}}^\top$, and then find the waypoints $\overline{\boldsymbol{S}}$ and orientations $\overline{\boldsymbol{U}}$ that lie in the nullspace. By letting $r = \text{rank}(\overline{\boldsymbol{N}})$, we select r linearly independent columns of $\overline{\boldsymbol{N}}^\top$ and assume that the remaining columns are their linear combinations. In addition to these linear dependencies, the columns of $\overline{\boldsymbol{N}}^\top$ are subject to non-linear constraints; recall that the rows of $\overline{\boldsymbol{N}}^\top$ contain unit normals $\boldsymbol{n}_k$ and $\boldsymbol{n}'_k$ which introduce non-linear relationships among the columns.

The solutions to those linear and non-linear equations determine the room-trajectory configurations that are not uniquely specified by $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^2$. The absence of solutions is therefore desired; it proves that two distinct room-trajectory configurations cannot have the same $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^2$.

In Chapter 2 we provided a thorough analysis of the nullspace of $\overline{\boldsymbol{N}}^\top$ and identified all configurations that satisfy (5.16), i.e. $\overline{\boldsymbol{S}}^\top\overline{\boldsymbol{N}} = \boldsymbol{0}$. More precisely, we first found the basis of the nullspace of $\overline{\boldsymbol{N}}^\top$ by solving the aforementioned system of linear and non-linear constraints. Then, we determined $\overline{\boldsymbol{S}}$ such that its columns lie in the nullspace; as there are no other constraints on $\overline{\boldsymbol{S}}$, the solution always existed. Now we have to incorporate an additional condition (5.17) to the previous analysis. It states that the columns of $\overline{\boldsymbol{U}}$ are also in the nullspace of $\overline{\boldsymbol{N}}^\top$, but contrary to (5.16), the normalization constraint on the columns of $\overline{\boldsymbol{U}}$ narrows the solution space.

In the following we show that the system of (5.16) and (5.17) either does not have a solution or it happens with probability 0 under Assumption 1, and hence, the room-trajectory configurations are uniquely defined by the first-order echoes recorded by two receivers. As the nullspace of $\overline{\boldsymbol{N}}^\top$ is already computed for every $r = \{1, 2, 3\}$ in Section 2.4, we start from that result and focus our analysis on incorporating the condition $\overline{\boldsymbol{U}}^\top\overline{\boldsymbol{N}} = \boldsymbol{0}$.

5.3.2.1 Rank-1: Infinitely long corridors

We assume that $\text{rank}(\overline{\boldsymbol{N}}) = 1$, so that every column of $\overline{\boldsymbol{N}}^\top$ is a scaled version of the first column,

$$\begin{bmatrix} \sin\varphi'_k \\ \cos\varphi_k \\ \sin\varphi_k \end{bmatrix} = \begin{bmatrix} a \\ b \\ 0 \end{bmatrix} \cos\varphi'_k. \tag{5.21}$$

Recall from Section 2.4.2 that solutions to (5.21) give infinitely long corridors which are uniquely defined by $\overline{\boldsymbol{S}}^\top\overline{\boldsymbol{N}} = \boldsymbol{0}$, but their waypoints are not. The nullspace of $\overline{\boldsymbol{N}}^\top$ is spanned by three vectors,

$$\boldsymbol{e}_1 = \begin{bmatrix} -a & 1 & 0 & 0 \end{bmatrix}^\top, \boldsymbol{e}_2 = \begin{bmatrix} -b & 0 & 1 & 0 \end{bmatrix}^\top, \boldsymbol{e}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top,$$

so for the columns of $\overline{\boldsymbol{U}}$ to lie in the nullspace of $\overline{\boldsymbol{N}}^\top$, they have to be of the form

$$\begin{bmatrix} \boldsymbol{u}_n \\ -\boldsymbol{u}'_n \end{bmatrix} = \begin{bmatrix} \cos\alpha_n \\ \sin\alpha_n \\ -\cos\alpha'_n \\ -\sin\alpha'_n \end{bmatrix} = \lambda_1\boldsymbol{e}_1 + \lambda_2\boldsymbol{e}_2 + \lambda_3\boldsymbol{e}_3, \tag{5.22}$$

where λ_1, λ_2 and $\lambda_3 \in \mathbb{R}$. By setting $a = 0$ and $b = 1$, we fix the rotation between two equivalent corridors, and together with (5.22) we obtain

$$\cos\alpha_n = \cos\alpha'_n, \tag{5.23}$$

which implies $\alpha_n = \alpha'_n$ or $\alpha_n = \pi - \alpha'_n$.

The columns of $\overline{\boldsymbol{S}}$ also satisfy

$$\begin{bmatrix} \boldsymbol{s}_n \\ -\boldsymbol{s}'_n \end{bmatrix} = \mu_1 \boldsymbol{e}_1 + \mu_2 \boldsymbol{e}_2 + \mu_3 \boldsymbol{e}_3, \tag{5.24}$$

where μ_1, μ_2 and $\mu_3 \in \mathbb{R}$. This further implies that the coordinates of $\{\boldsymbol{s}_n\}_{n=1}^N$ and the x coordinates of $\{\boldsymbol{s}'_n\}_{n=1}^N$ can be chosen arbitrarily, while the y coordinates of $\{\boldsymbol{s}'_n\}_{n=1}^N$ are computed from (5.24).

These results prove that two receivers and one source cannot be uniquely localized in infinitely long corridors: if 1) we translate the robot along the line parallel to the walls by (5.24), or 2) we reflect the robot across the line perpendicular to the walls by (5.23), we obtain the same first-order echoes. Note however that 1) and 2) are valid for a robot with any number of sources and receivers. A mobile setup without any external sensor or a fixed landmark suffers from localization ambiguities in infinitely long corridors for any $M \geq 1$.

5.3.2.2 Rank-2: Parallelogram rooms

Let $\text{rank}(\overline{\boldsymbol{N}}) = 2$, so that

$$\begin{bmatrix} \cos\varphi_k \\ \sin\varphi_k \end{bmatrix} = \begin{bmatrix} a & b \\ 0 & d \end{bmatrix} \begin{bmatrix} \cos\varphi'_k \\ \sin\varphi'_k \end{bmatrix}. \tag{5.25}$$

From Section 2.4.1 we know that (5.25) defines infinitely many distinct parallelograms that satisfy $\overline{\boldsymbol{S}}^\top \overline{\boldsymbol{N}} = \boldsymbol{0}$, so the condition (5.16) is not sufficient to uniquely specify room-trajectory configurations. A basis for the nullspace $\overline{\boldsymbol{N}}^\top$ is

$$\boldsymbol{e}_1 = \begin{bmatrix} -a & -b & 1 & 0 \end{bmatrix}^\top, \boldsymbol{e}_2 = \begin{bmatrix} 0 & -d & 0 & 1 \end{bmatrix}^\top,$$

so the columns of $\overline{\boldsymbol{U}}$ are related as

$$\begin{bmatrix} \boldsymbol{u}_n \\ -\boldsymbol{u}'_n \end{bmatrix} = \boldsymbol{e}_1 \lambda_1 + \boldsymbol{e}_2 \lambda_2, \tag{5.26}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}$. We can rewrite (5.26) as

$$\begin{bmatrix} \cos\alpha_n \\ \sin\alpha_n \end{bmatrix} = \begin{bmatrix} a & 0 \\ b & d \end{bmatrix} \begin{bmatrix} \cos\alpha'_n \\ \sin\alpha'_n \end{bmatrix}, \tag{5.27}$$

for $n = 1, \ldots, N$. For some arbitrarily chosen a, b and d, the solution to (5.27) shows that we can find two distinct parallelograms with the same $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^2$ only if $\boldsymbol{u}_n$ and $\boldsymbol{u}'_n$ happen with probability 0 under Assumption 1. More precisely, from (5.27) we can compute the robot's orientation at two waypoints (e.g. α_1 and α_2), while for $n = 3, \ldots, N$ we must have $\alpha_n = \alpha_1$, $\alpha_n = \alpha_1 + \pi$, $\alpha_n = \alpha_2$ or $\alpha_n = \alpha_2 + \pi$ for the system to be consistent. This happens with probability 0 under Assumption 1.

5.3.2.3 Rank-3: Linear trajectories

For $\text{rank}(\overline{\boldsymbol{N}}) = 3$ we assume that one column of $\overline{\boldsymbol{N}}^\top$ is a linear combination of the remaining independent columns,

$$\cos\varphi_k = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} \sin\varphi_k \\ \cos\varphi'_k \\ \sin\varphi'_k \end{bmatrix}. \tag{5.28}$$

As shown in Section 2.4.3, (5.28) implies that we can find infinitely many distinct rooms that satisfy $\overline{\boldsymbol{S}}^\top \overline{\boldsymbol{N}} = \boldsymbol{0}$ when the corresponding robot's trajectory is linear.

The nullspace of $\overline{\boldsymbol{N}}^\top$ is spanned by one vector $\boldsymbol{e}_1 = \begin{bmatrix} -b & -c & 1 & -a \end{bmatrix}^\top$, so the columns of $\overline{\boldsymbol{U}}$ are of the form $[\boldsymbol{u}_n, -\boldsymbol{u}'_n]^\top = \lambda \boldsymbol{e}_1$, where $\lambda \in \mathbb{R}$. Rewritten as

$$\begin{bmatrix} \sin\alpha_n \\ \cos\alpha'_n \\ \sin\alpha'_n \end{bmatrix} = \begin{bmatrix} b \\ c \\ -a \end{bmatrix} \cos\alpha'_n,$$

we recognize the form of (5.21) for which the solution exists only for $\alpha_n = \alpha_1$ or $\alpha_n = \alpha_1 + \pi$, and $\alpha'_n = \alpha'_1$ or $\alpha'_n = \alpha'_1 + \pi$, for every $n = 1, \ldots, N$.

On the one hand, these constant values for the robot's orientation occur with probability 0 under Assumption 1 and prove that two distinct rooms have two distinct sets of measurements $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^2$ obtained at linear trajectories with arbitrarily chosen robot's orientations. On the other hand, the result shows that with one source and two receivers we cannot uniquely reconstruct rooms of arbitrary shapes if the robot moves in linear trajectory and it does not change its direction along the path.

The conducted analysis proves the following theorem:

THEOREM 5.3 Consider a mobile robot with one acoustic source whose waypoints and orientations are picked independently at random in a 2D room. Two is the minimal number of receivers for which equivalence classes of room-trajectory configurations with respect to first-order echoes recorded by these receivers have measure 0.

In Table 5.1 we summarize all the results on uniqueness in room reconstruction and robot's localization in 2D. The table illustrates the outcome of moving from a collocated to a non-collocated setup with one source and one receiver, and the advantage of introducing a second receiver.

5.4 Three-dimensional setup

In 3D, the walls are represented as planes with normals parametrized in spherical coordinates, $\boldsymbol{n}_k = [\sin\theta_k \cos\varphi_k,\ \sin\theta_k \sin\varphi_k,\ \cos\theta_k]^\top$ where $\varphi_k \in [0, 2\pi)$ and $\theta_k \in [0, \pi)$.

We start with $M = 2$ and investigate if (5.16) and (5.17) are sufficient for a unique reconstruction of room-trajectory configurations in 3D. While this is the minimal number of receivers

Table 5.1: Uniqueness in room reconstruction and robot's localization in 2D for different M. Symbols ✓ and ✗ indicate if the reconstruction is unique or not, respectively.

	$M = 1^*$		$M = 1$		$M = 2$	
	room	loc.	room	loc.	room	loc.
infinitely long corridors, $r = 1$	✓	✗	✗	✗	✓	✗
parallelogram rooms, $r = 2$	✗	✗	✓	✗	✓	✓
linear trajectories, $r = 3$	✗	✗	✗	✗	✓	✓

* Collocated source and receiver; results derived in Chapter 2.

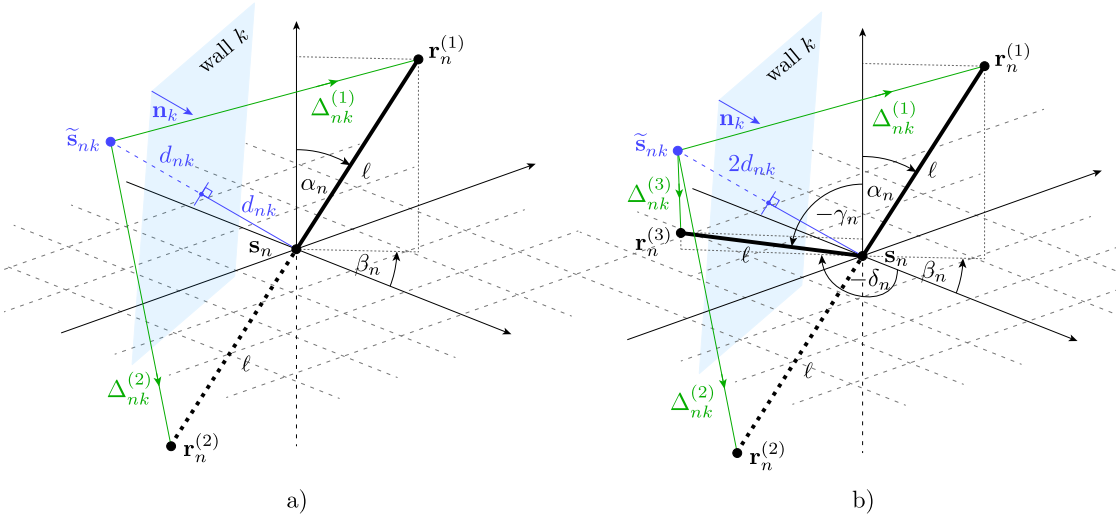


Figure 5.7: a) One source $\boldsymbol{s}_n$ and two receivers $\{\boldsymbol{r}_n^{(m)}\}_{m=1}^2$ provide measurements $\{\boldsymbol{\Delta}^{(m)}\}_{m=1}^2$ which are not sufficient for a unique reconstruction of room-trajectory configurations in 3D. b) One source $\boldsymbol{s}_n$ and three receivers $\{\boldsymbol{r}_n^{(m)}\}_{m=1}^3$ ensure a unique reconstruction of room-trajectory configurations in 3D.

that ensures uniqueness in 2D, Section 5.4.1 proves that this is not the case in 3D. We increase the number of receivers to $M = 3$ in Section 5.4.2 and show that this resolves all ambiguities in the reconstruction.

5.4.1 3D: Two receivers

We take the setup from Section 5.3.2 and place it in a 3D room. The receiver's locations are given by (5.11), where $\boldsymbol{u}_n = [\sin\alpha_n \cos\beta_n, \sin\alpha_n \sin\beta_n, \cos\alpha_n]^\top$. An example of one wall and one waypoint is illustrated in Fig. 5.7a.

To confirm that the case of $M = 2$ does not uniquely specify room-trajectory configurations

from $\{\boldsymbol{\Delta}^{(m)}\}_{m=1}^{2}$ defined by (5.12) and (5.13), we consider $\operatorname{rank}(\overline{\boldsymbol{N}}) = 2$ and assume that

$$\begin{bmatrix} \cos\theta'_k \\ \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \\ \cos\theta_k \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \\ 0 & f \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sin\theta'_k \cos\varphi'_k \\ \sin\theta'_k \sin\varphi'_k \end{bmatrix}. \tag{5.29}$$

In Section 2.5.2, we show that we can find infinitely many "rooms" that satisfy (5.29). These "rooms" are described by two pairs of parallel walls without the ceiling and the floor. The nullspace of $\overline{\boldsymbol{N}}$ is spanned by four vectors

$$\boldsymbol{e}_1 = \begin{bmatrix} -a & -b & 1 & 0 & 0 & 0 \end{bmatrix}^\top, \qquad \boldsymbol{e}_2 = \begin{bmatrix} -c & -d & 0 & 1 & 0 & 0 \end{bmatrix}^\top,$$
$$\boldsymbol{e}_3 = \begin{bmatrix} 0 & -f & 0 & 0 & 1 & 0 \end{bmatrix}^\top, \qquad \boldsymbol{e}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top,$$

so the columns of $\overline{\boldsymbol{U}}$ are

$$\begin{bmatrix} \boldsymbol{u}_n \\ -\boldsymbol{u}'_n \end{bmatrix} = \lambda_1 \boldsymbol{e}_1 + \lambda_2 \boldsymbol{e}_2 + \lambda_3 \boldsymbol{e}_3 + \lambda_4 \boldsymbol{e}_4, \tag{5.30}$$

where $\lambda_1, \lambda_2, \lambda_3$ and $\lambda_4 \in \mathbb{R}$. We can further rewrite (5.30) as

$$\begin{bmatrix} \sin\alpha_n \cos\beta_n \\ \sin\alpha_n \sin\beta_n \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \sin\alpha'_n \cos\beta'_n \\ \sin\alpha'_n \sin\beta'_n \\ \cos\alpha'_n \\ \cos\alpha_n \end{bmatrix}, \quad \text{where} \quad \boldsymbol{T} = \begin{bmatrix} c & 0 & 0 & -a \\ d & f & 0 & -b \end{bmatrix}, \tag{5.31}$$

and obtain two equations with four unknown angles for every $n = 1, \ldots, N$. In such an underdetermined system, we can pick any values for $\{\alpha_n, \beta_n\}_{n=1}^{N}$ and compute $\{\alpha'_n, \beta'_n\}_{n=1}^{N}$. In a "room" without ceiling and floor we can therefore pick a robot trajectory and orientations at random, measure the distances $\{\boldsymbol{\Delta}^{(m)}\}_{m=1}^{2}$, and find a multitude of sheared "rooms" without ceiling and floor with the same $\{\boldsymbol{\Delta}^{(m)}\}_{m=1}^{2}$ from (5.29) and (5.31). The reconstruction with one source and two receivers is thus not unique in 3D.

5.4.2 3D: Three receivers

Consider a setup from the previous section enhanced by a new noncollinear receiver, as illustrated in Fig. 5.7b. We denote its location by $\boldsymbol{r}_n^{(3)}$ and together with (5.11) have

$$\boldsymbol{r}_n^{(1)} = \boldsymbol{s}_n + \ell \boldsymbol{u}_n, \quad \boldsymbol{r}_n^{(2)} = \boldsymbol{s}_n - \ell \boldsymbol{u}_n, \quad \boldsymbol{r}_n^{(3)} = \boldsymbol{s}_n + \ell \boldsymbol{v}_n.$$

Here, $\boldsymbol{v}_n$ is a unit vector determining the orientation of $\boldsymbol{r}_n^{(3)}$, $\boldsymbol{v}_n = [\sin\gamma_n \cos\delta_n, \sin\gamma_n \sin\delta_n, \cos\gamma_n]^\top$, such that for every $n = 1, \ldots, N$, we have

$$\boldsymbol{u}_n^\top \boldsymbol{v}_n = C, \quad \text{where } C \in \mathbb{R}. \tag{5.32}$$

In addition to $\Delta_{nk}^{(1)}$ and $\Delta_{nk}^{(2)}$ from (5.12) and (5.13), we obtain the measurements of $\Delta_{nk}^{(3)}$ that correspond to the times of flight of the first-order echoes between $\boldsymbol{s}_n$ and $\boldsymbol{r}_n^{(3)}$,

$$(\Delta_{nk}^{(3)})^2 = \|\widetilde{\boldsymbol{s}}_{nk} - \boldsymbol{r}_n^{(3)}\|^2 = 4d_{nk}^2 + \ell^2 - 4d_{nk}\ell \boldsymbol{n}_k^\top \boldsymbol{v}_n. \tag{5.33}$$

By combining (5.33) with (5.16) and (5.17), we can establish the following lemma:

LEMMA 5.4 Two room-trajectory configurations $\mathcal{R}$ and $\mathcal{R}'$ in 3D generate the same set of distances $\{\boldsymbol{\Delta}_{\mathcal{R}}^{(m)}\}_{m=1}^{3}$ and $\{\boldsymbol{\Delta}_{\mathcal{R}'}^{(m)}\}_{m=1}^{3}$ if and only if

$$\begin{aligned} \boldsymbol{D} &= \boldsymbol{D}', \\ \overline{\boldsymbol{U}}^{\top}\overline{\boldsymbol{N}} &= \boldsymbol{0}, \quad \text{and} \\ \overline{\boldsymbol{V}}^{\top}\overline{\boldsymbol{N}} &= \boldsymbol{0}, \end{aligned} \tag{5.34}$$

where $\boldsymbol{D}$ is defined by (2.2), $\overline{\boldsymbol{U}}$ by (5.18), $\overline{\boldsymbol{N}}$ by (5.19) and

$$\overline{\boldsymbol{V}} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{V} \\ -\boldsymbol{V}' \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_1 & \dots & \boldsymbol{v}_N \\ -\boldsymbol{v}_1' & \dots & -\boldsymbol{v}_N' \end{bmatrix}.$$

PROOF. The proof follows from (5.33) and (5.16). ■

A new constraint (5.34) specifies that the columns of $\overline{\boldsymbol{V}}$ have to be in the nullspace of $\overline{\boldsymbol{N}}^{\top}$, so the existence of room-trajectory configurations that satisfy Lemma 5.4 also relies on the nullspace of $\overline{\boldsymbol{N}}^{\top}$. We can reuse the analysis from Section 2.5, and extend it by requiring $\overline{\boldsymbol{U}}^{\top}\overline{\boldsymbol{N}} = \boldsymbol{0}$ and $\overline{\boldsymbol{V}}^{\top}\overline{\boldsymbol{N}} = \boldsymbol{0}$.

5.4.2.1 Rank-1: Infinitely long and tall corridors

When $\operatorname{rank}(\overline{\boldsymbol{N}}) = 1$ in 3D, five columns of $\overline{\boldsymbol{N}}^{\top}$ are scaled version of a single column,

$$\begin{bmatrix} \sin\theta_k' \sin\varphi_k' \\ \cos\theta_k' \\ \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \\ \cos\theta_k \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ 0 \\ 0 \end{bmatrix} \sin\theta_k' \cos\varphi_k'. \tag{5.35}$$

Normals that satisfy (5.35) define rooms with parallel walls to which we refer as infinitely long and tall corridors (see Section 2.5.1). They are uniquely defined by $\overline{\boldsymbol{S}}^{\top}\overline{\boldsymbol{N}}$, but their waypoints are not. Such a result holds even if we arbitrarily increase the number of receivers. To see that, let us find the nullspace of $\overline{\boldsymbol{N}}$. It is spanned by five vectors

$$\boldsymbol{e}_1 = \begin{bmatrix} -a & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^{\top}, \quad \boldsymbol{e}_2 = \begin{bmatrix} -b & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^{\top}, \quad \boldsymbol{e}_3 = \begin{bmatrix} -c & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^{\top},$$
$$\boldsymbol{e}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^{\top}, \quad \boldsymbol{e}_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^{\top},$$

so the columns of $\overline{\boldsymbol{U}}$ have to be of the form

$$\begin{bmatrix} \boldsymbol{u}_n \\ -\boldsymbol{u}_n' \end{bmatrix} = \lambda_1\boldsymbol{e}_1 + \lambda_2\boldsymbol{e}_2 + \lambda_3\boldsymbol{e}_3 + \lambda_4\boldsymbol{e}_4 + \lambda_5\boldsymbol{e}_5, \tag{5.36}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ and $\lambda_5 \in \mathbb{R}$. From (5.36) we have

$$c \sin \alpha'_n \cos \beta'_n = \sin \alpha_n \cos \beta_n + a \sin \alpha_n \sin \beta_n + b \cos \alpha_n. \tag{5.37}$$

By setting $a = b = 0$ and $c = 1$ we only fix the rotation of $\mathcal{R}$ and from (5.37) we get that

$$\begin{aligned} \sin \alpha_n = \sin \alpha'_n, \cos \beta_n = \cos \beta'_n, \quad \text{or} \\ \sin \alpha_n = -\sin \alpha'_n, \cos \beta_n = -\cos \beta'_n. \end{aligned} \tag{5.38}$$

There are four solutions to (5.38):

$$\begin{aligned} \boldsymbol{u}'_n &= \begin{bmatrix} \sin \alpha_n \cos \beta_n & \sin \alpha_n \sin \beta_n & \cos \alpha_n \end{bmatrix}^\top, \quad \text{or} \\ \boldsymbol{u}'_n &= \begin{bmatrix} \sin \alpha_n \cos \beta_n & \sin \alpha_n \sin \beta_n & -\cos \alpha_n \end{bmatrix}^\top, \quad \text{or} \\ \boldsymbol{u}'_n &= \begin{bmatrix} \sin \alpha_n \cos \beta_n & -\sin \alpha_n \sin \beta_n & \cos \alpha_n \end{bmatrix}^\top, \quad \text{or} \\ \boldsymbol{u}'_n &= \begin{bmatrix} \sin \alpha_n \cos \beta_n & -\sin \alpha_n \sin \beta_n & -\cos \alpha_n \end{bmatrix}^\top, \end{aligned} \tag{5.39}$$

which are related by reflection. Moreover, the columns of $\overline{\boldsymbol{V}}$ have to be of the same form as (5.36), so we can repeat the derivation from (5.36) to (5.39) for $\overline{\boldsymbol{V}}$. The four values of $\boldsymbol{v}'_n$ are analogous to the expressions in (5.39), where α_n and β_n are replaced by γ_n and δ_n, respectively. Therefore, we can find different setups that satisfy (5.16), (5.17), (5.34) and (5.32) for every n, which proves that three receivers and one source cannot uniquely recover robot's orientations in infinitely long and tall corridors.

This result is not surprising; equivalently to the 2D case, this nonuniqueness occurs for any setup regardless of the number of sources and receivers: 1) we can always translate the setup along the plane parallel to the walls, and 2) we can always reflect the setup across the planes perpendicular to the walls, and obtain the same first-order echoes.

5.4.2.2 Rank-2: Parallelepipeds and prisms without bases

We considered $\operatorname{rank}(\overline{\boldsymbol{N}}) = 2$ in Section 5.4.1 to prove that one source and two receivers are not sufficient to uniquely reconstruct room-trajectory configurations from the first-order echoes. Here we show that the additional constraint (5.34) resolves this ambiguity.

For (5.34) to be satisfied for $r = 2$, we assume that

$$\begin{bmatrix} \sin \gamma_n \cos \delta_n \\ \sin \gamma_n \sin \delta_n \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \sin \gamma'_n \cos \delta'_n \\ \sin \gamma'_n \sin \delta'_n \\ \cos \gamma'_n \\ \cos \gamma_n \end{bmatrix}, \tag{5.40}$$

where $\boldsymbol{T}$ is given in (5.31).

Moreover, the locations of the third receivers in the two rooms satisfy

$$\boldsymbol{u}_n^\top \boldsymbol{v}_n = C \quad \text{and} \quad \boldsymbol{u}_n'^\top \boldsymbol{v}'_n = C', \tag{5.41}$$

so they have one degree of freedom for every n in every room.

From (5.31), (5.40) and (5.41), we then obtain 10 equations with eight unknowns for every $n = 1, \ldots, N$. This overdetermined system of equations is inconsistent for arbitrarily chosen $\boldsymbol{u}_n$ and $\boldsymbol{u}'_n$, and it has a solution only with probability 0 under Assumption 1.

5.4.2.3 Rank-3: Miscellaneous geometries

For $\operatorname{rank}(\overline{\boldsymbol{N}}) = 3$, we have

$$\begin{bmatrix} \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \\ \cos\theta_k \end{bmatrix} = \begin{bmatrix} a & b & c \\ 0 & e & f \\ 0 & 0 & i \end{bmatrix} \begin{bmatrix} \sin\theta'_k \cos\varphi'_k \\ \sin\theta'_k \sin\varphi'_k \\ \cos\theta'_k \end{bmatrix}. \tag{5.42}$$

This case is studied in Section 2.5.4, where we obtain that rooms of miscellaneous geometries, including the practically relevant shoebox rooms and all rooms with less than six walls, are not uniquely defined by (5.16).

The nullspace of $\overline{\boldsymbol{N}}$ is spanned by three vectors

$$\boldsymbol{e}_1 = \begin{bmatrix} -a & -b & -c & 1 & 0 & 0 \end{bmatrix}^\top, \boldsymbol{e}_2 = \begin{bmatrix} 0 & -e & -f & 0 & 1 & 0 \end{bmatrix}^\top, \boldsymbol{e}_3 = \begin{bmatrix} 0 & 0 & -i & 0 & 0 & 1 \end{bmatrix}^\top,$$

so the columns of $\overline{\boldsymbol{U}}$ are of the form

$$\begin{bmatrix} \boldsymbol{u}_n \\ -\boldsymbol{u}'_n \end{bmatrix} = \lambda_1 \boldsymbol{e}_1 + \lambda_2 \boldsymbol{e}_2 + \lambda_3 \boldsymbol{e}_3,$$

where λ_1, λ_2 and $\lambda_3 \in \mathbb{R}$. This can further be written as

$$\begin{bmatrix} \sin\alpha_n \cos\beta_n \\ \sin\alpha_n \sin\beta_n \\ \cos\alpha_n \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ b & e & 0 \\ c & f & i \end{bmatrix} \begin{bmatrix} \sin\alpha'_n \cos\beta'_n \\ \sin\alpha'_n \sin\beta'_n \\ \cos\alpha'_n \end{bmatrix},$$

which is analogous to (5.42) and solved in Section 2.5.4. There are two results: α_n is a function of arbitrarily chosen β_n, or, α_n is chosen arbitrarily, while β_n is a constant. In any case, either one of the angles is constrained, which occurs with probability 0 under Assumption 1.

5.4.2.4 Rank-4: Planar trajectories

The case of $\operatorname{rank}(\overline{\boldsymbol{N}}) = 4$ assumes

$$\begin{bmatrix} \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \end{bmatrix} \begin{bmatrix} \sin\theta'_k \cos\varphi'_k \\ \sin\theta'_k \sin\varphi'_k \\ \cos\theta'_k \\ \cos\theta_k \end{bmatrix}. \tag{5.43}$$

In Section 2.5.6 we solve (5.43) and find infinitely many distinct rooms that satisfy (5.16) if the robot's trajectory is planar.

The nullspace of $\overline{\boldsymbol{N}}^\top$ is spanned by two vectors

$$\boldsymbol{e}_1 = \begin{bmatrix} -a & -b & -c & 1 & 0 & -d \end{bmatrix}^\top, \boldsymbol{e}_2 = \begin{bmatrix} -e & -f & -g & 0 & 1 & -h \end{bmatrix}^\top,$$

so the columns of $\overline{\boldsymbol{U}}$ satisfy

$$\begin{bmatrix} \boldsymbol{u}_n \\ -\boldsymbol{u}'_n \end{bmatrix} = \lambda_1 \boldsymbol{e}_1 + \lambda_2 \boldsymbol{e}_2,$$

where λ_1 and $\lambda_2 \in \mathbb{R}$, or equivalently,

$$\begin{bmatrix} \cos\alpha'_n \\ \sin\alpha_n \cos\beta_n \\ \sin\alpha_n \sin\beta_n \\ \cos\alpha_n \end{bmatrix} = \begin{bmatrix} -d & -h \\ a & e \\ b & f \\ c & g \end{bmatrix} \begin{bmatrix} \sin\alpha'_n & \cos\beta'_n \\ \sin\alpha'_n & \sin\beta'_n \end{bmatrix}. \tag{5.44}$$

To solve (5.44), we refer to the analogous form (5.29) solved in Section 2.5.2. It has two distinct solutions: α_n and β_n are constants, or, α_n is a function of an arbitrarily chosen β_n. Since $\boldsymbol{u}_n$ cannot be chosen at random for every n, we conclude that this case happens with probability 0 under Assumption 1 even without the third receiver.

5.4.2.5 Rank-5: Linear trajectories

Finally, let $\operatorname{rank}(\overline{\boldsymbol{N}}) = 5$, so that one column of $\overline{\boldsymbol{N}}^\top$ is a linear combination of the remaining independent columns,

$$\cos\theta_k = \begin{bmatrix} a & b & c & d & e \end{bmatrix} \begin{bmatrix} \sin\theta_k \cos\varphi_k \\ \sin\theta_k \sin\varphi_k \\ \sin\theta'_k \cos\varphi'_k \\ \sin\theta'_k \sin\varphi'_k \\ \cos\theta'_k \end{bmatrix}. \tag{5.45}$$

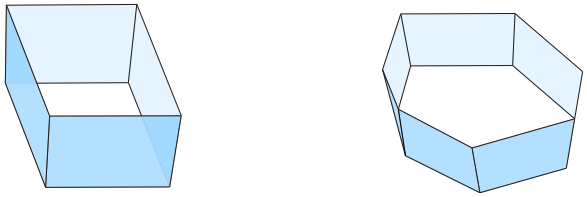
In Section 2.5.7 we show that from (5.45) we can find infinitely many distinct rooms that satisfy (5.16) if the robot's trajectory is linear.

The nullspace of $\overline{\boldsymbol{N}}$ is spanned by $\boldsymbol{e}_1 = \begin{bmatrix} -c & -d & -e & -a & -b & 1 \end{bmatrix}^\top$ so the columns of $\overline{\boldsymbol{U}}$ have to be of the form $[\boldsymbol{u}_n,\ -\boldsymbol{u}'_n]^\top = \lambda \boldsymbol{e}_1$, where $\lambda \in \mathbb{R}$. Equivalently, we have that

$$\begin{bmatrix} \sin\alpha_n \cos\beta_n \\ \sin\alpha_n \sin\beta_n \\ \cos\alpha_n \\ -\sin\alpha'_n \cos\beta'_n \\ -\sin\alpha'_n \sin\beta'_n \end{bmatrix} = \begin{bmatrix} c \\ d \\ e \\ -a \\ -b \end{bmatrix} \cos\alpha'_n.$$

Analogously to (5.35), this system has solutions only for the constant values of $\{\alpha_n, \beta_n\}_{n=1}^N$ and $\{\alpha'_n, \beta'_n\}_{n=1}^N$. As this happens with probability 0 under Assumption 1, we do not have to carry out an analysis for the third receiver.

The results provided in this section prove the following theorem:



a) b)

Figure 5.8: The only room geometries that cannot be uniquely reconstructed from the first-order echoes recorded by two receivers: a) "rooms" with two pairs of parallel walls, b) "rooms" with any number of walls parallel to a line.

Table 5.2: Uniqueness in room reconstruction and robot's localization in 3D for different M. Symbols ✓ and ✗ indicate if the reconstruction is unique or not, respectively.

	$M = 1^*$		$M = 2$		$M = 3$	
	room	loc.	room	loc.	room	loc.
$r = 1$	✓	✗	✓	✗	✓	✗
$r = 2$	✗	✗	✗	✗	✓	✓
$r = 3$	✗	✗	✓	✓	✓	✓
$r = 4$	✗	✗	✓	✓	✓	✓
$r = 5$	✗	✗	✓	✓	✓	✓

* Collocated source and receiver; results derived in Chapter 2.

THEOREM 5.5 Consider a mobile robot with one acoustic source whose waypoints and orientations are picked at random in a 3D room. Three is the minimal number of receivers for which equivalence classes of room-trajectory configurations with respect to first-order echoes recorded by these receivers have measure 0.

It is noteworthy that the third receiver is required to ensure uniqueness in the 3D reconstruction only for the case of $r = 2$. In Section 2.5.2 and Section 2.5.3 we showed that this corresponds to "rooms" with two pairs of parallel walls and "rooms" with any number of walls parallel to a line, illustrated in Fig. 5.8. Any other room in 3D can be uniquely identified by one source and two receivers mounted on a robot, as proved for $r \in \{1, 3, 4, 5\}$. Furthermore, any robot's trajectory can be uniquely deduced by three receivers, unless the robot moves in infinitely long and tall corridors, i.e. $r = 1$, which does not allow unique localization of mobile robots with any number of sources and receivers. All the results on uniqueness in room reconstruction and robot's localization in 3D are summarized in Table 5.2.

5.5 Conclusion

We showed that geometries of convex polyhedral rooms can be uniquely described by the first-order echoes recorded by a few omnidirectional receivers. We established conditions on the number of receivers, room geometry and robot trajectories under which the first-order echoes define a room and trajectory uniquely.

Our characterization is complete in the sense that uniqueness is guaranteed for a robot with two receivers in 2D and three receivers in 3D, while the setups with fewer receivers are proven to suffer from various ambiguities. Throughout the analysis the uniqueness in considered under the assumption that the robot changes its direction arbitrarily. We however identify room-trajectory configurations which are uniquely specified by first-order echoes even without this assumption, as well as those for which the assumption is the essential enabler of uniqueness. Finally, we showed that there exist rooms in which it is impossible to uniquely localize a robot from its first-order echoes with any number of sources and receivers.

These theoretical findings provide a fundamental understanding of the constraints under which one can achieve a unique solution to the range-only simultaneous localization and mapping problem without fixed beacons.

5.A Linear trajectories in 2D

Consider a room with three walls $\{\mathcal{P}_k\}_{k=1}^{K=3}$, where translation and rotation are fixed by $q_1 = 0$, $q_2 = 0$ and $\boldsymbol{n}_1 = [0,1]^\top$. Let the waypoints $\{\boldsymbol{s}_n\}_{n=1}^N$ lie on the wall $k = 1$, $s_{n,y} = 0$ for $n = 1, \dots, N$, while the receiver's locations are given by (5.1). Here we analyze setups with only one receiver, so we can omit the superscript $^{(1)}$ denoting its index, $\boldsymbol{r}_n = \boldsymbol{r}_n^{(1)}$.

Now consider a second room of a different shape $\{\mathcal{P}'_k\}_{k=1}^{K=3}$ with fixed rotation $\boldsymbol{n}'_1 = [0,1]^\top$ and translation $q'_1 = 0$, $s'_{1,x} = 0$. The proof that the two rooms can have the same measurements $\boldsymbol{\Delta}_\mathcal{R} = \boldsymbol{\Delta}_{\mathcal{R}'}$ is based on the observation that every wall k is a tangent to the circles centered at $\{\boldsymbol{s}'_n\}_{n=1}^N$ with radii $\{d'_{nk}\}_{n=1}^N$. Therefore, every wall k is given by

$$\cos\varphi'_k(x - s'_{n,x}) + \sin\varphi'_k(y - s'_{n,y}) = d'_{nk}, \tag{5.46}$$

where d'_{nk} is computed from (5.2),

$$d'_{nk} = 2\ell\cos(\varphi'_k - \alpha'_n) \pm 2\sqrt{\ell^2\sin(\varphi'_k - \alpha'_n)^2 + \Delta_{nk}^2} \tag{5.47}$$

and the fact that $\Delta'_{nk} = \Delta_{nk}$. From (5.46) it also follows that

$$s'_{n,x} = \frac{d'_{nk} - d'_{1k}}{\cos\varphi'_k} - \tan\varphi'_k(s'_{n,y} - s'_{1,y}) \text{ for } n = 2, \dots, N. \tag{5.48}$$

As $s_{n,y} = 0$ implies that $\Delta_{n1}^2 = \ell^2$ for every n, the waypoints $\boldsymbol{s}'_n$ satisfy

$$\begin{aligned}(s'_{n,x} - r'_{n,x})^2 + (s'_{n,y} - r'_{n,y})^2 &= \ell^2\\ (s'_{n,x} - r'_{n,x})^2 + (s'_{n,y} + r'_{n,y})^2 &= \ell^2.\end{aligned}$$

For $s'_{n,y} \neq 0$ we obtain $r'_{n,y} = 0$; thus, all receivers $\{\boldsymbol{r}'_n\}_{n=1}^N$ must lie on the wall $k = 1$ and

$$s'_{n,y} = -\ell\sin\alpha'_n. \tag{5.49}$$

Given $\boldsymbol{\Delta}_\mathcal{R}$, we now construct walls $\mathcal{P}'_2$ and $\mathcal{P}'_3$ different than in the original room $\{\mathcal{P}_k\}_{k=2}^3$, such that $\boldsymbol{\Delta}_{\mathcal{R}'} = \boldsymbol{\Delta}_\mathcal{R}$. Let us start with $k = 2$. We can set the wall arbitrarily by choosing φ'_2 and α'_1 from $[0, 2\pi)$. They determine d'_{12} by (5.47) and $s'_{1,y}$ by (5.49), such that $\Delta'_{12} = \Delta_{12}$. The distance of the wall from the origin is $q'_2 = d'_{12} + s'_{1,y}\sin\varphi'_2$.

For the fixed wall (φ'_2, q'_2), we can find the coordinates of the second waypoint that satisfy $\Delta'_{22} = \Delta_{22}$. Observe that we can choose α'_2 arbitrarily from $[0, 2\pi)$ and compute $s'_{y,2}$ from (5.49). These values determine the distance d'_{22} by (5.47), from which we obtain $s'_{x,2}$ in (5.48). Hence, the waypoints $\boldsymbol{s}'_1$ and $\boldsymbol{s}'_2$ are the centers of the circles whose common tangent is the wall $k = 2$ and the radii d'_{12} and d'_{22} are such that $\Delta'_{12} = \Delta_{12}$ and $\Delta'_{22} = \Delta_{22}$.

By letting $\boldsymbol{s}'_1$ and $\boldsymbol{s}'_2$ be also the centers of the circles with radii d'_{13} and d'_{23}, respectively, we can find their common tangent, which is the wall $k = 3$. We solve the system of two tangent equations from (5.46) with $k = 3, n = 1$ and $k = 3, n = 2$, and have

$$s'_{x,2}\cos\varphi'_3 + (s'_{y,2} - s'_{1,y})\sin\varphi'_3 = d'_{13} - d'_{23}.$$

By replacing d'_{13} and d'_{23} with their definitions in (5.47), we ensure that $\Delta'_{13} = \Delta_{13}$ and $\Delta'_{23} = \Delta_{23}$, and obtain one equation from which we compute φ'_3. As the explicit expression is rather

unwieldy, we do not present it here. The distance of the wall from the origin is $q'_3 = d'_{13} + s'_{1,y} \sin \varphi'_3$.

From the two fixed walls $\mathcal{P}'_2 = (\varphi'_2, q'_2)$ and $\mathcal{P}'_3 = (\varphi'_3, q'_3)$, we can now find the waypoints $\boldsymbol{s}'_n$ for every $n = 3, \ldots, N$ for which $\Delta'_{n2} = \Delta_{n2}$ and $\Delta'_{n3} = \Delta_{n3}$. We do so by solving the system of two equations with two unknowns $s'_{n,x}$ and α'_n:

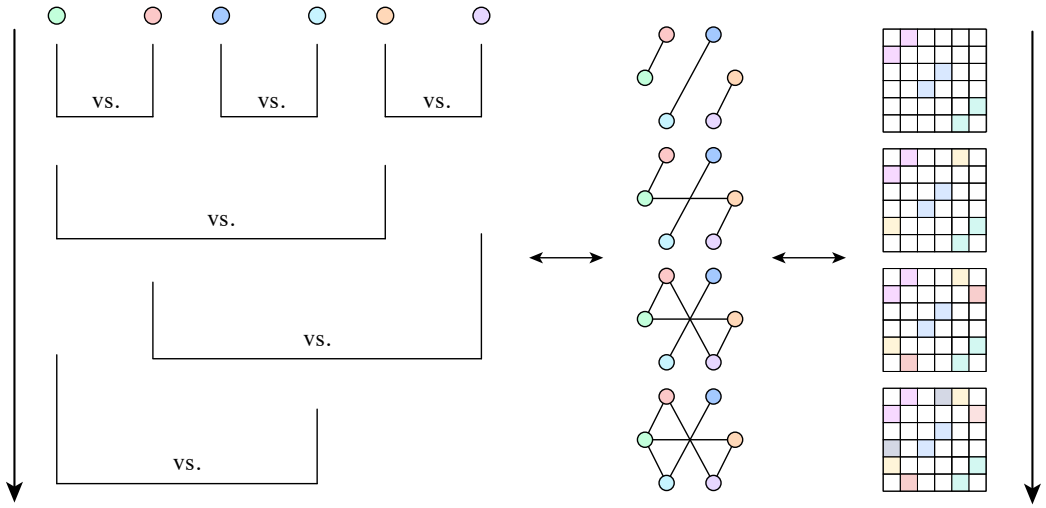
$$\begin{aligned} s'_{n,x} \cos \varphi'_2 - \ell \sin \alpha'_n \sin \varphi'_2 &= d'_{n2} - d'_{12} + s'_{1,y} \sin \varphi'_2 \\ s'_{n,x} \cos \varphi'_3 + \ell \sin \alpha'_n \sin \varphi'_3 &= d'_{n3} - d'_{13} + s'_{1,y} \sin \varphi'_3, \end{aligned}$$

where d'_{n2} and d'_{n3} are defined by (5.47). Finally, from α'_n we get $s'_{n,y}$ by (5.49).

We showed that given $\boldsymbol{\Delta}_{\mathcal{R}}$ measured at collinear waypoints in a room with three walls, we can construct a different room such that $\boldsymbol{\Delta}_{\mathcal{R}'} = \boldsymbol{\Delta}_{\mathcal{R}}$. Remark that, differently from the collocated setup, the trajectory in the equivalent room does not have to be linear.

Chapter 6

Coordinate Difference Matrices: Theory*



Experience arises together with theoretical assumptions not before them, and an experience without theory is just as incomprehensible as is (allegedly) a theory without experience.

Against Method
Paul Karl Feyerabend

*The material in this chapter is the result of joint work of the author (MK) with Gilles Baechler (GB), Frederike Dümbgen (FD), Goolnosh Elhami (GE), Marta Martínez Cámara (MMC) and Martin Vetterli (MV). Author contributions: MK, GB, GE, MMC and MV designed the project. MK, GB, FD and GE proposed CDMs and derived their properties (Section 6.2). MK and GB derived and implemented the point recovery in 1D (Section 6.3), and proposed the vector-form approach (Section 6.4.1). MK derived the generalization to higher dimensions (Section 6.4.2), and performed the overall statistical analysis (Section 6.5). MK, FD and GB put CDMs in perspective with EDMs (Section 6.6). MK created all figures and graphs, as well as the code to generate the graphs. MK wrote the chapter based on [92], which was written by MK, GB, GE and FD.

6.1 Introduction

In Chapter 2 we introduced a point-to-plane distance matrix (PPDM), that is a matrix whose entries are simply distances between the points and planes. We showed that PPDM presents a good model for simultaneous localization and mapping, as its factorization allows us to jointly reconstruct the unknown walls of a room (planes) and the unknown measurement locations (points).

In this chapter, we address a more restricted and easier instance of simultaneous localization and mapping—indoor localization in known environments. This problem happens to be related to a number of other seemingly unrelated ones. For example, imagine that you are given the scores of sport teams participating in a tournament and you are asked to rank them. If you are more inclined towards acoustics, imagine that you need to calibrate the geometry of a microphone array. Even though they appear very different at first glance, these three problems can be solved with a common tool that we describe and analyze in this chapter. The key is to recast these problems into a common framework, in which the goal is to recover a set of points from their pairwise *coordinate differences*. Although we motivate the use of coordinate differences in indoor localization, sport rankings and sensor array calibration, their applications extend to more general ranking problems such as movie and commercial product ratings, as well as other fields such as phase retrieval, calibration in circular tomography devices, and source localization, to name a few.

A vast and mature literature looks at a framework similar to ours, where the task is to retrieve points from their pairwise Euclidean distances. This problem belongs to distance geometry problems (DGPs), where a common scheme is to arrange these distances in a Euclidean distance matrix (EDM) [48]. Several cost functions and optimization methods have been proposed for recovering a point set from its EDM, and in particular the cases of missing and noisy measurements have been treated [42, 61, 71, 137]. There exist thorough surveys on EDMs and distance geometry [99, 115, 129]. The interest in EDMs resides in its wealth of its applications (wireless sensor networks, dimensionality reduction, molecular conformation, localization, robotics), as well as in the beauty of the related mathematical theory. For example, in wireless sensor networks, we can obtain the distance estimation between pairs of nodes from the signal strengths or the times of arrival of the packets sent by other nodes, and rely on those distances to reconstruct the network topology [2, 20, 43]. The molecular conformation problem is another instance of a distance problem [66, 75], where the interatomic distances obtained from nuclear magnetic resonance (NMR) data are used to recover the molecular structures.

In many applications however, more information than mere Euclidean distances between pairs of points is available. For instance, NMR spectroscopy in addition to distances provides the relative orientation of the atoms induced from bond and torsion angles [152]. In Chapter 7, we showcase several other applications in which we can measure length and orientation of the vectors between points, or directly obtain multidimensional difference vectors.

6.1.1 Main contributions

As adapting EDM-based approaches for additional angular information is not straightforward, in this chapter we propose and study a new framework suitable for a class of distance geometry problems extended to vector measurements, known as vector geometry problems (VGPs), which has recently attracted growing interest [17]. To establish and further develop the foundations of VGPs, and inspired by EDMs, we propose to introduce new objects that we call coordinate

difference matrices (CDMs). Similar to their Euclidean distance counterpart, CDMs contain differences between the coordinates of pairs of points.

We put CDMs in perspective with respect to EDMs in terms of properties, characteristics and algorithms. We propose methods to recover points from a variable number of coordinate differences and discuss the uniqueness and optimality of the solutions. We provide conditions on the number and structure of measurements for the recovery to be well-defined, as well as perform statistical analyses of the reconstruction errors.

The main building block for point recovery in any dimension is the CDM constructed from one-dimensional point sets: given D-dimensional difference vectors between D-dimensional points, we can simply create one CDM for each coordinate, and recover the coordinates of the points independently. If we are given the difference vectors of the points projected onto $K \geq D$ frame vectors, we can rely on the same idea to obtain a suboptimal but efficient approach. Interestingly, we show that this method in certain cases also leads to an optimal solution, even in the presence of noise and missing measurements.

To summarize, the main contribution of this chapter is to lay down important mathematical concepts for vector geometry problems by introducing and thoroughly studying CDMs. In the next chapter, we will solve a variety of applications which can be formulated as coordinate difference problems and demonstrate the practical relevance of the CDM theory.

6.1.2 Outline

In Section 6.2 we introduce CDMs and study their properties. We provide a simple 1D point recovery algorithm in Section 6.3, while in Section 6.4 we propose two algorithms for the point recovery in higher dimensions: one which is optimal in the least-squares sense, and one which is computationally more efficient. We also identify cases in which these two algorithms coincide and conduct an error analysis in Section 6.5. We put CDMs in perspective with EDMs in Section 6.6 and compare their properties, required minimum number of measurements, uniqueness of reconstruction, and their usefulness and applicability in problems other than point recovery.

6.2 Coordinate difference matrices

Consider N one-dimensional points $\boldsymbol{x} = [x_1, \dots, x_N]^\top \in \mathbb{R}^N$. We define a coordinate difference matrix $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ with entries $C_{ij} = x_i - x_j$ as the matrix that contains the pairwise coordinate differences of points in $\boldsymbol{x}$. It can be expressed as

$$\boldsymbol{C} = \boldsymbol{x}\mathbf{1}^\top - \mathbf{1}\boldsymbol{x}^\top, \tag{6.1}$$

where $\mathbf{1}$ is the all-one vector.

In Table 6.1, we provide a non-exhaustive list of properties of CDMs that are leveraged later to reconstruct a point set from a given CDM (for derivations, see Appendix 6.A). These ten properties are necessary conditions for a matrix $\boldsymbol{C}$ to be a CDM. It is worth mentioning that the special case of equally spaced points results in the CDM being a Toeplitz matrix. For example, if the points are integers on the interval $[\ell, \ell + N]$, for $\ell \in \mathbb{Z}$, their corresponding CDM is of the

	Properties	Description
P.1	Rank-2	$\text{rank}(\boldsymbol{C}) = 2$ for $N > 1$
P.2	Triangle equality	$C_{ij} = C_{ik} + C_{kj}\ \forall i, j, k$
P.3	Skew-symmetry	$\boldsymbol{C} = -\boldsymbol{C}^\top$
P.4	Hollowness	$\text{diag}(\boldsymbol{C}) = \boldsymbol{0}$
P.5	Zero-sum	$\boldsymbol{1}^\top \boldsymbol{C} \boldsymbol{1} = \boldsymbol{0}$
P.6	Columns as a solution set	CDM of $\boldsymbol{x} = \{C_{ij} \mid \forall i \leq N\}$ is $\boldsymbol{C}$
P.7	Row averaging	$\exists c \in \mathbb{R}$ such that $\boldsymbol{x} + c\boldsymbol{1} = \frac{1}{N}\boldsymbol{C}\boldsymbol{1}$
P.8	Translation invariance	$\boldsymbol{x}$ and $\boldsymbol{x} + c$ have the same CDM $\boldsymbol{C}$
P.9	Imaginary eigenvalues	$\mathfrak{Re}(\lambda_i) = 0$ for $i = \{1, 2\}$
P.10	Anti-symmetric eigenvalues	$\lambda_1 = -\lambda_2$

Table 6.1: Properties of CDMs.

form:

$$\begin{bmatrix} 0 & 1 & 2 & 3 & \dots & N \\ -1 & 0 & 1 & 2 & \dots & N-1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -N & -(N-1) & -(N-2) & -(N-3) & \dots & 0 \end{bmatrix}.$$

Additionally, one of the ten necessary conditions from Table 6.1 is also a sufficient condition:

PROPOSITION 6.1 A matrix $\boldsymbol{C}$ is a CDM if and only if its elements satisfy the triangle equality $C_{ij} = C_{ik} + C_{kj}$ for all triples (i, j, k).

PROOF. The triangle equality of a matrix $\boldsymbol{C}$ 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 triangle equality with the skew-symmetry, we can express all elements as $C_{ij} = C_{ik} - C_{jk}$. We define $\boldsymbol{x} = [C_{1k}, C_{2k}, \dots, C_{Nk}]^\top$, so that we can rewrite C_{ij} as $C_{ij} = x_i - x_j$ for every i and j; therefore, $\boldsymbol{C}$ is of the form (6.1), which concludes our proof. ■

An alternative way to check if the sufficient condition is satisfied is based on the close relation between CDMs and consistent positive reciprocal matrices. For more details, see Appendix 6.B.

6.3 Recovering point embeddings from CDMs

When recovering points from a CDM, we might not know all its entries. To take into account missing entries, we introduce a symmetric weight matrix $\boldsymbol{W}$ with non-negative entries, where

$W_{ij} = 0$ indicates that the entry (i, j) is missing. Moreover, $W_{ij} > 0$ denotes the importance of each difference C_{ij} and it can encompass for example multiple measurements of the same difference or the certainty about each measurement. To simplify the notation, we adopt the convention that $W_{ii} = 0$ for all i.

In addition to be missing, the measured differences can also be noisy; in that regard, we introduce the noise matrix $\boldsymbol{Z}$, whose entries are independent noise realizations. We define an incomplete and noisy CDM as:

$$\widetilde{\boldsymbol{C}} = (\boldsymbol{C} + \boldsymbol{Z}) \circ \boldsymbol{W}. \tag{6.2}$$

In case we have multiple measurements between the points x_i and x_j, the elements $\widetilde{C}_{ij}$ and Z_{ij} are the weighted average of the measurements and the noise realizations, respectively.

The inverse problem that arises naturally from (6.1) and (6.2) is formalized as follows:

PROBLEM 6.2 Given a set of noisy one-dimensional differences $\widetilde{C}_{ij}$ for some i, j, recover the set of points $\{x_i\}_{i=1}^N$ whose pairwise differences best match the measurements.

6.3.1 Reconstruction algorithm

To solve Problem 6.2, we propose to estimate the points from a measured subset of their pairwise differences as:

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}} \left\| \boldsymbol{W} \circ (\boldsymbol{x}\mathbf{1}^\top - \mathbf{1}\boldsymbol{x}^\top - \widetilde{\boldsymbol{C}}) \right\|_F^2.$$

The function $f(\boldsymbol{x})$ is convex and differentiable, so global minima correspond to its stationary points. Setting the gradient to zero, we obtain the linear system $\boldsymbol{A}\boldsymbol{x} = \widetilde{\boldsymbol{v}}$, where $\boldsymbol{A} = \boldsymbol{\Lambda} - \boldsymbol{W}$, $\widetilde{\boldsymbol{v}} = (\widetilde{\boldsymbol{C}} \circ \boldsymbol{W})\mathbf{1}$ and the entries of $\boldsymbol{\Lambda}$ are given by

$$\Lambda_{ij} = \begin{cases} \sum_{k=1}^N W_{ik} & i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{6.3}$$

The matrix $\boldsymbol{A}$ has a particular structure and belongs to the class of so-called *M-matrices* [83, 142]. We study its invertibility in the following section.

6.3.2 Invertibility of $\boldsymbol{A}$

Let us define the weighted graph $G = (\boldsymbol{x}, \boldsymbol{W})$, where vertices are represented by the points $\boldsymbol{x}$ and their connecting edges are given by $\boldsymbol{W}$. In general, graphs provide an interesting alternative representation for CDMs, but in the scope of this chapter we solely leverage them to study the invertibility of $\boldsymbol{A}$.

Observe that $\boldsymbol{A}$ is the Laplacian matrix of its corresponding graph G, as it is the difference of the degree matrix $\boldsymbol{\Lambda}$ and the adjacency matrix $\boldsymbol{W}$. Hence, $\operatorname{rank}(\boldsymbol{A})$ is at most $N - 1$ [72] and it is not invertible.

This result is not surprising; indeed, Property P.8 states that we can recover the original points only up to a translation. Without loss of generality, we arbitrarily fix $x_1 = 0$ to anchor the translation. To that end, we remove the first entry of $\boldsymbol{x}$ and $\widetilde{\boldsymbol{v}}$ and denote the new vectors

by $\boldsymbol{x}'$ and $\widetilde{\boldsymbol{v}}'$. Similarly, we remove the first row and column of $\boldsymbol{A}$ to get $\boldsymbol{A}'$, and we define the matrices $\boldsymbol{\Lambda}'$, $\boldsymbol{W}'$ and $\widetilde{\boldsymbol{C}}'$ analogously. This yields the following linear system:

$$\boldsymbol{A}'\boldsymbol{x}' = \widetilde{\boldsymbol{v}}'. \tag{6.4}$$

6.3.2.1 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 $(\boldsymbol{A}')^{-1}$ exists. Indeed, with $\boldsymbol{\Lambda}' = (N-1)\boldsymbol{I}$ and $\boldsymbol{W}' = \mathbf{1}\mathbf{1}^\top - \boldsymbol{I}$, we obtain

$$(\boldsymbol{A}')^{-1} = \left(\boldsymbol{I} - \frac{\mathbf{1}\mathbf{1}^\top - \boldsymbol{I}}{N-1}\right)^{-1} \frac{1}{N-1} = \frac{\boldsymbol{I} + \mathbf{1}\mathbf{1}^\top}{N}. \tag{6.5}$$

This can be easily verified by computing:

$$\left(\boldsymbol{I} - \frac{\mathbf{1}\mathbf{1}^\top - \boldsymbol{I}}{N-1}\right)\left(\frac{N-1}{N}\left(\boldsymbol{I} + \mathbf{1}\mathbf{1}^\top\right)\right) = \boldsymbol{I}.$$

To recover $\boldsymbol{x}$ from a complete CDM, we substitute (6.5) into (6.4):

$$\widehat{\boldsymbol{x}}' = (\boldsymbol{A}')^{-1}\,\widetilde{\boldsymbol{v}}' = \frac{\boldsymbol{I} + \mathbf{1}\mathbf{1}^\top}{N}\widetilde{\boldsymbol{C}}'\mathbf{1} = \frac{1}{N}\widetilde{\boldsymbol{C}}'\mathbf{1} + c\mathbf{1},$$

where $c = \frac{1}{N}\sum_{ij}\widetilde{C}'_{ij}$ is a constant that only translates the solution; this result confirms Property P.7. We conclude that the optimal point recovery in the complete case corresponds to a simple average of the rows of a CDM.

6.3.2.2 Incomplete and weighted CDM

In the following, we study the invertibility of $\boldsymbol{A}'$ when some entries of the CDM are missing or/and they have different assigned weights.

We say that the CDM $\boldsymbol{C}$ 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 $i_1, i_2, \ldots, i_m$ such that $W_{ii_1} \neq 0, W_{i_1i_2} \neq 0, \ldots, W_{i_{m-1}i_m} \neq 0, W_{i_mj} \neq 0$. According to Kirchhoff's matrix tree theorem [32], the number of spanning trees of G is given by $\kappa(G) = \det \boldsymbol{A}'$. Obviously, G is connected if and only if $\kappa(G) \neq 0$, or equivalently, if and only if $\boldsymbol{A}'$ is nonsingular.

Observe that $\boldsymbol{A}'$ is weakly diagonally dominant, as $\sum_{j=1}^{N} W_{ij} \geq \sum_{j=2}^{N} W_{ij}$ for every row i. When $\boldsymbol{C}$ is connected, then $\boldsymbol{A}'$ 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, $\boldsymbol{A}'$ is irreducibly diagonally dominant. This proves the same result as above, that $\boldsymbol{A}'$ is nonsingular [148], but it provides two additional insights: it shows that we can solve (6.4) in nearly-linear time in N [161], and it enables us to express the inverse of $\boldsymbol{A}'$ by the Neumann series:

THEOREM 6.3 For a connected matrix $\boldsymbol{C}$, the inverse of the corresponding matrix $\boldsymbol{A}'$ is given by the Neumann series

$$(\boldsymbol{A}')^{-1} = \left(\sum_{k=0}^{\infty}((\boldsymbol{\Lambda}')^{-1}\boldsymbol{W}')^k\right)(\boldsymbol{\Lambda}')^{-1}. \tag{6.6}$$

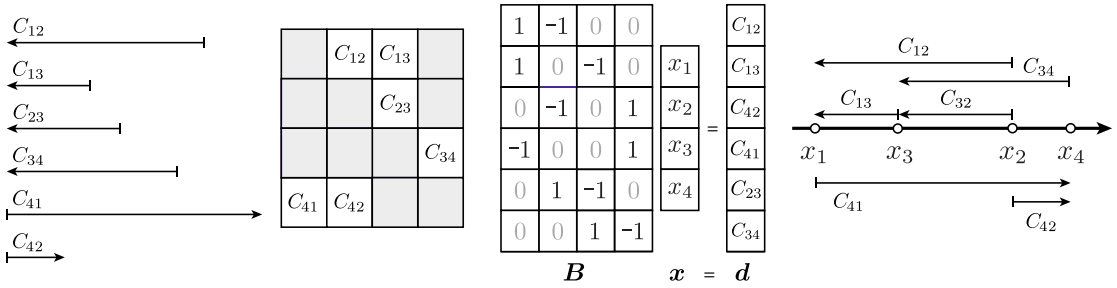


Figure 6.1: An instance of Problem 6.2. a) Assume that we can measure a subset of noisy 1D differences between the points. b) We arrange them in a CDM and use it either directly, or c) its vectorized form to d) reconstruct the points that give rise to measurements.

PROOF. Let $\boldsymbol{\Gamma} = (\boldsymbol{\Lambda}')^{-1}\boldsymbol{W}'$, so that $\boldsymbol{A}' = \boldsymbol{\Lambda}'(\boldsymbol{I} - \boldsymbol{\Gamma})$, and denote the eigenvalues of $\boldsymbol{\Gamma}$ by γ_i. According to Gershgorin circle theorem [64],

$$|\gamma_i| \leq \frac{1}{n_i} \sum_{j \neq i, j \geq 2} W_{ij} = \frac{1}{n_i} \left(\sum_{j \neq i} W_{ij} - W_{i1} \right) = 1 - \frac{1}{n_i} W_{i1} \leq 1.$$

As $\boldsymbol{A}'$ is invertible for connected $\boldsymbol{C}$, $\boldsymbol{\Lambda}'$ and $\boldsymbol{I} - \boldsymbol{\Gamma}$ do not have zero eigenvalues, so $\gamma_i \neq 1$. We conclude that $|\gamma_i| < 1$ and we can invert $\boldsymbol{A}'$ using (6.6). ■

In Appendix 6.C we confirm that the infinite sum (6.6) converges to (6.5) in the case of complete measurements.

The invertibility of $\boldsymbol{A}'$ enables us to reconstruct the points $\boldsymbol{x}$ from (6.4) as

$$\boldsymbol{x} = \begin{bmatrix} 0 \\ (\boldsymbol{A}')^{-1} \widetilde{\boldsymbol{v}}' \end{bmatrix}. \tag{6.7}$$

Clearly, we cannot apply (6.7) when $\boldsymbol{C}$ is not connected. What we can do in such a case is recover the points by invoking (6.7) within each *connected component* of $\boldsymbol{C}$. As the recovered connected components can shift independently, we have an infinite number of solutions.

6.4 Generalization to higher dimensions

In practice, we can often measure the differences between multidimensional points projected onto some measurement directions; examples of such setups are given in Section 7.3 and Section 7.4. This naturally motivates a generalization of CDMs to higher dimensions.

6.4.1 Vector form in 1D

Before discussing the multidimensional case, we introduce an alternative way to represent a CDM by arranging its entries in a vector $\boldsymbol{d} \in \mathbb{R}^M$, where M denotes the total number of measurements. Then, $d_m = x_i - x_j$ are the pairwise differences between the points, and m is indexing the observed

entries of the flattened CDM. To assign the difference d_m to the pair of points x_i and x_j, we introduce a sparse measurement matrix $\boldsymbol{B} \in \mathbb{R}^{M \times N}$ with $B_{mi} = 1$, $B_{mj} = -1$ and 0 otherwise. Using this notation, we can write

$$\boldsymbol{B}\boldsymbol{x} = \boldsymbol{d}. \tag{6.8}$$

In the case of noisy measurements, we define $\widetilde{\boldsymbol{d}} = \boldsymbol{d} + \boldsymbol{z}$, where $\boldsymbol{z}$ contains independent noise realizations. In the case of multiple measurements, $\boldsymbol{B}$ has identical rows for every repeated measurement and $\widetilde{\boldsymbol{d}}$ contains all of their realizations. An instance of Problem 6.2, both in matrix and vectorized form, is illustrated in Fig. 6.1.

Recall that in Section 6.3 we defined $\boldsymbol{W}$ 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 $\boldsymbol{B}$), but also different weights for these measurements. To do so, we can simply scale rows of $\boldsymbol{B}$ and $\widetilde{\boldsymbol{d}}$ with the same factors. To keep the notation clean, in the rest of the chapter we assume that $\boldsymbol{B}$ and $\widetilde{\boldsymbol{d}}$ encompass these weights.

We can estimate $\boldsymbol{x}$ by solving the normal equations, $\boldsymbol{B}^\top \boldsymbol{B}\boldsymbol{x} = \boldsymbol{B}^\top \widetilde{\boldsymbol{d}}$. Due to translation ambiguity, a system is non-invertible. We resolve it by removing the first column of $\boldsymbol{B}$ to get $\boldsymbol{B}'$, which corresponds to setting $x_1 = 0$. This brings us to our well-studied problem (6.4), where $\boldsymbol{A}' = \boldsymbol{B}'^\top \boldsymbol{B}'$, $\widetilde{\boldsymbol{v}}' = \boldsymbol{B}'^\top \widetilde{\boldsymbol{d}}'$ and the points are reconstructed as:

$$\widehat{\boldsymbol{x}}' = (\boldsymbol{B}')^\dagger \widetilde{\boldsymbol{d}} = ((\boldsymbol{B}')^\top \boldsymbol{B}')^{-1} (\boldsymbol{B}')^\top \widetilde{\boldsymbol{d}}. \tag{6.9}$$

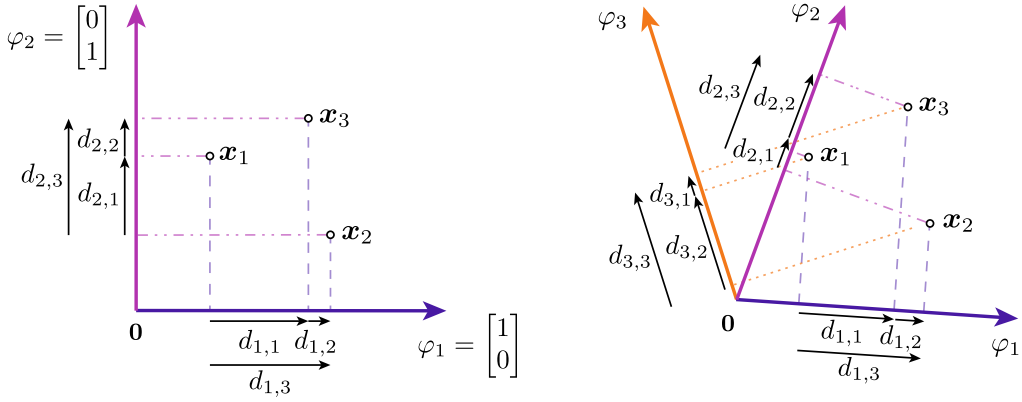
Such a problem definition is not novel; it appears in the broad literature on statistical ranking from pairwise comparisons. For instance, Massey used it to rank a collection of sport teams based on their scores [121], or Osting et al. to design tournaments that maximally improve the informativeness of a ranking for a given number of future comparisons [133]. The reason why we introduce it here is twofold: 1) As we are the first to show the connection of (6.8) with CDMs, the existing problems that rely on (6.8) [121, 133] might find it useful to be recast to the CDM framework and take advantage of the devised properties, bounds on the reconstruction error, more efficient implementations, and connections to other applications. 2) The vector form proves to be beneficial in the error analysis.

6.4.2 Vector form in higher dimension

We extend points and their pairwise differences to D dimensions and consider $K \geq D$ frame vectors $\{\boldsymbol{\varphi}_k\}_{k=1}^K$ arranged in a matrix that is the analysis operator of the frame, $\boldsymbol{\Phi}_0 = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_K]^\top \in \mathbb{R}^{K \times D}$ [174]. Then, we formulate the generalization of Problem 6.2 and expand the formulation introduced in Section 6.4.1:

PROBLEM 6.4 Given a subset of noisy D-dimensional coordinate-wise differences observed in the frame $\{\boldsymbol{\varphi}_k\}_{k=1}^K$, recover the set of points whose pairwise differences best match the measurements.

We assume that we measure M_k differences in each frame direction k, $k = 1, \ldots, K$ and construct the measurement matrix $\boldsymbol{B}_k \in \mathbb{R}^{M_k \times N}$ and the vector of differences $\widetilde{\boldsymbol{d}}_k \in \mathbb{R}^{M_k}$ for every k, analogously to $\boldsymbol{B}$ and $\widetilde{\boldsymbol{d}}$ in (6.8). The total number of differences is denoted by $M = \sum_{k=1}^K M_k$.



a) b)

Figure 6.2: An example of the point recovery in 2D. a) The simple case of the Cartesian coordinate system. We use $d_{k,m}$ for $k = 1, 2$ and $m = 1, 2, 3$, to indicate the measured pairwise differences between the Cartesian coordinates of the points $\{\boldsymbol{x}_n\}_{n=1}^3$. b) A frame with $K = 3$ vectors. We assume to measure $d_{k,m}$ for $k = 1, 2, 3$ and $m = 1, 2, 3$, and we want to recover $\{\boldsymbol{x}_n\}_{n=1}^3$.

We define the operator $\mathcal{T}(\boldsymbol{B}_k, k, K)$ that takes a matrix $\boldsymbol{B}_k$ and generates a matrix in $\mathbb{R}^{M_k \times NK}$ such that we interleave $K - 1$ zero columns between the columns of $\boldsymbol{B}_k$, append $k - 1$ zero columns before the first column of $\boldsymbol{B}_k$ and $K - k$ columns after the last column of $\boldsymbol{B}_k$. For example,

$$\mathcal{T}\left(\begin{bmatrix}1 & -1 & 0\\ 1 & 0 & -1\end{bmatrix}, 2, 3\right) = \begin{bmatrix}0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0\end{bmatrix}.$$

Then, we can formalize Problem 6.4 as follows:

$$\begin{bmatrix}\mathcal{T}(\boldsymbol{B}_1, 1, K)\\ \mathcal{T}(\boldsymbol{B}_2, 2, K)\\ \vdots\\ \mathcal{T}(\boldsymbol{B}_K, K, K)\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}_0 & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Phi}_0 & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\Phi}_0\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \vdots\\ \boldsymbol{x}_N\end{bmatrix} = \begin{bmatrix}\widetilde{\boldsymbol{d}}_1\\ \widetilde{\boldsymbol{d}}_2\\ \vdots\\ \widetilde{\boldsymbol{d}}_K\end{bmatrix}, \tag{6.10}$$

or in matrix form

$$\boldsymbol{B}\boldsymbol{\Phi}\boldsymbol{x} = \widetilde{\boldsymbol{d}}, \tag{6.11}$$

where $\widetilde{\boldsymbol{d}} \in \mathbb{R}^M$ contains all M measured K-dimensional differences in all frames stacked in one vector, $\boldsymbol{\Phi} \in \mathbb{R}^{NK \times ND}$ has the matrices $\boldsymbol{\Phi}_0$ on its diagonal, and $\boldsymbol{x} \in \mathbb{R}^{DN}$ is the vector of all N D-dimensional points $\boldsymbol{x}_n$ that we aim to recover, $n = 1, \ldots, N$.

Fig. 6.2 presents a simple example of three points in a 2D space to illustrate the notation and clarify the above expressions. In Fig. 6.2a, we plot a basic case of $K = D = 2$, where $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ simply contain the differences of the Cartesian coordinates between the points. For example, for

$\boldsymbol{x}_n = [x_{n,1}, x_{n,2}]^\top$, the difference vectors are

$$\boldsymbol{d}_1 = \begin{bmatrix} x_{3,1} - x_{1,1} & x_{2,1} - x_{3,1} & x_{2,1} - x_{1,1} \end{bmatrix}^\top \text{ and } \boldsymbol{d}_2 = \begin{bmatrix} x_{1,2} - x_{2,2} & x_{3,2} - x_{1,2} & x_{3,2} - x_{2,2} \end{bmatrix}^\top.$$

In Fig. 6.2b we consider $K = 3$ frame vectors, $\{\boldsymbol{\varphi}_k\}_{k=1}^K$. For every k, we identify the pairs of points whose pairwise differences are measured, and we construct the measurement matrices $\boldsymbol{B}_k$. To obtain a matrix equation (6.11), which allows us to reconstruct the Cartesian coordinates of points $\{\boldsymbol{x}_n\}_{n=1}^N$ from $\{\boldsymbol{d}_k\}_{k=1}^K$, we need to apply the operator $\mathcal{T}$ on $\boldsymbol{B}_k$ for every $k = 1, 2, 3$. For the given example in Fig. 6.2b and $k = 1$, these matrices are:

$$\boldsymbol{B}_1 = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & -1 \\ -1 & 1 & 0 \end{bmatrix}, \quad \mathcal{T}(\boldsymbol{B}_1, 1, 3) = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

One can then analogously find $\mathcal{T}(\boldsymbol{B}_2, 2, 3)$ and $\mathcal{T}(\boldsymbol{B}_3, 3, 3)$, and by direct multiplication verify that (6.11) is satisfied for the presented noiseless case.

6.4.3 Optimal solution

Analogously to (6.8), the system (6.11) is non-invertible, so we fix the first coordinate of every dimension to zero. This corresponds to removing the first K columns of $\boldsymbol{B}$, as well as the first K rows and D columns of $\boldsymbol{\Phi}$, resulting in the new matrices $\boldsymbol{B'} \in \mathbb{R}^{M \times (N-1)K}$ and $\boldsymbol{\Phi'} \in R^{(N-1)K \times (N-1)D}$, respectively. Then, we can reconstruct the canonical coordinates of the points as

$$\widehat{\boldsymbol{x}}' = (\boldsymbol{B'}\boldsymbol{\Phi'})^\dagger \widetilde{\boldsymbol{d}}. \tag{6.12}$$

This approach is optimal in the least-squares sense.

6.4.4 Splitting algorithm

To reduce the complexity of the solution, we propose and study an alternative approach that decomposes Problem 6.4 into many instances of Problem 6.2 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 $(\boldsymbol{B'})^\dagger \widetilde{\boldsymbol{d}}$, and then we find their canonical coordinates by a simple change of basis:

$$\widehat{\boldsymbol{x}}' = (\boldsymbol{\Phi'})^\dagger (\boldsymbol{B'})^\dagger \widetilde{\boldsymbol{d}}. \tag{6.13}$$

Note that in a more efficient implementation, (6.13) is solved in two steps. The computation of $(\boldsymbol{B'})^\dagger \widetilde{\boldsymbol{d}}$ in the first step is divided into K independent problems, each giving an estimate of $\boldsymbol{f}_k \in \mathbb{R}^K$, $\boldsymbol{f}_k = (\boldsymbol{B}'_k)^\dagger \widetilde{\boldsymbol{d}}_k$. Then, the canonical coordinates of the estimated points $\widehat{\boldsymbol{x}}'$ are in the columns of $(\boldsymbol{\Phi}'_0)^\dagger \boldsymbol{F}$, where $\boldsymbol{F} \in \mathbb{R}^{K \times N}$ contains vectors $\boldsymbol{f}_k$ in its columns. Besides the reduction in the size of the problem, the main benefit of this formulation is that we can take advantage of the structure of $\boldsymbol{B}'_k$ to solve the problem faster: indeed, as mentioned in Section 6.3.2.2, $(\boldsymbol{B}'_k)^\top \boldsymbol{B}'_k$ is irreducibly diagonally dominant, which enables us to solve the linear system in nearly-linear time. The splitting formulation also allows for the method to be easily parallelized.

6.4.5 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 requirement. At the coordinate level, 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 is $D(N-1)$. When using the splitting algorithm, this requirement is more restrictive: we require that the CDM corresponding to every frame vector is connected, and hence, we need at least $(N-1)K$ measurements. We summarize the results on the uniqueness and number of solutions in the following proposition.

PROPOSITION 6.5 The CDM problem in 1D can have either one or 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 and to the best of our knowledge, no trivial solution exists at this point.

6.5 Reconstruction error

In this section, we analytically compute the expected value and variance of the estimation error of the points. For the multidimensional case, we derive the gap in the reconstruction accuracy between the optimal solution and the splitting algorithm. We perform numerical simulations to validate the theoretical analysis and illustrate the dependence of the estimation error on the amount of noise, the number of missing measurements and the number of frame vectors.

6.5.1 1D setup

We can rewrite (6.9) as $\widehat{\boldsymbol{x}}' = \boldsymbol{x}' + (\boldsymbol{B}')^\dagger \boldsymbol{z}$, where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. As $\widehat{\boldsymbol{x}}'$ and $\boldsymbol{x}'$ contain $N-1$ points, we have to prepend the removed leading zero to $\widehat{\boldsymbol{x}}'$ and $\boldsymbol{x}'$, and align them before computing the estimation error. More precisely, to achieve the smallest ℓ^2 error, we set their centroids to the origin by using the centering matrix $\boldsymbol{J} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^\top \in \mathbb{R}^{N\times N}$ and $\boldsymbol{J}' \in \mathbb{R}^{N\times N-1}$ obtained from $\boldsymbol{J}$ by removing its first column. We compute the centered point sets as $\widehat{\boldsymbol{x}}_c = \boldsymbol{J}'\widehat{\boldsymbol{x}}'$ and $\boldsymbol{x}_c = \boldsymbol{J}'\boldsymbol{x}'$. Their difference is the desired estimation error vector, given as $\boldsymbol{e}_c = \boldsymbol{J}'(\widehat{\boldsymbol{x}}'_c - \boldsymbol{x}'_c)$. Thus,

$$\boldsymbol{e}_c \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{e}_c}), \text{ with } \boldsymbol{\Sigma}_{\boldsymbol{e}_c} = \sigma^2 \boldsymbol{J}' \left((\boldsymbol{B}')^\top \boldsymbol{B}'\right)^{-1} (\boldsymbol{J}')^\top.$$

It follows that the reconstructed points $\widehat{\boldsymbol{x}}_c$ are also normally distributed with mean $\boldsymbol{x}_c$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{e}_c}$.

We define the estimation error ϵ as the mean squared error (MSE) between $\widehat{\boldsymbol{x}}_c$ and $\boldsymbol{x}_c$, $\epsilon = \frac{1}{N}\left\|\boldsymbol{e}_c\right\|^2$, and we can find its expected value from:

$$\mathbb{E}\left[\epsilon\right] = \frac{\sigma^2}{N}\operatorname{tr}\left(\boldsymbol{J}''\left((\boldsymbol{B}')^\top \boldsymbol{B}'\right)^{-1}\right), \tag{6.14}$$

where we leverage the cyclic invariance of the trace and use $\boldsymbol{J}''$ to denote $\boldsymbol{J}$ without the first column and row, $\boldsymbol{J}'' = (\boldsymbol{J}')^\top \boldsymbol{J}'$.

This proves that the expected value of the error depends on the noise level σ^2 and the structure of measurements. To better understand the latter, we further rewrite (6.14) as:

$$\begin{aligned}
\mathbb{E}\left[\epsilon\right] &= \frac{\sigma^2}{N}\left[\operatorname{tr}\left((\boldsymbol{A}')^{-1}\right) - \frac{1}{N}\mathbf{1}^\top(\boldsymbol{A}')^{-1}\mathbf{1}\right] \\
&= \frac{\sigma^2}{N}\sum_{k=0}^{\infty}\left[\operatorname{tr}\left(\left(\boldsymbol{W}'(\boldsymbol{\Lambda}')^{-1}\right)^k(\boldsymbol{\Lambda}')^{-1}\right) - \frac{1}{N}\mathbf{1}^\top\left(\boldsymbol{W}'(\boldsymbol{\Lambda}')^{-1}\right)^k(\boldsymbol{\Lambda}')^{-1}\mathbf{1}\right] \\
&\stackrel{\text{def}}{=} \frac{\sigma^2}{N}\sum_{k=0}^{\infty} f_k(\boldsymbol{W}', \boldsymbol{\Lambda}'),
\end{aligned}$$

where $(\boldsymbol{A}')^{-1}$ is from (6.6).

To ease the notation, in this section we use Λ'_i instead of Λ'_{ii}. The expression $f_k(\boldsymbol{W}', \boldsymbol{\Lambda}')$ simplifies for $k = 0$ and $k = 1$ to $\frac{N-1}{N}\sum_{i=1}^{N-1}{\Lambda'_i}^{-1}$ and $-\frac{1}{N}\sum_{i,j=1}^{N-1} W'_{ij}{\Lambda'_i}^{-2}$, respectively. For $k \geq 2$, we have

$$f_k(\boldsymbol{W}', \boldsymbol{\Lambda}') = \sum_{i,j=1}^{N-1}\frac{1}{\Lambda'_i}\sum_{m_1,\ldots,m_{k-1}}\frac{\left(\prod_{\ell=0}^{k-2} W'_{m_\ell,m_{\ell+1}}\right)\left(NW'_{i,m_{k-1}} - (N-1)W'_{m_{k-1},j}\right)}{\prod_{\ell=0}^{k-1}\Lambda'_{m_\ell}}, \tag{6.15}$$

where the second sum is over all $k-1$-tuples $(m_1, \ldots, m_{k-1})$ with $1 \leq m_\ell \leq N-1$ for $\ell = 1, \ldots, k-1$, and $m_0 = i$.

In what follows, we use (6.15) to prove that the smallest error is achieved when the measurements are equally distributed among pairs of points. Let us consider any $\boldsymbol{W}^a$ with the corresponding $\boldsymbol{\Lambda}^a$ defined by (6.3). From the first row, we take two weights W^a_{1j} and W^a_{1k} such that $W^a_{1j} > W^a_{1k}$, and we choose Δ such that $0 < \Delta \leq (W^a_{1j} - W^a_{1k})/2$.

Additionally, let us define the matrix $\boldsymbol{W}^b$ such that it is equal to $\boldsymbol{W}^a$ except for $W^b_{1j} = W^b_{j1} = W^a_{1j} - \Delta$ and $W^b_{1k} = W^b_{k1} = W^a_{1k} + \Delta$. Using (6.3), we also associate the matrix $\boldsymbol{\Lambda}^b$ to $\boldsymbol{W}^b$. Furthermore, let us construct $\boldsymbol{W}^{a\prime}$ and $\boldsymbol{W}^{b\prime}$ from $\boldsymbol{W}^a$ and $\boldsymbol{W}^b$ by removing their first column and row.

Remark that $\boldsymbol{W}^b$ brings the measurements closer to being uniformly distributed. Our goal is to show that $\boldsymbol{W}^b$ leads to a lower estimation error than $\boldsymbol{W}^a$. To see this, observe that, on the one hand, $\boldsymbol{W}^{a\prime} = \boldsymbol{W}^{b\prime}$, but on the other hand $\Lambda^b_j = \Lambda^a_j - \Delta$, $\Lambda^b_k = \Lambda^a_k + \Delta$ and $\Lambda^b_i = \Lambda^a_i$ for $i \neq j$, $i \neq k$. As a consequence, since the numerators in (6.15) depend only on the weights, they are identical for both $\boldsymbol{W}^{a\prime}$ and $\boldsymbol{W}^{b\prime}$, and only their denominators differ. From (6.15) we can therefore compute the difference of $f_k(\boldsymbol{W}^{a\prime}, \boldsymbol{\Lambda}^{a\prime})$ and $f_k(\boldsymbol{W}^{b\prime}, \boldsymbol{\Lambda}^{b\prime})$ as

$$\begin{aligned}
&f_k(\boldsymbol{W}^{a\prime}, \boldsymbol{\Lambda}^{a\prime}) - f_k(\boldsymbol{W}^{b\prime}, \boldsymbol{\Lambda}^{b\prime}) \\
&\qquad = \sum_{i,j=1}^{N-1}\sum_{m_1,\ldots,m_{k-1}} h(\boldsymbol{W}')\left(\frac{1}{\Lambda^{a\prime}_i\prod_{\ell=0}^{k-1}\Lambda^{a}_{m_\ell}{}'} - \frac{1}{\Lambda^{b\prime}_i\prod_{\ell=0}^{k-1}\Lambda^{b}_{m_\ell}{}'}\right),
\end{aligned} \tag{6.16}$$

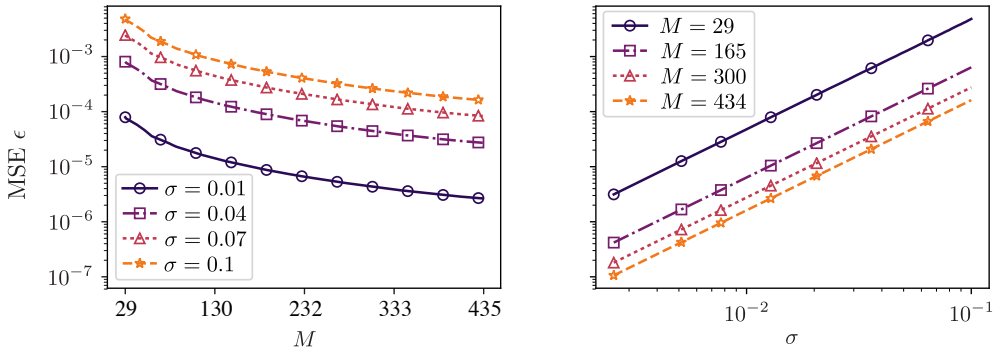


Figure 6.3: CDM estimation error ϵ. We consider $N = 30$ points drawn independently and uniformly from $[0, 1]$ and assume Gaussian noise on the differences with 0-mean and a variance of σ^2. a) Dependence of ϵ on number of measurements M for fixed σ. b) Dependence of ϵ on σ for fixed M.

where $h(\cdot)$ is used to shorten the expressions of the numerators in (6.15).

As $0 \leq \Lambda_j^b - \Lambda_k^b < \Lambda_j^a - \Lambda_k^a$ and $\sum_{i=1}^N \Lambda_i^a = \sum_{i=1}^N \Lambda_i^b$, it is not hard to show that (6.16) is positive for every k. In other words, we have shown that if we rearrange the entries of the first row in $\boldsymbol{W}$ such that they are more uniform and that the variance of $\boldsymbol{\Lambda}$ is smaller, then the estimation error decreases. With a derivation analogous to the above, we can show that the contrary is also true; by rearranging the entries of the first row of $\boldsymbol{W}$ such that the variance in $\boldsymbol{W}$ and $\boldsymbol{\Lambda}$ increases, then the estimation error also increases. As we can arbitrarily swap rows in $\boldsymbol{W}$, as long as we swap their corresponding columns, the entries in the first row of $\boldsymbol{W}$ can be associated to any point $i = 1, \ldots, N$. Therefore, the above proof is general and valid for any i. In fact, we can show that by maximizing expression (6.16) over the entries in $\boldsymbol{\Lambda}^{b\prime}$, the global maximum is achieved when $\Lambda_1^{b\prime} = \ldots = \Lambda_{N-1}^{b\prime}$; these values are equal to the sum of all entries in $\boldsymbol{W}^a$ divided by $N - 1$.

Lastly, we provide expressions for the best and the worst arrangements of measurements with the total sum of the weights fixed to $L(N^2 - N)$, $L \in \mathbb{N}$. As shown above, the smallest error is achieved when $W_{ij} = L$ for every (i, j), $i \neq j$,

$$\epsilon_{min} = \frac{\sigma^2}{N} \sum_{k=0}^{\infty} \frac{N \operatorname{tr}(\boldsymbol{W}^k) - \mathbf{1}^\top \boldsymbol{W}^k \mathbf{1}}{N(L(N-1))^{k+1}} = \frac{\sigma^2}{N} \frac{L(N-1)}{L(N-1)+1}.$$

On the other extreme, the largest error occurs when there exists p for which $W_{pj} = W_{jp} = LN/2$ for every j, $j \neq p$, while $W_{ij} = 0$ for every other entry (i, j), $i \neq p$,

$$\epsilon_{max} = \frac{\sigma^2}{N} \operatorname{tr}((\boldsymbol{\Lambda}')^{-1}) - \frac{1}{N} \mathbf{1}^\top (\boldsymbol{\Lambda}')^{-1} \mathbf{1} = \sigma^2 \frac{2(N-1)^2}{LN^3}.$$

Simulation results. The dependence of the error on the noise level and number of measurements is illustrated in Fig. 6.3. The number of measured differences spreads from the minimum required for reconstruction, $M = N - 1$, to the complete case, $M = N(N - 1)/2$. We compute the error ϵ for multiple realizations of the matrix $\boldsymbol{B}$, i.e. different connectivities between the points, and

take their average to estimate the expected value of ϵ defined in (6.14). It is clear that the error increases with the amount of noise and number of missing entries.

6.5.2 Multidimensional setup

We extend the statistical analysis from Section 6.5.1 to the multidimensional setting and 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 $\widetilde{\boldsymbol{d}} \sim \mathcal{N}(\boldsymbol{d}, 2\sigma^2\boldsymbol{I})$. Analogous to the 1D in Section 6.5.1, we first estimate the partial point sets $\widehat{\boldsymbol{x}}'_o$ and $\widehat{\boldsymbol{x}}'_s$ from (6.12) and (6.13), aligned such that their first points are $\mathbf{0}$. To generalize the centering matrix $\boldsymbol{J}'$ to D dimensions, we apply the operator $\mathcal{T}(\boldsymbol{J}', D, k)$, such that

$$\boldsymbol{J}'_D = \begin{bmatrix} \mathcal{T}(\boldsymbol{J}', 1, D) \\ \mathcal{T}(\boldsymbol{J}', 2, D) \\ \vdots \\ \mathcal{T}(\boldsymbol{J}', D, D) \end{bmatrix} \in \mathbb{R}^{DN \times D(N-1)}.$$

Then, it follows that the centered estimated points, $\widehat{\boldsymbol{x}}_o = \boldsymbol{J}'_D\widehat{\boldsymbol{x}}'_o$ and $\widehat{\boldsymbol{x}}_s = \boldsymbol{J}'_D\widehat{\boldsymbol{x}}'_s$ have Gaussian distributions with following parameters:

$$\begin{aligned} \widehat{\boldsymbol{x}}_o &\sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_o}), \text{ where } \boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_o} = \sigma^2\boldsymbol{J}_D(\boldsymbol{B}\boldsymbol{\Phi})^\dagger((\boldsymbol{B}\boldsymbol{\Phi})^\dagger)^\top(\boldsymbol{J}_D)^\top, \\ \widehat{\boldsymbol{x}}_s &\sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_s}), \text{ where } \boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_s} = \sigma^2\boldsymbol{J}_D\boldsymbol{\Phi}^\dagger\boldsymbol{B}^\dagger(\boldsymbol{B}^\dagger)^\top(\boldsymbol{\Phi}^\dagger)^\top(\boldsymbol{J}_D)^\top. \end{aligned} \tag{6.17}$$

For a less cluttered notation, in (6.17) and the rest of the section, we omit the prime symbol $'$ on $\boldsymbol{J}_D, \boldsymbol{B}, \boldsymbol{\Phi}$.

We define the estimation error vectors of the optimal and splitting algorithms as $\boldsymbol{e}_o = \widehat{\boldsymbol{x}}_o - \boldsymbol{x}$ and $\boldsymbol{e}_s = \widehat{\boldsymbol{x}}_s - \boldsymbol{x}$, respectively. The expectations of the mean-squared errors are $\mathbb{E}\left[\frac{1}{N}\left\|\boldsymbol{e}_o\right\|^2\right] = \frac{1}{N}\operatorname{tr}(\boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_o})$ and $\mathbb{E}\left[\frac{1}{N}\left\|\boldsymbol{e}_s\right\|^2\right] = \frac{1}{N}\operatorname{tr}(\boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_s})$.

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 $\boldsymbol{B}$. Fig. 6.4 shows that the estimation error decreases with the number of frame vectors K and the number of points N.

6.5.3 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, $c = \frac{1}{N}\left\|\widehat{\boldsymbol{x}}_o - \widehat{\boldsymbol{x}}_s\right\|^2$. We can compute the expected cost of splitting

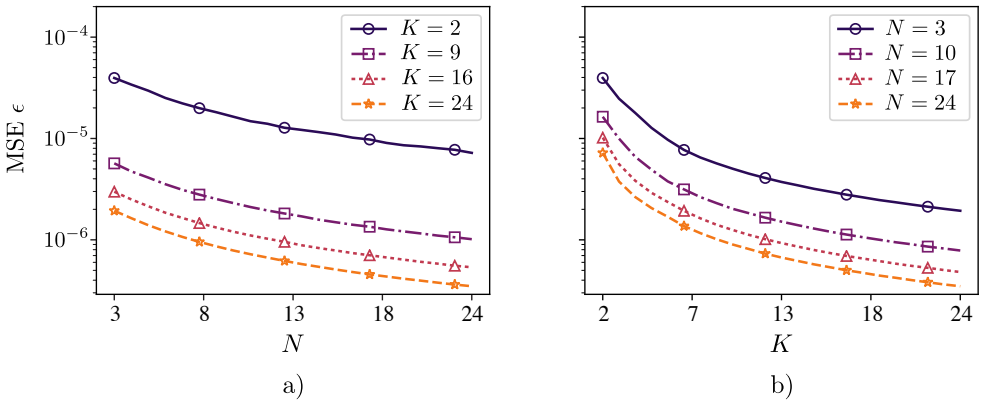


Figure 6.4: Estimation error ϵ in 2D. We assume complete CDM for every frame vector and Gaussian noise with 0-mean and $\sigma = 0.01$ on the measurements. a) Dependence of ϵ on N for fixed K. b) Dependence of ϵ on K for fixed N.

as:

$$\begin{aligned}
\mathbb{E}[c] &= \frac{\sigma^2}{N} \operatorname{tr}\left[\left(\boldsymbol{J}_D \left((\boldsymbol{B}\boldsymbol{\Phi})^\dagger - \boldsymbol{\Phi}^\dagger \boldsymbol{B}^\dagger\right)\right)^\top \left(\boldsymbol{J}_D \left((\boldsymbol{B}\boldsymbol{\Phi})^\dagger - \boldsymbol{\Phi}^\dagger \boldsymbol{B}^\dagger\right)\right) \right] \\
&= \frac{\sigma^2}{N} \left(\operatorname{tr}\left[(\boldsymbol{J}_D)^\top \boldsymbol{J}_D \boldsymbol{\Phi}^\dagger \left(\boldsymbol{B}^\top \boldsymbol{B}\right)^{-1} (\boldsymbol{\Phi}^\dagger)^\top \right] - \operatorname{tr}\left[(\boldsymbol{J}_D)^\top \boldsymbol{J}_D \left((\boldsymbol{B}\boldsymbol{\Phi})^\top (\boldsymbol{B}\boldsymbol{\Phi})\right)^{-1} \right] \right) \\
&= \frac{\sigma^2}{N} \operatorname{tr}(\boldsymbol{\Psi}(\boldsymbol{\Phi}, \boldsymbol{B})) - \frac{\sigma^2}{N^2} \mathbf{1}^\top \boldsymbol{\Psi}(\boldsymbol{\Phi}, \boldsymbol{B}) \mathbf{1} \\
&= \frac{\sigma^2}{N} \left\langle \boldsymbol{\Psi}(\boldsymbol{\Phi}, \boldsymbol{B}), \mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^\top \right\rangle,
\end{aligned} \tag{6.18}$$

where $\boldsymbol{\Psi}(\boldsymbol{\Phi}, \boldsymbol{B}) = \boldsymbol{\Phi}^\dagger (\boldsymbol{B}^\top \boldsymbol{B})^{-1} (\boldsymbol{\Phi}^\dagger)^\top - (\boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{B} \boldsymbol{\Phi})^{-1}$.

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 (6.18) it follows that the cost is equal to 0 if $(\boldsymbol{B}\boldsymbol{\Phi})^\dagger = \boldsymbol{\Phi}^\dagger \boldsymbol{B}^\dagger$, or equivalently, if $\boldsymbol{\Psi}(\boldsymbol{\Phi}, \boldsymbol{B}) = \mathbf{0}$. For $\boldsymbol{B}$ and $\boldsymbol{\Phi}$ defined in (6.10), we prove that there are two practical cases for which $\mathbb{E}[c] = 0$: 1) $K = D$ and 2) $\boldsymbol{B}_k = \boldsymbol{B}_0$ for every $k = 1, \ldots, K$.

6.5.3.1 Case $K = D$

When the number of frame vectors is the same as the dimension of the space, $\boldsymbol{\Phi}$ is invertible, thus $\boldsymbol{\Phi}^\dagger = \boldsymbol{\Phi}^{-1}$. The splitting algorithm results in the optimal solution:

$$\begin{aligned}
\widehat{\boldsymbol{x}}_o &= (\boldsymbol{B}\boldsymbol{\Phi})^\dagger \widetilde{\boldsymbol{d}} \\
&= (\boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{B} \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{B}^\top \widetilde{\boldsymbol{d}} \\
&= \boldsymbol{\Phi}^{-1} (\boldsymbol{B}^\top \boldsymbol{B})^{-1} \boldsymbol{B}^\top \widetilde{\boldsymbol{d}} = \widehat{\boldsymbol{x}}_s.
\end{aligned}$$

6.5.3.2 Case $\boldsymbol{B}_k = \boldsymbol{B}_0$ for every k

Assume that all measurement matrices for every frame direction are equal to the matrix $\boldsymbol{B}_0$. For instance, this is the case when we observe all pairwise differences on all frame vectors k. We

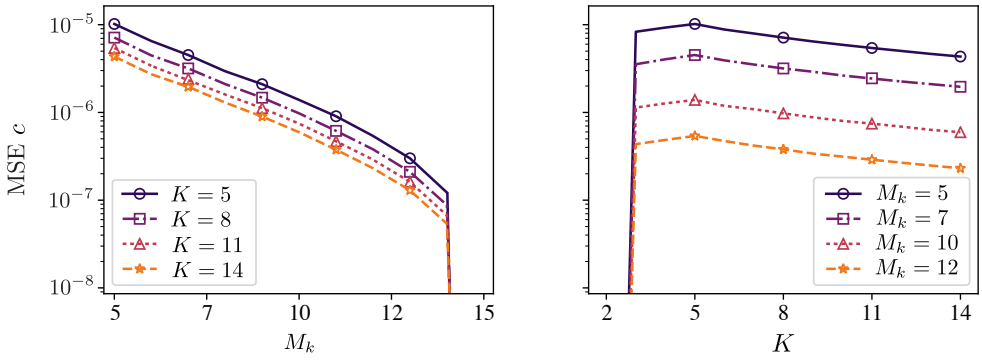


Figure 6.5: The cost of splitting c in 2D. We consider the setup of $N = 6$ points and assume Gaussian noise with 0-mean and $\sigma = 0.01$ on the measurements. a) Dependence of c on M_k for fixed K. b) Dependence of c on K for fixed M_k.

change the order of the entries in $\widetilde{\boldsymbol{d}}$ and the order of the corresponding rows in $\boldsymbol{B}$, so that we can write $\boldsymbol{B} = \boldsymbol{B}_0 \otimes \boldsymbol{I}$, where $\otimes$ is the Kronecker product. Note that this does not influence the estimation of the points. In addition, we also have $\boldsymbol{\Phi} = \boldsymbol{I} \otimes \boldsymbol{\Phi}_0$. Then, it is easy to show that the optimal solution is equal to the solution of the splitting algorithm:

$$\begin{aligned}
\widehat{\boldsymbol{x}}_o &= (\boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{B} \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{d} \\
&= \left((\boldsymbol{I} \otimes \boldsymbol{\Phi}_0^\top) \left((\boldsymbol{B}_0^\top \boldsymbol{B}_0) \otimes \boldsymbol{I} \right) (\boldsymbol{I} \otimes \boldsymbol{\Phi}_0) \right)^{-1} \boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{d} \\
&= \left((\boldsymbol{B}_0^\top \boldsymbol{B}_0)^{-1} \otimes \boldsymbol{\Phi}_0^\dagger \right) \boldsymbol{B}^\top \boldsymbol{d} \\
&= (\boldsymbol{I} \otimes \boldsymbol{\Phi}_0^\dagger) \left((\boldsymbol{B}_0^\top \boldsymbol{B}_0)^{-1} \otimes \boldsymbol{I} \right) \boldsymbol{B}^\top \boldsymbol{d} \\
&= \boldsymbol{\Phi}^\dagger \boldsymbol{B}^\dagger \boldsymbol{d} = \widehat{\boldsymbol{x}}_s.
\end{aligned}$$

Simulation results. To visualize the difference between the two approaches, in Fig. 6.5 we plot the cost of splitting (6.18) for different number of measurements M_k and different number of frame vectors K. We consider $N = 6$ two-dimensional points, so the number of measurements ranges from the smallest value $M_k = N-1 = 5$ to the complete case $M_k = N(N-1)/2 = 15$. To simplify the experiment, we assume that M_k is the same for every direction k, i.e. CDMs have the same number of measured differences. As shown in Section 6.5.3.1 and Section 6.5.3.2, the cost is equal to 0 for $K = D = 2$ and for the complete case; Fig. 6.5 confirms these two special cases. In addition, we observe that the cost decreases with M_k and K.

6.6 Comparison of CDMs with EDMs

In this section, we relate the introduced framework of CDMs with the more mature field of EDMs. The EDM and the CDM theories arise in similar applications, but one can be more useful than the other depending on the measured information. CDMs can be used to solve assigned vector geometry problems, while EDMs play an important role in assigned distance geometry problems, which are harder by nature (the difference vectors are *collapsed* into one dimension). Oftentimes,

Properties	CDM	EDM
Rank	$\operatorname{rank}(\boldsymbol{C}) = 2$	$\operatorname{rank}(\boldsymbol{E}) = D + 2$
Triangle (in)equality	$C_{ij} = C_{ik} + C_{kj}$	$\sqrt{E_{ij}} \leq \sqrt{E_{ik}} + \sqrt{E_{kj}}$
Symmetry	$\boldsymbol{C} = -\boldsymbol{C}^\top$	$\boldsymbol{E} = \boldsymbol{E}^\top$
Hollowness	$\operatorname{diag}(\boldsymbol{C}) = \boldsymbol{0}$	$\operatorname{diag}(\boldsymbol{E}) = \boldsymbol{0}$
Invariance to	translations	all rigid motions

Table 6.2: Comparison of CDMs $\boldsymbol{C}$ and EDMs $\boldsymbol{E}$.

EDM is the tool of choice simply because the coordinate information is not available by design; however, due to its popularity, the EDM framework is sometimes used where CDMs would be more adequate, precise and efficient. 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 [17]. We compare below CDMs and EDMs with respect to: 1) matrix properties, 2) minimal number of observed entries required for the reconstruction, 3) uniqueness of the reconstruction, and 4) applications other than point recovery.

There are some obvious algebraic differences between CDMs and EDMs, summarized in Table 6.2. 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 $D+2$. Furthermore, the entries of CDMs satisfy the triangle equality, while the elements of EDMs respect the triangle inequality. Moreover, CDMs are skew-symmetric, while EDMs are symmetric. Finally, when transitioning from a point set to a CDM, information about the absolute translation of the points is irremediably lost. When estimating points from EDMs, in addition to translation, we also lose the information about their rotation and reflection.

Let us now consider the recovery of points in 1D. Assuming a connected CDM, we require at least one measurement per point, that is $N-1$ coordinate differences to recover N points. To visualize this, we can think in terms of a build-up algorithm and fix the first point at 0. Then we can iteratively reconstruct the remaining points in one pass from the coordinate differences. The entries of EDMs in 1D are simply the squared values of the entries of CDMs, but this small difference makes the recovery problem significantly harder. In addition to fixing the first point at 0, we also need to pick the sign of the second point to fix the reflection. Then we can iteratively build a solution, observing that for every newly added point we have two possibilities. To identify the correct one, we need at least one additional measurement from a previously recovered point.

Moving to point recovery in higher dimensions, the measurements in CDMs are given as K-dimensional coordinate differences between D-dimensional points, where $K \geq D$. The minimal number of measurements is achieved for $N-1$ connected measurements per each dimension, resulting in the total number of $D(N-1)$ 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. We have DN unknowns (N D-dimensional points) to recover, but these points are only recovered up 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 $DN - (D+1)D/2$. Moreover, to ensure that every point is rigidly connected to the others, at least $D+1$ measurements are needed

for every point. As the dimension of the space in most applications is typically 2 or 3, the contribution of the second term is negligible and the reconstruction of points from CDMs and noiseless EDMs requires the same number of measurements. When distances are noisy, there is no clear expression for the minimal number of measurements in the EDM reconstruction problem and no existing algorithm guarantees an optimal solution [48]. On the other side, our proposed algorithm within the CDM framework has an optimal closed-form solution even with imperfect measurements.

This leads to another important difference between CDMs and EDMs. As no algorithm guarantees an optimal solution for EDM reconstruction with incomplete matrices, many methods in the EDM literature split the problem into two independent steps: matrix completion and denoising, followed by point recovery. The goal of the matrix completion and denoising is to determine the closest EDM for a given incomplete matrix. Having the complete EDM, the point recovery is then obtained via a simple SVD. When using CDMs, the completion and denoising steps are not required because one can directly apply the point recovery algorithm on incomplete CDMs, which implicitly denoises and completes CDMs.

Lastly, although EDMs and CDMs are both designed to solve assigned distance and vector problems, they prove to be useful as labeling and denoising 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 [49]. When we measure unlabeled coordinate differences instead of unlabeled distances, CDMs could be used in the same way. Concretely, one could iterate over possible permutations and check if the resulting matrix satisfies Proposition 6.1, the sufficient condition of CDMs. Alternatively, EDMs and CDMs can be used in unassigned geometry problems to denoise a partial solution set during any iterative point recovery algorithm, as shown in Section 7.5.

6.7 Conclusion

The main contribution of this chapter is the formulation and analysis of coordinate difference matrices, simple tools that enabled us to introduce an efficient optimization framework for reconstructing point sets from their noisy and partial coordinate differences. The structure of CDMs also relates to graph theory which in turn helped us provide necessary and sufficient conditions for the proposed framework to work. A significant advantage of CDMs is their easy generalization to the multidimensional setting, where we proposed two methods for the point reconstruction and presented a statistical analysis of their reconstruction errors. The first method is optimal even in the presence of noisy and partial measurements, while the main advantage of the second approach lies in its efficiency. We also found cases where the second method achieves the optimal solution. When compared to algorithms in a similar setup that retrieve points from a set of pairwise Euclidean distances, we showed that our algorithms are advantageous in terms of stability, resistance to noise and the number of required measurements.

In the next chapter we put these theoretical findings to test in four different practical applications in active research fields and show that CDMs can be used to solve a wide range of problems with both simulated and real data.

6.A Proofs of CDM properties

P.1. Following a basic rank inequality, $\operatorname{rank}(\boldsymbol{C}) \leq \operatorname{rank}(\boldsymbol{x}\mathbf{1}^\top) + \operatorname{rank}(\mathbf{1}\boldsymbol{x}^\top) = 2$. Observe that $\operatorname{rank}(\boldsymbol{C}) = 1$ can only happen when $N = 1$.

P.2. $C_{ij} = x_i - x_k + x_k - x_j = C_{ik} + C_{kj}$.

P.3. $C_{ij} = x_i - x_j = -(x_j - x_i) = -C_{ji}$.

P.4. $C_{ii} = x_i - x_i = 0$.

P.5. Follows from Properties P.3 and P.4.

P.6. $\sum_j C_{ij} = \sum_j (x_i - x_j) = Nx_i - \sum_j x_j$. Dividing both sides by N, $\frac{1}{N}\sum_j C_{ij} = x_i + c$, where $c = -\frac{1}{N}\sum_j x_j$.

P.7. Follows from (6.1): every column j of a CDM is equal to $\boldsymbol{x}$ shifted by $-x_j$.

P.8. $(\boldsymbol{x} + c)\mathbf{1}^\top - \mathbf{1}(\boldsymbol{x} + c)^\top = \boldsymbol{x}\mathbf{1}^\top - \mathbf{1}\boldsymbol{x}^\top$.

P.9. Denote λ as an eigenvalue of $\boldsymbol{C}$ and $\boldsymbol{v}$ as a corresponding eigenvector. Then, $\langle \boldsymbol{C}\boldsymbol{v}, \boldsymbol{v}\rangle = \langle \boldsymbol{v}, \boldsymbol{C}^\top \boldsymbol{v}\rangle = -\langle \boldsymbol{v}, \boldsymbol{C}\boldsymbol{v}\rangle$. Observe further that $\langle \boldsymbol{C}\boldsymbol{v}, \boldsymbol{v}\rangle = \langle \lambda\boldsymbol{v}, \boldsymbol{v}\rangle = \lambda \|\boldsymbol{v}\|^2$, and $-\langle \boldsymbol{v}, \boldsymbol{C}\boldsymbol{v}\rangle = -\langle \boldsymbol{v}, \lambda\boldsymbol{v}\rangle = -\lambda^* \|\boldsymbol{v}\|^2$. We conclude that $\lambda = -\lambda^*$, i.e. the eigenvalues of $\boldsymbol{C}$ are imaginary.

P.10. Since $\operatorname{rank}(\boldsymbol{C}) = 2$, we can denote nonzero eigenvalues as λ_1 and λ_2. Knowing that $\operatorname{eig}(\boldsymbol{C}) = \operatorname{eig}(\boldsymbol{C}^\top)$ and $\boldsymbol{C} = -\boldsymbol{C}^\top$, we can write

$$\det \boldsymbol{I} - \lambda\boldsymbol{C} = \det \boldsymbol{I} - \lambda\boldsymbol{C}^\top = \det \boldsymbol{I} - (-\lambda)\boldsymbol{C}.$$

Therefore, if λ is an eigenvalue of $\boldsymbol{C}$, then $-\lambda$ is also an eigenvalue. As there are only 2 nonzero eigenvalues, it follows that $\lambda_1 = -\lambda_2$.

6.B Connection with reciprocal matrices

An alternative way to check if the sufficient condition $C_{ij} = C_{ik} + C_{kj}$ is satisfied for all (i, j, k) is based on consistent positive reciprocal matrices [159], introduced below for the completeness.

DEFINITION 6.6 The matrix $\boldsymbol{R} \in \mathbb{R}^{N\times N}$ is positive reciprocal if $R_{ij} > 0$ and $R_{ij} = R_{ji}^{-1}$, for any $i, j = 1, \ldots, N$. If $R_{ik} = R_{ij}R_{jk}$, it is said to be consistent.

We can transform a CDM $\boldsymbol{C}$ into a reciprocal matrix $\boldsymbol{R}$ with element-wise exponentiation, $\boldsymbol{R} = \exp(\boldsymbol{C})$. The consistency property is then the natural extension of the triangle equality to reciprocal matrices. In combination with the following proposition, this provides an alternative way of testing if a matrix is a CDM.

PROPOSITION 6.7 (SHIRAISHI 1998 [159]) A positive reciprocal matrix $\boldsymbol{R}$ is consistent if and only if $\mathcal{P}_{\boldsymbol{R}}(\gamma) = \gamma^N - N\gamma^{N-1}$, where $\mathcal{P}_{\boldsymbol{R}}(\gamma)$ is the characteristic function of $\boldsymbol{R}$. In other words, $\boldsymbol{R}$ has $N - 1$ zero eigenvalues and one eigenvalue equal to N.

Instead of examining if every triplet (i, j, k) of a given matrix $\boldsymbol{C}$ satisfies the triangle equality, we can verify if the corresponding reciprocal matrix $\boldsymbol{R}$ has $N-1$ zero eigenvalues and one eigenvalue equal to N. If yes, this implies that $\boldsymbol{C}$ satisfies the triangle equality. Following Proposition 6.1, $\boldsymbol{C}$ is a CDM.

6.C Proof of convergence of (6.6) to (6.5) for complete CDMs

Let us define matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ as

$$\begin{aligned}\boldsymbol{P} &= \sum_k \boldsymbol{P}_0^k = \sum_{k=0}^{\infty} \left(\frac{\mathbf{1}\mathbf{1}^\top - \boldsymbol{I}}{N-1}\right)^k, \\ \boldsymbol{Q} &= \frac{N-11}{N}(\mathbf{1}\mathbf{1}^\top + \boldsymbol{I}).\end{aligned}$$

We need to show that $\boldsymbol{P} = \boldsymbol{Q}$. To this end, we establish that the two matrices have the same eigenvalues and corresponding eigenvectors. First, observe that we can rewrite any square matrix in $\mathbb{R}^{(N-1)\times(N-1)}$ that has diagonal values equal to α and non-diagonal values equal to β as

$$\beta\mathbf{1}\mathbf{1}^\top + (\alpha-\beta)\boldsymbol{I} = (\alpha-\beta)\left(\boldsymbol{I} + \frac{\beta}{\beta-\alpha}\mathbf{1}\mathbf{1}^\top\right).$$

From the matrix determinant lemma, we know that

$$\det \beta\mathbf{1}\mathbf{1}^\top + (\alpha-\beta)\boldsymbol{I} = (\alpha-\beta)^{N-2}\left(1 + (N-1)\frac{\beta}{\alpha-\beta}\right). \tag{6.19}$$

Using (6.19), we compute the eigenvalues of $\boldsymbol{P}_0$ by solving its characteristic equation:

$$\det \boldsymbol{P}_0 - \lambda\boldsymbol{I} = \left(\lambda + \frac{1}{N-1}\right)^{N-2}\left(1 - \frac{N-1}{\lambda(N-1)+1}\right) = 0.$$

Thus, the eigenvalues of $\boldsymbol{P}_0$ are $\lambda_0 = \frac{-1}{N-1}$ (with multiplicity $N-2$) and $\lambda_1 = \frac{N-2}{N-1}$. The two eigenvalues of $\boldsymbol{P}$ are consequently equal to

$$\begin{aligned}\kappa_0 &= \sum_{k=0}^{\infty}\left(\frac{-1}{N-1}\right)^k = \frac{N-1}{N}, \\ \kappa_1 &= \sum_{k=0}^{\infty}\left(\frac{N-2}{N-1}\right)^k = N-1.\end{aligned}$$

Furthermore, the eigenvectors of $\boldsymbol{P}_0$ (and by extension of $\boldsymbol{P}$) can be found by solving the eigenvalue equations

$$\begin{aligned}(\boldsymbol{P}_0 - \lambda_0\boldsymbol{I})\boldsymbol{u}_0 &= \mathbf{1}\mathbf{1}^T\boldsymbol{u}_0 = \mathbf{0}. \\ (\boldsymbol{P}_0 - \lambda_1\boldsymbol{I})\boldsymbol{u}_1 &= \left(\mathbf{1}\mathbf{1}^T - (N-1)\boldsymbol{I}\right)\boldsymbol{u}_1 = \mathbf{0}.\end{aligned} \tag{6.20}$$

We use the matrix determinant lemma again to compute the eigenvalues of $\boldsymbol{Q}$:

$$\det \boldsymbol{Q} - \lambda\boldsymbol{I} = \left(\lambda - \frac{N-1}{N}\right)^{N-2}\left(1 + \frac{(N-1)^2}{N-1-\lambda N}\right) = 0.$$

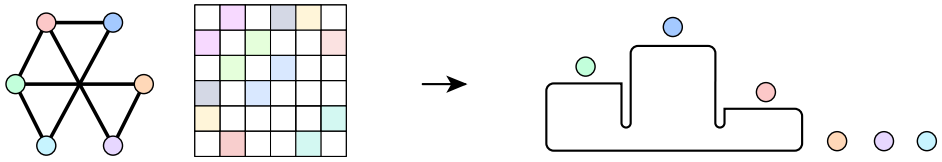
The corresponding eigenvalues are $\frac{N-1}{N}$ (with multiplicity $N-2$) and $N-1$, which are the same as $\boldsymbol{P}$. Moreover, the eigenvectors of $\boldsymbol{Q}$ can be found by solving

$$\begin{aligned}(\boldsymbol{Q}-\kappa_0\boldsymbol{I})\boldsymbol{v}_0 &= \mathbf{1}\mathbf{1}^T\boldsymbol{v}_0 = \mathbf{0}.\\ (\boldsymbol{Q}-\kappa_1\boldsymbol{I})\boldsymbol{v}_1 &= \left(\mathbf{1}\mathbf{1}^T-(N-1)\boldsymbol{I}\right)\boldsymbol{v}_1 = \mathbf{0}.\end{aligned} \tag{6.21}$$

By comparing equations (6.20) and (6.21), we see that matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ also have equal eigenvectors and thus are equal.

Chapter 7

Coordinate Difference Matrices: Applications*



The final test of a theory is its capacity to solve the problems which originated it.

Linear Programming and Extensions
George Dantzig

7.1 Main contributions and outline

Inspired by Euclidean distance matrices (EDMs) and their broad use in solving distance geometry problems (DGPs), in Chapter 6 we introduced coordinate difference matrices (CDMs). In this chapter, we show that our CDM framework is well suited for the vector geometry problems (VGPs), where the dissimilarity between points is not only given by their distance, but also by the orientation of the difference vector between them. One of the main contributions of this chapter is to demonstrate and reveal connections between problems that seem very different at first glance. In particular, we stress the generality and practicality of the theoretical framework

*The material in this chapter is the result of joint work of the author (MK) with Gilles Baechler (GB), Juri Ranieri (JR), Frederike Dümbgen (FD), Goolnosh Elhami (GE) and Martin Vetterli (MV) [91–93]. Section 7.2: MK analyzed the data and wrote the section. Section 7.3: FD worked on real measurements; MK performed numerical simulations and wrote the section based on [92], written by MK, GB, FD and GE. Section 7.4: FD implemented E-MDS and MDS algorithms; MK wrote the section based on [91], written by MK, GB, FD and GE. Section 7.5: JR devised the support recovery algorithm; MK and GB proposed algorithmic improvements, analyzed the complexity and influence of different point configurations, and carried out experiments; MK wrote the section based on [93], written by MK, GB and JR.

established in Chapter 6 via four different applications, ranging from ranking in sports to phase retrieval.

In Section 7.2 we study ranking in sports and abstract the problem by considering the strength of teams as points on a line. With such a formulation, ranking teams simplifies to solving 1D CDM problem. By applying CDMs to rank sports teams, we do not aim to demonstrate the superior performance of our method over state-of-the-art ranking techniques, but to show an application of the CDM framework in 1D and gain more insights about the proposed algorithm using real data.

In Section 7.3 we exploit CDMs for sensor array calibration in 2D, that is determining the locations of sensors given measurements from external calibration sources. We demonstrate the robustness of our algorithm by performing measurements in real environment. Even though the solution to this problem is quite simple, it nicely visualizes the multidimensional setup introduced in the context of CDMs for the case when the number of measurement directions is higher than the dimension of the space.

In Section 7.4 we consider a self-localization of nodes in a sensor network in 2D. We assume that every sensor is equipped with angle-of-arrival technology, and it can measure its distance and angle from a subset of other sensors in the network. Given a set of noisy and incomplete measurements from every sensor, our goal is to recover sensors' positions. We show that such a localization problem can be framed as the multidimensional CDM problem for the case when the measurements are observed in Cartesian coordinates. Moreover, we present results of numerical simulations to confirm that the proposed CDM-based algorithm outperforms the state-of-the-art multimodal localization.

Lastly, in Section 7.5 we demonstrate that CDM framework can be leveraged in the context of unlabeled vector geometry problems (uVGPs) as a denoising tool. The distinction between this and the first three problems is that the labels are not known: in the context of CDMs, it corresponds to having access to the matrix elements but not their positions. A typical example of an uVGP is the (noisy) turnpike problem [41]. It finds applications in many fields, including phase retrieval for sparse signals, which is studied in this chapter. We propose a pipeline of algorithms that solve the phase retrieval for sparse signals defined on continuous domain. In the proposed pipeline, the most challenging step is labeling entries of CDMs. As our overall algorithm results in significant improvement over the state of the art, in this chapter we describe the complete solution to the phase retrieval problem and emphasize parts related to CDMs.

For each application, we present a literature review, make a connection with CDMs, propose the solution and discuss the main results.

7.2 Rankings and ratings in sports

7.2.1 Introduction

In the sports community, ranking teams based on their performances has been a long-standing question with a number of proposed solutions. The problem statement is elegantly simple—arrange teams in order such that the *better* team has a higher rank. We are interested in a more general problem called rating, which assigns to teams absolute scores that reflect their performance.

As the rating problem goes beyond sports, the literature is very rich and connects many different fields such as optimization, statistics, game theory, etc. Langville and Meyer explain

fundamental ideas behind mathematical rating systems and summarize about a dozen of the best known ranking techniques [106]. The most important ones differ in the considered model of the teams and the amount of used information. As the CDM framework revolves around pairwise differences, we are interested in pairwise comparison methods, which are a subset of ranking methods of the most widespread interest. We mention here only a few of the most fundamental approaches: Colley's matrix model [36] relies solely on the number of wins and loses and formulates the ranking problem as a linear set of equations. To incorporate more information, Keener [84] proposes to arrange the absolute scores of each team in a non-negative matrix and uses the eigenvalues decomposition to rank the teams. Jech [82] takes a probabilistic formulation and describes conditions for a unique ranking in incomplete tournaments. The theory of Markov chains is also leveraged to produce rankings [33, 68]. More recent algorithms take into account advanced statistics [103, 177, 182].

The evaluation and the comparison of different approaches can be troublesome in the absence of one absolute *ground truth ranking* [106]. We therefore solely aim to show that CDMs can be leveraged in the ranking application using real data.

7.2.2 Problem setup

One of the most widely used rating algorithms in the sports community is called Massey's method [121]. It abstracts the problem by considering the strength of teams as points on a line. We undertake the same approach and assume that the strength of a team encompasses all its characteristics, including the quality of the players, points scored in the past games, statistics, etc. The greater the strength, the higher the likelihood that a team is going to win a game. Then, we model the *net score* of a game between team i and team j as $x_i - x_j + z_{ij}$, where x_i represents the strength of the team i and z_{ij} are independent noise realizations. This formulation enables us to directly use the net scores as CDM entries. Moreover, the entries of the weight matrix $\boldsymbol{W}$ are $W_{ij} = 1$ if teams i and j played against each other and 0 otherwise. If the teams i and j play more than one game with each other, C_{ij} is simply a weighted average of the score difference and W_{ij} represents the sum of these weights. This also makes it possible to assign different weights to games. For instance, the first game of the championship is probably less important than the final of the playoffs, and hence could have a lower weight assigned. The strengths of the teams $\{x_i\}_{i=1}^N$ are then recovered from (6.7).

7.2.3 Discussion

We illustrate an application of our algorithm to the results of a regular NBA season, which ranks teams based on their winning percentage. We construct the CDM $\boldsymbol{C}$ and the weight matrix $\boldsymbol{W}$ from all the team scores of the NBA regular season 2015/16. We assign the same weight to all games and estimate the ranking of the teams by (6.7). Table 7.1 lists the first 9 teams ranked based on our proposed algorithm. We observe that the same teams are among the first 9 in the actual NBA ranking, but in a slightly different order. The two most outstanding cases are the Oklahoma City Thunder and the Toronto Raptors. The Thunder is placed 3^{rd} in our ranking and 5^{th} in the NBA standings, while the Raptors are placed 6^{th} in our ranking and 4^{th} in the NBA standings.

To find clues for this disparity, we look at the percentage of games that were won or lost by a small difference. More precisely, for each team we calculate the ratio of the number of games that were won (lost) by a score difference less than 10 (tight win/loss) over the total number of

Team	CDM	NBA	W	L	TW/W	TL/L	NETRG
Golden State Warriors	1	1	73	9	0.425	0.444	11.6
San Antonio Spurs	2	2	67	15	0.418	0.600	11.8
Oklahoma City Thunder	3	5	55	27	0.418	**0.852**	6.9
Cleveland Cavaliers	4	3	57	25	0.491	0.680	5.8
Los Angeles Clippers	5	6	53	29	0.547	0.586	5.5
Toronto Raptors	6	4	56	26	**0.589**	0.731	4.3
Atlanta Hawks	7	7	48	34	0.417	0.588	4.1
Boston Celtics	8	8	48	34	0.500	0.676	3.0
Charlotte Hornets	9	9	48	34	0.521	0.588	3.3

Table 7.1: Comparison of rankings with CDMs, traditional rankings (NBA), and the NETRG advanced statistics. Here, W is the total number of wins, L is the total number of losses, TW is the number of tight wins (+10 or less), and TL is the number of tight losses (-10 or less).

games won (lost) by a team. We display these values in columns TW/W and TL/L in Table 7.1. Observe that the highest value of the column TW/W belongs to the Raptors, who won 58.9% of their games by a small margin. Clearly, our algorithm takes that into account and places this team two spots lower than the NBA ranking. Similarly, we observe that the largest value of the column TL/L belongs to the Thunder: 85.2% of their losses were tight, which explains why our algorithm ranks the team two spots higher than the NBA standings.

We also underline the correlation between our proposed ranking and the most widely used advanced statistics called *net rating* (NETRG). It is defined as the difference between points scored and allowed per 100 possessions, and it reflects how efficient a team is when it possesses the ball, which is believed to be strongly correlated with the team's strength. We observe that the correlation between the ranking obtained by our algorithm and the advanced statistics (0.983) is higher than the correlation between the NBA standings and the NETRG (0.9). In fact, our proposed ranking is identical to the NETRG, except for the first and last two teams that are switched. This is remarkable given that we just use the net scores to rank the teams.

We have therefore demonstrated that the CDM framework developed in the previous chapter can be useful in estimating ranking and ratings of sports teams by using the NBA data set. However, we stress that our goal is not to revolutionize the way teams are officially ranked; after all, that would completely change the behavior of the teams, who would switch their objectives from *winning games* to *winning games with the largest margin*.

7.3 Sensor array calibration

7.3.1 Introduction

In our second application, we exploit CDMs for sensor array calibration, that is determining the locations of sensors given measurements from external calibration sources. We distinguish this from sensor *self-localization*, which is treated in Section 7.4.

Sensor arrays have been employed in real-time monitoring and measurement for decades, and the importance of accurate calibration of their relative positions is evident in numerous applications. For example, in signal processing, sensor arrays are often employed to measure physical phenomena. This includes wireless sensor networks measuring weather conditions [78], ultrasonic sensors detecting breast cancer in ultrasound tomography [54], and room geometry estimation from a microphone array [49]. Furthermore, in the field of acoustics, the most accurate solutions to common problems such as direction-of-arrival estimation [118, 135, 153], source separation [76, 171] and noise reduction [62] rely on microphone arrays with precisely known microphone locations, [14]. Therefore, accurate localization of all sensors in the network is critical for many tasks.

We demonstrate the application of CDMs for the accurate position calibration of sensor arrays. We consider measurements from a number of sources placed in far field at known locations, emitting waves that can be approximated to arrive with the same incident angles at all sensors. Even though the described setup is of practical relevance, it has not received much attention in the literature. Many proposed methods consider the sources in the near field and assume their exact synchronization [18, 20, 38, 63, 102]. These approaches use measured distances between sources and sensors, and are closely related to the framework of EDMs. Extensions to far-field calibration also exist [60, 90, 131, 167]; however, in these works the source locations are assumed to be unknown and the algorithms iteratively estimate the direction of arrival along with the sensor locations, which is not necessary in the method we propose. We show that knowledge of the source locations in far field can be used to directly apply a CDM-based algorithm for point recovery in higher dimensions.

7.3.2 Problem setup

We aim to localize N sensors from the measurements of K calibration sources placed at angles $\{\boldsymbol{\varphi}_k\}_{k=1}^K$ in the far field. The far-field assumption implies that the sources emit plane waves, and the incident angle $\boldsymbol{\varphi}_k$ of some fixed source k is the same for all sensors. The measurements from the calibration sources are conducted in the following way: The sources produce waves at unknown times, and the sensors register the absolute times of arrival (TOA) of the waves, denoted τ_{nk} for the nth sensor and the kth source. If there is a sensor n that registered all TOAs for every source k, then we can subtract τ_{nk} from the detected times τ_{mk} of all the sensors $m = 1, \dots, N$, for every direction k. This corresponds to fixing the nth sensor to the origin. However, if this is not the case, we cannot combine measurements from different sources k, as each of them has a different emission time. Nevertheless, we can compute the pairwise differences of the registered times, which brings us to the framework of CDMs. Such measurements can be seen as coordinate differences of the sensors, projected onto the directions of the frame vectors $\boldsymbol{\varphi}_k$. This is illustrated in Fig. 7.1a. In accordance with the theory developed in Section 6.4 of the previous chapter, 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 to satisfy the connectivity requirement (see Section 6.4.5).

To investigate the performance of the proposed method, we used the experimental setup in Fig. 7.1b. The setup included a microphone array called *Pyramic* [150] and three speakers placed in an anechoic chamber. As the speakers were located as far as possible to emulate far-field conditions, it was more practical to turn the microphone array instead of turning the speakers, yielding the same desired relative orientation.

The Pyramic array and the distribution of microphones in each branch are more closely dis-

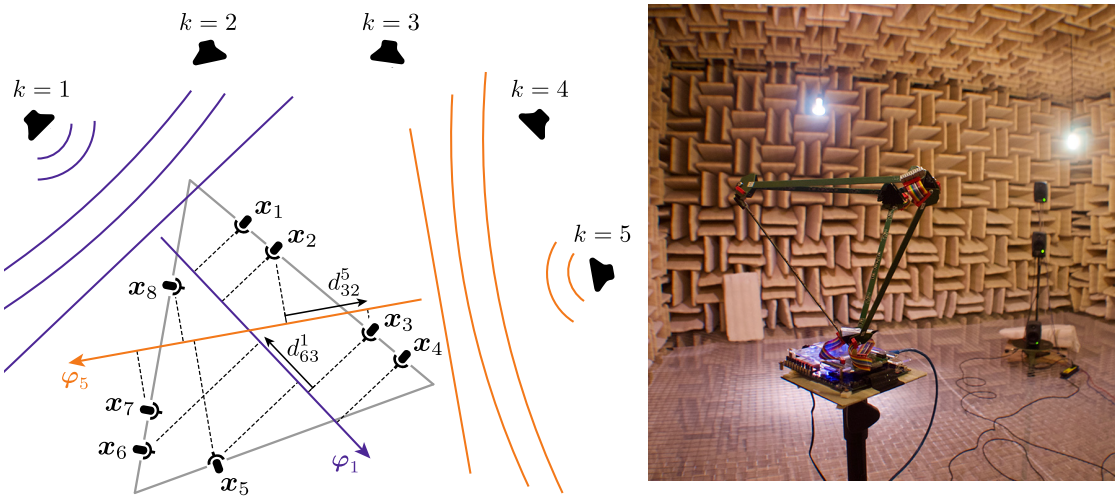


Figure 7.1: Calibration of sensor arrays: geometry and real experiment. a) Three edges of the Pyramic array with several microphones. The differences of the measured times of arrival for the calibration source k and two microphones i, j are denoted d^k_{ij}. b) Pyramic array (foreground) and speakers (background) in an anechoic chamber.

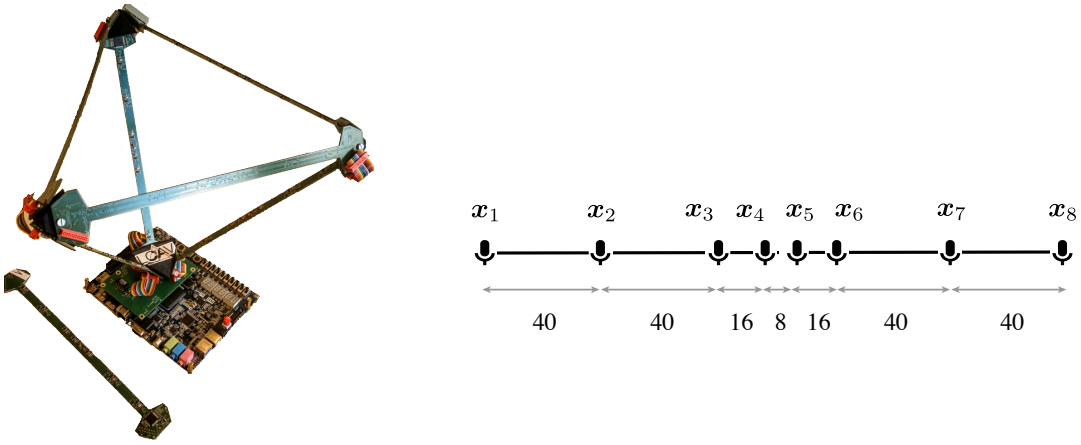
played in Fig. 7.2. The array was designed such that distances between microphones ensure high frequency resolution and avoid spatial aliasing. The shortest distance between two microphones is 8 mm, which implies that the maximum frequency of the recording without spatial ambiguity is $f_{max} = 42.5$ KHz. The maximum distance is 200 mm, corresponding to f_{max} of 1.7 KHz. Therefore, with such a microphone placement, the system is able to span a wide range of frequencies, while maintaining a reasonable size of the microphone array [7].

7.3.3 Results

To test our algorithm in a real environment, we localize the 21 microphones of the top triangle of the Pyramic array using 90 calibration directions uniformly spaced in $[0, 180°)$. In the experiment, the speaker is in the same plane as the top triangle of the Pyramic array, so we consider a two-dimensional setup, $D = 2$. The vectors in the frame $\boldsymbol{\Phi}_0$ correspond to the directions of arrival of the sound played by a speaker:

$$\boldsymbol{\Phi}_0 = \begin{bmatrix} \cos 0° & \cos 2° & \cos 4° & \dots & \cos 178° \\ \sin 0° & \sin 2° & \sin 4° & \dots & \sin 178° \end{bmatrix}^\top.$$

All microphones worked properly throughout the experiment, so for every $k = 1, \dots, 90$ we could compute the pairwise differences of the sound detection times for all pairs of 21 microphones,



a) b)

Figure 7.2: a) A picture of the 3D Pyramic array. Pyramic has 48 microphones spread on 6 printed circuit boards, each bearing 8 MEMS (microelectromechanical systems) microphones and an analog-to-digital converter. b) Eight microphones placed in one board of the Pyramic array, where distances are in mm.

arrange them in a measurement vector $\widetilde{\boldsymbol{d}}$ and create a corresponding matrix $\boldsymbol{B}_k$ as

$$\boldsymbol{B}_k = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 1 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \in \mathbb{R}^{210\times 21}.$$

We used the splitting algorithm (6.13) to find the locations of the microphones. As $\boldsymbol{B}_k = \boldsymbol{B}_0$ for every k, it leads to the optimal solution, but more efficiently than solving the original problem by (6.12). Our method localizes the microphones with a mean squared error (MSE) of 4.45 µm.

In addition to this real experiment, we performed numerical simulations with $K = 90$ and $N = 21$ to evaluate the impact on the number of active microphones on the robustness of the reconstruction. We started with a complete set of pairwise differences $M = \sum_{k=1}^{K} M_k = 90\binom{21}{2}$, as in the real experiment, and decreased the value of M until we had a sparsely connected graph for every direction k. The results are illustrated in Fig. 7.3. These simulations confirm that the MSE decreases at a faster rate when adding a measurement in a relatively incomplete setup (see the left side of the graph). On the other hand, adding a measurement to an almost complete setup has little benefit, as intuitively expected.

7.4 Multimodal sensor localization

7.4.1 Introduction

We consider a sensor network in which nodes can measure distances and angles between each other, and we want to recover the sensors' locations given a set of such measurements. This

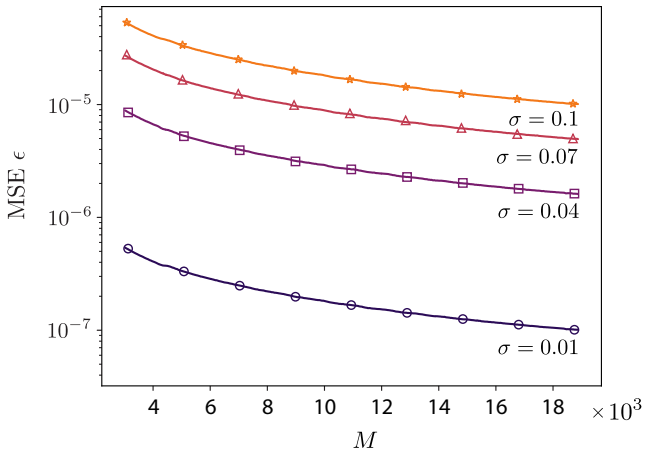


Figure 7.3: CDM estimation error ϵ. We consider $N = 21$ microphones, $K = 90$ frame vectors uniformly spread by 2°, and Gaussian noise on the differences with 0-mean and a variance of σ^2. We illustrate the dependence of ϵ on number of measurements M for different values of σ. The number of measured differences varies from $M = 3000$ to the complete case, $M = 18900$.

problem is central to many applications, including indoor localization, autonomous vehicles, or intelligent warehouses.

When the nodes can measure only their pairwise distances, the problem is well studied and can be solved with a number of algorithms relying on the EDM theory [38, 46, 63, 132]. Similarly, when the only available information are the angular measurements between the nodes, several studies have been proposed to localize the sensors [29, 105, 141]. Setups leveraging both measurement modalities did not attract as much attention. This is surprising given the fact that a multimodal approach could provide a significant improvement in accuracy and robustness. On the other hand, distance and angle measurements are often of a fundamentally different nature, so it is not obvious how to combine them into a precise mathematical framework; the consistent combination of quantities of different unities into one framework is studied by Hart [73].

For the particular case of angles and distances, trigonometric properties linking the two can be exploited [19], where the cosine law is applied to the triangles formed by triplets made of the points to be localized. Macagnano et al. [117] improve on this method by introducing edge-multidimensional scaling (E-MDS), which solves a multidimensional scaling (MDS) problem adapted for the vectors between points. In our previous work [91], we build upon the work of Macagnano et al. [117] and enhance their E-MDS method by introducing additional constraints that enforce geometric consistency between the edge vectors.

Here we demonstrate how to combine distance and angle measurements into CDMs, which allows us to localize sensors in closed form. We compare our algorithm with the state-of-the-art solution [117] and demonstrate its superior performance.

7.4.2 Problem setup

Consider a set of N points in 2D with coordinates denoted by $\boldsymbol{x}_i \in \mathbb{R}^2$ for $i = 1, \dots, N$. The vector of coordinate differences $\boldsymbol{d}_{ij} \in \mathbb{R}^2$ between the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is given by $\boldsymbol{d}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$. Its length is the Euclidean distance between the points, $\ell_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$, while its orientation

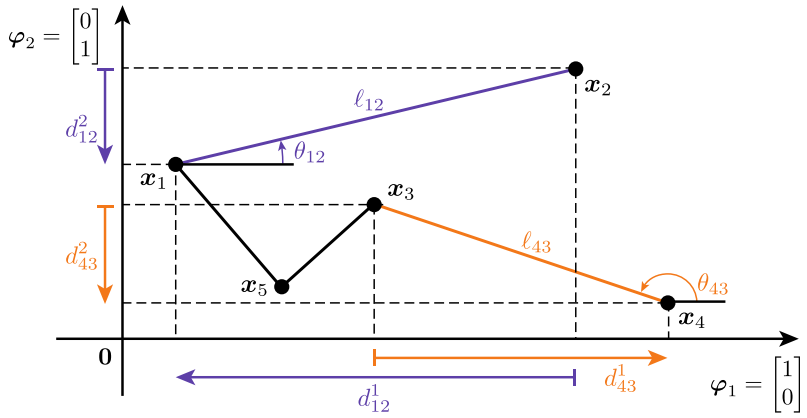


Figure 7.4: Localization setup in 2D. The difference vector $\boldsymbol{d}_{ij}$ between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined either with its distance ℓ_{ij} and directed angle θ_{ij}, or with its projections onto x- and y-axis, d^1_{ij} and d^2_{ij}, respectively.

θ_{ij} lies between 0 and 2π and is defined with respect to some common reference direction (see Fig. 7.4). Note that the angles are often measured in a coordinate system that is local and different for each node. Distributed algorithms can convert these relative angles to a common coordinate system [53].

Our goal is to estimate points $\widehat{\boldsymbol{x}}_i$ from a noisy subset of measured distances $\widetilde{\ell}_{ij}$ and angles $\widetilde{\theta}_{ij}$. Once again, we can leverage CDMs; the noisy coordinate differences are recovered from $\widetilde{\ell}_{ij}$ and $\widetilde{\theta}_{ij}$ as $\widetilde{\boldsymbol{d}}_{ij} = \left[\widetilde{\ell}_{ij}\cos\widetilde{\theta}_{ij},\quad \widetilde{\ell}_{ij}\sin\widetilde{\theta}_{ij}\right]^\top$. We observe that this is a 2D point recovery problem, which can be optimally solved with the splitting algorithm from Section 6.4.4; we can decompose the problem into two independent sub-problems and resolve each independently with (6.7). In this case, we always work in the canonical basis, hence $\boldsymbol{\Phi} = \boldsymbol{I}$.

7.4.3 Results

To evaluate the proposed CDM approach, we compare with the state-of-the-art in multimodal localization, E-MDS [117], and the most common approach for range-only measurements, MDS [100]. 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_ℓ and σ_θ, respectively. Note that the noise exceeding $\pm\pi$ will 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 Fig. 7.5. For a more convenient comparison, we slice the graphs at four different values of σ_ℓ and σ_θ. Fig. 7.5a shows the dependence of the RMSE on the distance noise level for two chosen levels of angle noise: low ($\sigma_\theta = 0.11$), and high ($\sigma_\theta = 0.5$). We observe that for low σ_θ it is advantageous to include angle measurements, since both multimodal methods achieve smaller error than MDS. For higher σ_θ, our method still outperforms MDS, except for very small values of the distance noise. The dependence of the RMSE on the angle noise level for

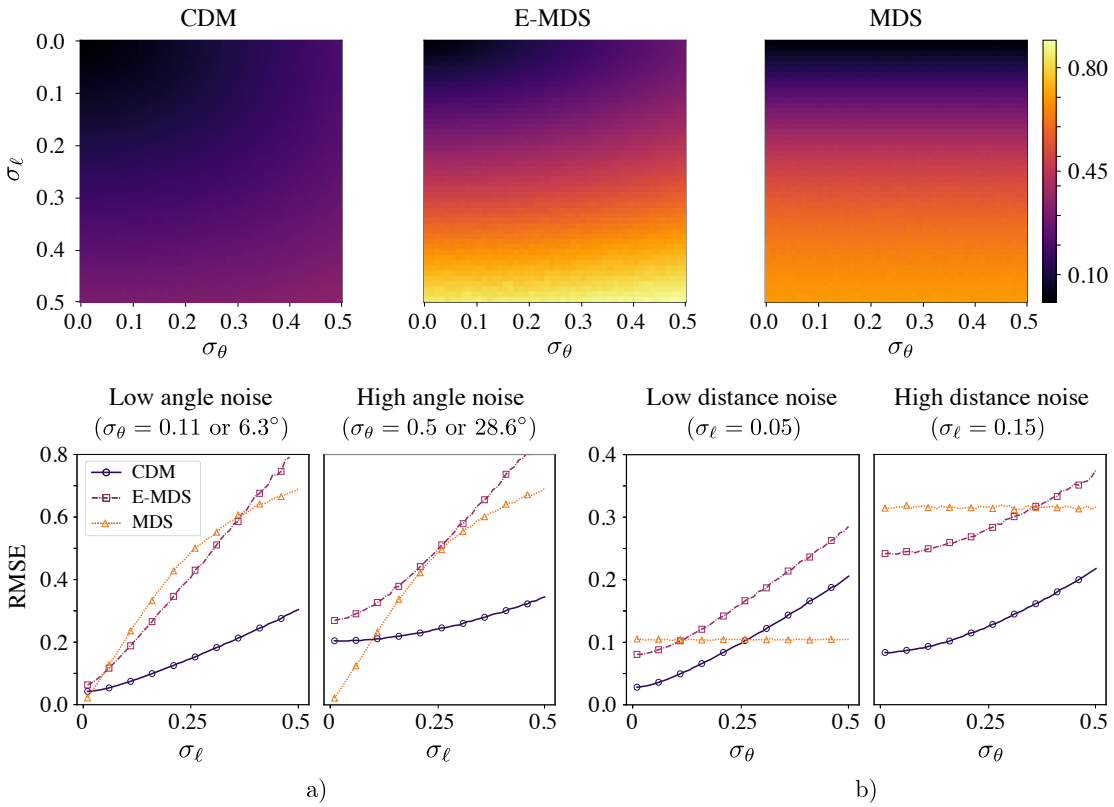


Figure 7.5: Comparison of the CDM with E-MDS and MDS for different distance and angle noise. a) Dependence of RMSE on σ_ℓ for two fixed σ_θ. b) Dependence of RMSE on σ_θ for two fixed σ_ℓ.

two chosen levels of distance noise is shown in Fig. 7.5b. For low distance noise ($\sigma_\ell = 0.05$), 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 ($\sigma_\ell = 0.15$), using angles significantly improves the result for all considered noise levels.

Numerical simulations show that the proposed algorithm based on CDMs surpasses the state-of-the-art multimodal localization method E-MDS for every pair of $(\sigma_\ell, \sigma_\theta)$. It also compares favorably with range-only based method MDS, except for the case of high σ_θ and low σ_ℓ, which is a relatively rare case in practice.

7.5 Super resolution phase retrieval for sparse sigals

7.5.1 Introduction

In our last application of CDMs we address the phase retrieval (PR) problem, where the goal is to recover a function from the magnitudes of its Fourier transform (FT). It is of great interest in many real-world scenarios, in which it is easier to measure the FT of a signal instead of the

signal itself. A phase loss or distortion can occur during the measurement process in many scientific disciplines, particularly those involving optics and communications. A few examples are X-ray crystallography [126], speckle imaging in astronomy [88] and blind channel estimation of multi-path communication channels [11].

In this chapter, we consider a realistic and challenging PR problem on sparse signals defined on a continuous domain. The sparsity assumption is legitimate and encountered in many applications; for example atoms in crystallography form a sparse structure and communication channels are often a collection of discrete echoes. We define a sparse signal by a set of N atoms determined by their locations and amplitudes.

Recently, several sparse PR algorithms have been proposed assuming a discrete spatial domain, often borrowing inspiration from the area of compressed sensing. Two notable examples are GrEedy Sparse PhAse Retrieval (GESPAR) [157], based on the 2-opt algorithm [40], and Two-stage Sparse Phase Retrieval (TSPR) [80], where the support is recovered by solving the discrete turnpike problem [41, 160]. Both algorithms differ from our approach in that their models are discrete and the locations of the spikes are bound to a discrete grid. Even though it was not designed with continuous setups in mind, TSPR can theoretically recover locations on a continuous domain. However, while it handles noise on the measured coefficients, it does not tolerate noise in the support, which makes it impractical for continuous setups. Other approaches with discrete support include [113], where the measurement matrix has random entries, PhaseLift [34], where the sampling matrix entries are independently sampled on the unit sphere, or [56], which is a variation of GESPAR based on the short-time Fourier transform.

The major benefit of having a continuous parametric model is that it enables estimation of the locations and amplitudes avoiding any discretization. In such a case, the achievable resolution is theoretically infinite and only limited by the noise corrupting the measurements. This is what we call *super resolution* phase retrieval.

Perhaps surprisingly, the continuous sparse phase retrieval problem has received little attention. The current literature offers only few results in this settings and they are not usable in practice mostly due to their instability to noise. Beinert et al. [12, 13] propose a super resolution approach based on the finite rate of innovation (FRI) framework, which is also one of the building blocks of the proposed algorithm in this chapter. Nonetheless, our work represents a significant improvement over the state of the art as it is robust to noise in the measurements.

7.5.1.1 Main contributions

We propose a three-stage algorithm that precisely determines a continuous sparse signal from the absolute value of its FT, even in presence of significant noise. First, we reconstruct the super-resolved auto-correlation function (ACF) from a set of its discrete Fourier coefficients. In the second step, we find the locations of Diracs that generate such ACF; this is known as the *support recovery* problem. Lastly, we estimate their amplitudes. In this chapter we focus on the second step of the algorithm for two reasons:

- It is closely related to the previously studied CDMs. The connection becomes clearer in Section 7.5.3 where we formally define the problem.
- It constitutes the critical element of the pipeline. In fact, the first step—the super resolution with FRI—is well covered in the literature, where theoretical analyses, extensive simulations in noisy scenarios and efficient denoising schemes have been proposed [52, 119, 135]. On

the other hand, the amplitude recovery, while being novel, only consists of simple algebraic manipulations that are not computationally costly.

Therefore, we describe the details of the proposed support recovery method, provide its complexity analysis together with a method to reduce its computational cost and propose a few improvements and variations of the algorithm to make it more robust to noise. We discuss the influence of the support configuration on the resulting reconstruction.

For a more detailed analysis of the other steps, we refer the reader to our paper [93]. In addition to the material presented in this chapter, we provide an approximated theoretical bound (confirmed by numerical simulations) to successfully recover the signal support in a noisy regime [93]. We show that the success rate of the proposed algorithm exhibits a sharp phase transition, which is a function of the number of sparse elements, and the noise affecting the inputs. In other words, given a randomly generated input, the algorithm has either a very high or very low probability of success. We also implement the PR pipeline for Charge Flipping, a state-of-the-art algorithm in crystallography, and demonstrate the superiority of our algorithm both in low and high noise regimes. As those topics go beyond the application of CDMs, we do not include them in this chapter.

7.5.2 Problem statement

We consider the most compact structure for a sparse signal: a set of N atoms defined by their locations $\boldsymbol{x}_n$ and their amplitudes c_n,

$$f(\boldsymbol{x}) = \sum_{n=1}^{N} c_n \phi(\boldsymbol{x} - \boldsymbol{x}_n) = f^s(\boldsymbol{x}) * \phi(\boldsymbol{x}),$$

where $f^s(\boldsymbol{x}) = \sum_{n=1}^{N} c_n \delta(\boldsymbol{x} - \boldsymbol{x}_n)$ represents the structure, $\boldsymbol{x}$ is a spatial variable defined over $\mathbb{R}^D$, D is the dimensionality of the signal, $\phi(\boldsymbol{x})$ is the scattering function induced by one atom and $*$ is the convolution operator.

An important component for solving the PR problem of sparse signals is the observation that the ACF structure of $f(\boldsymbol{x})$ is completely inherited from it:

$$\begin{aligned} a(\boldsymbol{x}) &= \sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j \psi(\boldsymbol{x} - (\boldsymbol{x}_i - \boldsymbol{x}_j)) \\ &= \left[\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j \delta(\boldsymbol{x} - (\boldsymbol{x}_i - \boldsymbol{x}_j))\right] * \psi(\boldsymbol{x}) \\ &= a^s(\boldsymbol{x}) * \psi(\boldsymbol{x}). \end{aligned} \tag{7.1}$$

The kernel $\psi(\boldsymbol{x})$ is the ACF of $\phi(\boldsymbol{x})$ and $a^s(\boldsymbol{x})$ is the ACF of the sparse structure of the train of Diracs $f^s(\boldsymbol{x})$. Moreover, we use the well-known fact that the ACF $a(\boldsymbol{x})$ of $f(\boldsymbol{x})$ is given by the inverse FT of $|F(\boldsymbol{\omega})|^2$:

$$a(\boldsymbol{x}) = f(\boldsymbol{x}) * f(-\boldsymbol{x}) = \mathcal{F}^{-1}\left[|F(\boldsymbol{\omega})|^2\right],$$

where $\mathcal{F}^{-1}$ is the inverse FT operator [174]. In the Fourier domain, we have

$$A(\boldsymbol{\omega}) = |F(\boldsymbol{\omega})|^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j \exp\left\{-j\boldsymbol{\omega}^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)\right\} |\Phi(\boldsymbol{\omega})|^2.$$

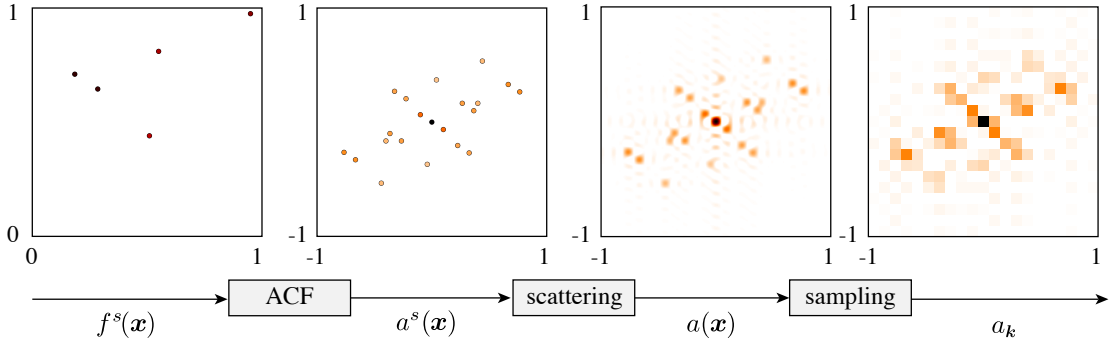


Figure 7.6: Typical PR measurement pipeline: the signal of interest $f^s(\boldsymbol{x})$ generates the autocorrelation function $a^s(\boldsymbol{x})$, which is first filtered by the scattering function $\psi(\boldsymbol{x})$ (here an ideal lowpass filter) to yield $a(\boldsymbol{x})$ and then sampled, resulting in $a_{\boldsymbol{k}}$. Note that the spatial samples $a_{\boldsymbol{k}}$ can be obtained via the inverse discrete FT of the Fourier samples $A_{\boldsymbol{\ell}}$, when the periodicity in the two domains holds. Darker colors represent higher intensities.

Due to limitations of the measurement setup, in practice we are usually only able to measure the absolute values of the samples of FT, that we denote $|F_{\boldsymbol{\ell}}|$, where $F_{\boldsymbol{\ell}} = F(\boldsymbol{\ell}\Omega)$, $\boldsymbol{\ell} = \mathbb{Z}^D$ and Ω is the sampling frequency. Analogously, we do not have an access to the whole $A(\boldsymbol{\omega})$, but only to its Fourier samples $A_{\boldsymbol{\ell}} = A(\boldsymbol{\ell}\Omega)$.

We now have all the ingredients to state our super resolution PR problem:

PROBLEM 7.1 Given Fourier samples $A_{\boldsymbol{\ell}}$ of the sparse ACF defined in (7.1), recover the support $\mathcal{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ and amplitudes $\{c_n\}_{n=1}^N$ determining the signal $f(\boldsymbol{x})$.

The PR acquisition pipeline can be summarized as the filtering of the ACF $a^s(\boldsymbol{x})$ followed by sampling, where the filtering represents the scattering operation, as illustrated in Fig. 7.6. The information we are interested in is hidden behind two walls: the convolution with the kernel $\psi(\boldsymbol{x})$ that spatially blurs the sparse structure of the ACF and the phase loss of the original sparse signal, $f^s(\boldsymbol{x})$, that usually characterizes any PR problem.

7.5.3 A three-stage approach

We propose to solve Problem 7.1 in three distinct stages: 1) reconstruct the continuous ACF $a(\boldsymbol{x})$ from a set of its discrete Fourier coefficients, 2) estimate the support $\mathcal{X}$ of $f(\boldsymbol{x})$ given $a(\boldsymbol{x})$, and 3) estimate its amplitudes $\{c_n\}_{n=1}^N$.

Sparsity has two antagonistic effects on PR: it makes the problem combinatorial and hence hard to solve, but at the same time enables a divide-and-conquer approach, in which we first recover the support $\{\boldsymbol{x}_n\}_{n=1}^N$ and then the amplitudes $\{c_n\}_{n=1}^N$ of $f(\boldsymbol{x})$. We argue that the support contains more information than the amplitudes, hence we choose to estimate it first. As an example, if all the atoms have the same amplitude, then only the support is useful to recover the original signal. On the other hand, if all the atoms have the same location, the problem is trivially solvable.

The first step is a classical sampling problem where we would like to fully characterize a continuous sparse signal from a set of discrete measurements:

PROBLEM 7.2 (SPARSE ACF SUPER RESOLUTION) Given samples $A_{\boldsymbol{\ell}}$ of the sparse ACF as defined in (7.1), recover its continuous version $a(\boldsymbol{x})$.

The second step is the support recovery problem, which is equivalent to labeling the entries of CDMs:

PROBLEM 7.3 (SUPPORT RECOVERY) Assume we are given the complete set of unlabeled differences $\mathcal{D} = \{\boldsymbol{d}_{i,j}\}_{i,j} = \{\boldsymbol{x}_i - \boldsymbol{x}_j\}_{i,j}$, recover the support $\mathcal{X} = \{\boldsymbol{x}_n\}_{n=1}^{N}$ of the sparse signal $f(\boldsymbol{x})$.

The set $\mathcal{D}$ is derived from the noisy samples $a_{\boldsymbol{k}}$ of the ACF $a(\boldsymbol{x})$ by solving Problem 7.2 and hence the unlabeled differences of Problem 7.3 are corrupted by noise. Noise is first introduced when measuring the samples of the ACF $a_{\boldsymbol{k}}$; it is independently distributed and added to the ACF samples, and not to the differences $\mathcal{D}$. Next, the algorithm that recovers the differences from the ACF deforms and amplifies the initial measurement noise. Hence, we always obtain a noisy and non-symmetrical set of differences, instead of the noiseless set $\mathcal{D}$.

Throughout this chapter, we generically refer to the sampling artifacts and their propagation due to the reconstruction algorithm as noise. In our specific case, we propose to solve the super resolution of the ACF with an FRI-based algorithm, whose noise analysis is not trivial. We refer the interested reader to [10, 22, 52, 119] for more details about it: theoretical bounds such as the Cramér-Rao bound [37] and the Barankin bound [9] are derived in [10, 22, 52]. Moreover, Monte Carlo simulations show the reconstruction noise distribution along with the theoretical Cramér-Rao bound [22].

While the literature clearly defines the noise affecting the distances $\boldsymbol{d}_{i,j}$ as neither Gaussian nor independently distributed, we model such a noise as i.i.d. Gaussian random variables,

$$\widetilde{\boldsymbol{d}}_{i,j} = \boldsymbol{d}_{i,j} + \boldsymbol{z}_{i,j},$$

where $\boldsymbol{z}_{i,j} \sim \mathcal{N}(\boldsymbol{0}, \sigma \boldsymbol{I})$.

If the labels i and j are known, we could simply arrange the coordinates of the differences $\widetilde{\boldsymbol{d}}_{i,j} = \left[\widetilde{d}^1_{i,j}, \dots, \widetilde{d}^D_{i,j}\right]$ in D CDMs $\{\widetilde{\boldsymbol{C}}^d\}_{d=1}^{D}$, such that $\widetilde{C}^d_{i,j} = \widetilde{d}^d_{i,j}$, where $d = 1, \dots, D$ is indexing the coordinates in the D-dimensional space. Then, we could apply (6.7) for every CDM independently and find the original points. The challenge of the PR problem is to find the labeling of the differences.

In what follows, we state a few interesting observations related to Problem 7.3. First, when we measure a set of differences, some information is inevitably lost.

OBSERVATION 7.4 A set of points can be reconstructed from their pairwise differences, even when labeled, only up to shifts and reflections.

To show that, we first translate and reflect the set of points $\mathcal{X}$ as $\mathcal{X}' = -\mathcal{X} + \bar{\boldsymbol{x}}$, where we overload the arithmetic operators on sets to transform each point as $\boldsymbol{x}'_n = -\boldsymbol{x}_n + \bar{\boldsymbol{x}}$. Then, the set of differences of the transformed points is equivalent to the original one,

$$\boldsymbol{d}'_{i,j} = \boldsymbol{x}'_i - \boldsymbol{x}'_j = -\boldsymbol{x}_i + \bar{\boldsymbol{x}} + \boldsymbol{x}_j - \bar{\boldsymbol{x}} = \boldsymbol{x}_j - \boldsymbol{x}_i = -\boldsymbol{d}_{i,j},$$

where the natural symmetry of $\mathcal{D}$ compensates for the negative sign.

Second, while excluding shifts and reflections does not lead to a unique solution in general, we can still prove uniqueness under certain assumptions.

OBSERVATION 7.5 Assume that the points $\boldsymbol{x}_n$ are drawn independently at random from a sufficiently smooth distribution, then the solution is unique [145].

Third, we say that there is a collision in the ACF when two different pairs of distinct points from $\mathcal{X}$ map to the same difference in $\mathcal{D}$. Since we consider a continuous domain for the support, it natively prevents the appearance of collisions.

OBSERVATION 7.6 If the locations of the points are independently drawn uniformly from a finite interval, then collisions in the ACF occur with probability zero.

Last, we note that a valid solution $\widehat{\mathcal{X}}$ is contained in the set of differences $\mathcal{D}$.

OBSERVATION 7.7 The set of differences $\mathcal{D}$ is a superset of $2N$ valid solutions $\widehat{\mathcal{X}}$ to Problem 7.3 and such solutions always contain the point zero, that is $\mathbf{0} \in \widehat{\mathcal{X}}$.

To verify this, we pick an element of the support, e.g. $\boldsymbol{x}_j \in \mathcal{X}$, and build the following tentative solution,

$$\widehat{\mathcal{X}} = \{\boldsymbol{x}_i - \boldsymbol{x}_j \mid i = 1, \dots, N\}. \tag{7.2}$$

Then, we notice that 1) $\widehat{\mathcal{X}}$ is a valid solution with the shift fixed as $-\boldsymbol{x}_j$, 2) $\widehat{\mathcal{X}} \subset \mathcal{D}$ and 3) we have a solution for every element of $\mathcal{X}$. Moreover, due to the symmetry of the ACF, the set

$$\{\boldsymbol{x}_j - \boldsymbol{x}_i \mid i = 1, \dots, N\} \subset \mathcal{D}$$

is also a valid solution, so we reach the aforementioned $2N$ solutions.

In the context of CDMs, Observation 7.7 is even more apparent. Let us start with the one-dimensional case, $D = 1$. Due to Property P.6 from Section 6.2, we know that every column of a CDM $\boldsymbol{C}$ is a solution set. In other words, a CDM created from the jth column $\boldsymbol{x} = \{C_{ij}\}_{i=1}^{N}$ is identical to $\boldsymbol{C}$. Moreover, we know that the entries of $\boldsymbol{C}$ are the differences in $\mathcal{D}$. Therefore, every column of $\boldsymbol{C}$ creates the support $\widehat{\mathcal{X}}$, and since $\boldsymbol{C}$ is a hollow matrix, every solution set contains the point 0. The same is valid for $D > 1$, where the entry (i, j) in every CDM $\boldsymbol{C}^d$, $d = 1, \dots, D$

Algorithm 7.1 Support recovery

Input: A set of $M = N^2 - N + 1$ differences $\widetilde{\mathcal{D}} = \{\widetilde{\boldsymbol{d}}_m\}_{m=1}^{M}$ ordered by their norms.
Output: A set of N points $\widehat{\mathcal{X}}$ such that their pairwise differences generate $\widetilde{\mathcal{D}}$.

$\widehat{\mathcal{X}}_2 = \{\mathbf{0}, \widetilde{\boldsymbol{d}}_M\}$
$\mathcal{P}_2 = \widetilde{\mathcal{D}} \setminus \{\widetilde{\boldsymbol{d}}_1, \widetilde{\boldsymbol{d}}_M\}$
for $n = 2, \dots, N-1$ **do**
  $\widehat{\boldsymbol{x}}_{n+1} = \arg\min_{\boldsymbol{p} \in \mathcal{P}_n} \sum_{\widehat{\boldsymbol{x}} \in \widehat{\mathcal{X}}_n} \min_{\widetilde{\boldsymbol{d}} \in \widetilde{\mathcal{D}}} \left\| \boldsymbol{p} - \widehat{\boldsymbol{x}} - \widetilde{\boldsymbol{d}} \right\|^2$
  $\widehat{\mathcal{X}}_{n+1} = \widehat{\mathcal{X}}_n \cup \widehat{\boldsymbol{x}}_{n+1}$
  $\mathcal{P}_{n+1} = \mathcal{P}_n \setminus \widehat{\boldsymbol{x}}_{n+1}$
end for
return $\widehat{\mathcal{X}}_N$

is the dth coordinate of the ith point in the solution set created from the jth column. As the zero-element is located on the diagonal of every $\boldsymbol{C}^d$, the point $\mathbf{0}$ is in the solution set.

Once the support $\widehat{\mathcal{X}}$ of the solution has been retrieved, it remains to find the amplitudes $\{c_n\}_{n=1}^{N}$ of the signal $f(\boldsymbol{x})$.

Problem 7.8 (Amplitude recovery) Given an ACF $a(\boldsymbol{x})$ as defined in (7.1) together with the estimated support $\widehat{\mathcal{X}}$ of $f(\boldsymbol{x})$, find the amplitudes $\{c_n\}_{n=1}^{N}$.

7.5.4 Algorithms

In this section, we lay down our solutions to Problems 7.2, 7.3 and 7.8, effectively providing an end-to-end framework to solve the sparse PR problem.

7.5.4.1 ACF super resolution

When we look at (7.1), we notice that $a(\boldsymbol{x})$ is completely defined by the locations $\boldsymbol{x}_i - \boldsymbol{x}_j$ and the amplitudes $c_i c_j$. Hence, we can recast Problem 7.2 as a parametric estimation problem given the measured samples $A_{\boldsymbol{\ell}}$ of the FT of the ACF. An effective existing approach is known as finite rate of innovation sampling [51, 173]. The higher dimensional case was first discussed by Maravic [119], who proposed a first algorithm requiring $\mathcal{O}(N^D)$ samples, where D is the number of dimensions. More recently, Pan et al. [136] came up with a multidimensional reconstruction algorithm using only $\mathcal{O}(N)$ samples.

7.5.4.2 Support recovery

For the recovery of the support, i.e. for the estimation of a column of a CDM, we propose a novel greedy algorithm that is initialized with a partial solution $\widehat{\mathcal{X}}_2$, which contains two locations. At any given iteration n, we generate a partial solution $\widehat{\mathcal{X}}_{n+1}$ composed of $n+1$ locations, hence the algorithm has a total of $N-2$ iterations indexed from 2 to $N-1$.

Initialization. We denote the set of measured differences as $\widetilde{\mathcal{D}} = \{\widetilde{\boldsymbol{d}}_{i,j}\}_{i,j}$. For simplicity of notation, we convert the pairs of indices (i,j) to $m \in \{1,\dots,M\}$, where $M = N^2 - N + 1$, and order them such that $\|\widetilde{\boldsymbol{d}}_1\| \leq \|\widetilde{\boldsymbol{d}}_2\| \leq \ldots \leq \|\widetilde{\boldsymbol{d}}_M\|$. We do not assume any ordering on the elements of $\mathcal{X}$.

From Observation 7.7, we know that the solution set $\widehat{\mathcal{X}}$ is contained in $\widetilde{\mathcal{D}}$ and $\mathbf{0} \in \mathcal{X}$; this gives us the first point of the solution, that is $\widehat{\boldsymbol{x}}_1 = \mathbf{0}$. Next, we identify the element $\widetilde{\boldsymbol{d}}_M$ in $\widetilde{\mathcal{D}}$ with the largest norm, so that we maximize the noise resilience of our algorithm. Indeed, assuming that the locations are corrupted by identically distributed noise, picking the largest norm ensures the maximal SNR of our initial solution. Note that the value $\widetilde{\boldsymbol{d}}_M$ is the noisy difference between two unknown locations of $f(\boldsymbol{x})$; without loss of generality, we call them $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The elements $\widehat{\boldsymbol{x}}_1 = \mathbf{0}$ and $\widehat{\boldsymbol{x}}_2 = \widetilde{\boldsymbol{d}}_M$ are nothing but $\boldsymbol{x}_1$ and $\boldsymbol{x}_2 + \boldsymbol{z}_{2,1}$ translated by $-\boldsymbol{x}_1$. Therefore, we are always guaranteed that the initialized solution $\widehat{\mathcal{X}}_2 = \{\mathbf{0}, \widetilde{\boldsymbol{d}}_M\}$ is a (noisy) subset of the set of locations $\mathcal{X} - \boldsymbol{x}_1$.

Referring again to Observation 7.7, we know that the set of differences $\widetilde{\mathcal{D}}$ contains the rest of the points $\{\boldsymbol{x}_n - \boldsymbol{x}_1 + \boldsymbol{z}_{n,1}\}_{n=3}^{N}$, that should belong to the final solution $\widehat{\mathcal{X}} = \widehat{\mathcal{X}}_N$. Furthermore, since we do not want to duplicate points in $\widehat{\mathcal{X}}_n$, we initialize a set of possible elements of the solution $\mathcal{P}_2 = \widetilde{\mathcal{D}} \setminus \{\widetilde{\boldsymbol{d}}_1, \widetilde{\boldsymbol{d}}_M\}$. Due to noise, the vector $\mathbf{0}$ is not in $\widetilde{\mathcal{D}}$, so we remove the closest element $\widetilde{\boldsymbol{d}}_1$.

Main algorithm. We would like to identify the element $\widehat{\boldsymbol{x}}_{n+1} \in \mathcal{P}_n$ to be added to the current solution set $\widehat{\mathcal{X}}_n$, that is $\widehat{\mathcal{X}}_{n+1} = \widehat{\mathcal{X}}_n \cup \widehat{\boldsymbol{x}}_{n+1}$. In the noiseless case, this new element of the solution should be chosen such that the pairwise differences between the elements of $\widehat{\mathcal{X}}_{n+1}$ form a subset of $\mathcal{D}$.

This intuition can be generalized to the noisy case: we would like to identify the element $\widehat{\boldsymbol{x}}_{n+1} \in \mathcal{P}_n$ such that the set of pairwise differences of the points in $\widehat{\mathcal{X}}_{n+1} = \widehat{\mathcal{X}}_n \cup \widehat{\boldsymbol{x}}_{n+1}$ is *the closest to be a subset* of the measured $\widetilde{\mathcal{D}}$. To that end, we propose to generalize the concept of subset to a noisy environment by searching for the differences in $\widetilde{\mathcal{D}}$ that are closest in ℓ^2-norm to the pairwise differences of the elements in $\widehat{\mathcal{X}}_{n+1}$.

At each step n, we identify the element in $\mathcal{P}_n$ that, when added to the partial solution $\widehat{\mathcal{X}}_n$, minimizes the error with respect to the measured set of differences $\widetilde{\mathcal{D}}$. More precisely, at every iteration n we solve the following optimization problem,

$$\widehat{\boldsymbol{x}}_{n+1} = \underset{\boldsymbol{p}\in\mathcal{P}_n}{\arg\min} \sum_{\widehat{\boldsymbol{x}}\in\widehat{\mathcal{X}}_n} \min_{\widetilde{\boldsymbol{d}}\in\widetilde{\mathcal{D}}} \left\| \boldsymbol{p} - \widehat{\boldsymbol{x}} - \widetilde{\boldsymbol{d}} \right\|^2 . \tag{7.3}$$

This procedure is summarized in Algorithm 7.1 and its application on the ACF $a^s(\boldsymbol{x})$ from Fig. 7.6 is illustrated in Fig. 7.7.

7.5.4.3 Amplitude recovery

Due to Observation 7.6, we assume that there are no collisions and suggest a simple but efficient algebraic solution to Problem 7.8, inspired by the work of Ranieri et al. [145]. While their method relies on a matrix inversion step to solve the problem, here we propose to work in the logarithmic domain. Numerical simulations have shown that it is both faster and more robust to noise.

Let $\boldsymbol{c} = [c_1, c_2, \dots, c_N]^\top$ be a vector made of the amplitudes to be recovered. If we define a matrix $\boldsymbol{\Gamma} = \boldsymbol{c}\boldsymbol{c}^\top$, all the elements outside of the diagonal of such a matrix are the amplitudes of the measured ACF, that is $\Gamma_{i,j} = c_i c_j$. Notice that we cannot observe the diagonal entries

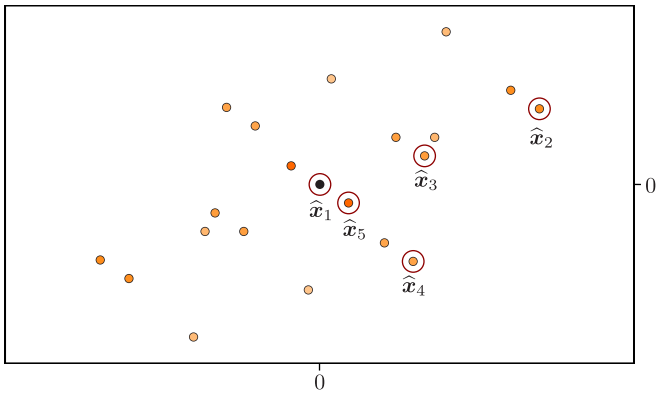


Figure 7.7: 2D instance of Algorithm 7.1 on the ACF $a^s(\boldsymbol{x})$ from Fig. 7.6. We start by setting $\widehat{\boldsymbol{x}}_1 = \mathbf{0}$ and identifying $\widehat{\boldsymbol{x}}_2$, the point with the largest norm. Points $\widehat{\boldsymbol{x}}_3$ to $\widehat{\boldsymbol{x}}_5$ are then selected in a greedy way according to (7.3). The solution coincides with the initial signal $f^s(\boldsymbol{x})$ displayed in Fig. 7.6.

$\Gamma_{i,i} = c_{i,i}^2$ as we just have access to their sum $a_{\mathbf{0}}^s = \sum_i c_{i,i}^2$, which is the value of the ACF at $\mathbf{0}$. This is unfortunate since they are precisely the values we are interested in, up to a squaring operator.

We recast Problem 7.8 as a matrix completion problem, where we would like to estimate the diagonal entries $\Gamma_{i,i}$ under the constraint of $\boldsymbol{\Gamma}$ being a rank-one matrix. The first step of our proposed method is to introduce a matrix $\boldsymbol{L}$ such that

$$L_{i,j} = \begin{cases} \log(\Gamma_{i,j}) = \ell_i + \ell_j & \text{for } i \neq j \\ 0 & \text{otherwise,} \end{cases}$$

where $\ell_i = \log(c_i)$. The sum of the ith row of $\boldsymbol{L}$ is given by

$$\sum_{j=1}^{N} L_{i,j} = (N-1)\ell_i + \sum_{j=1}^{N} \ell_j - \ell_i = (N-2)\ell_i + \sum_{j=1}^{N} \ell_j,$$

where the term $\sum_j \ell_j$ does not vary between rows. Hence, its value can be obtained from summing all the entries in $\boldsymbol{L}$,

$$s = \sum_{i=1}^{N}\sum_{j=1}^{N} L_{i,j} = (N-2)\sum_{i=1}^{N} \ell_i + N\sum_{j=1}^{N} \ell_j = 2(N-1)\sum_{j=1}^{N} \ell_j.$$

Then, we recover the vector $\boldsymbol{\ell} = [\ell_1, \ell_2, \ldots, \ell_N]^\top$ for $N > 2$ as

$$\boldsymbol{\ell} = \frac{1}{N-2}\left(\boldsymbol{L}^\top \mathbf{1} - \frac{s}{2(N-1)}\mathbf{1}\right),$$

where $\mathbf{1}$ is the all-ones vector.[1] Finally, it suffices to compute $c_i = \exp(\ell_i)$ to retrieve the amplitudes.

[1] When $N = 2$, the entries ℓ_1, ℓ_2 can be recovered by solving a system of two equations.

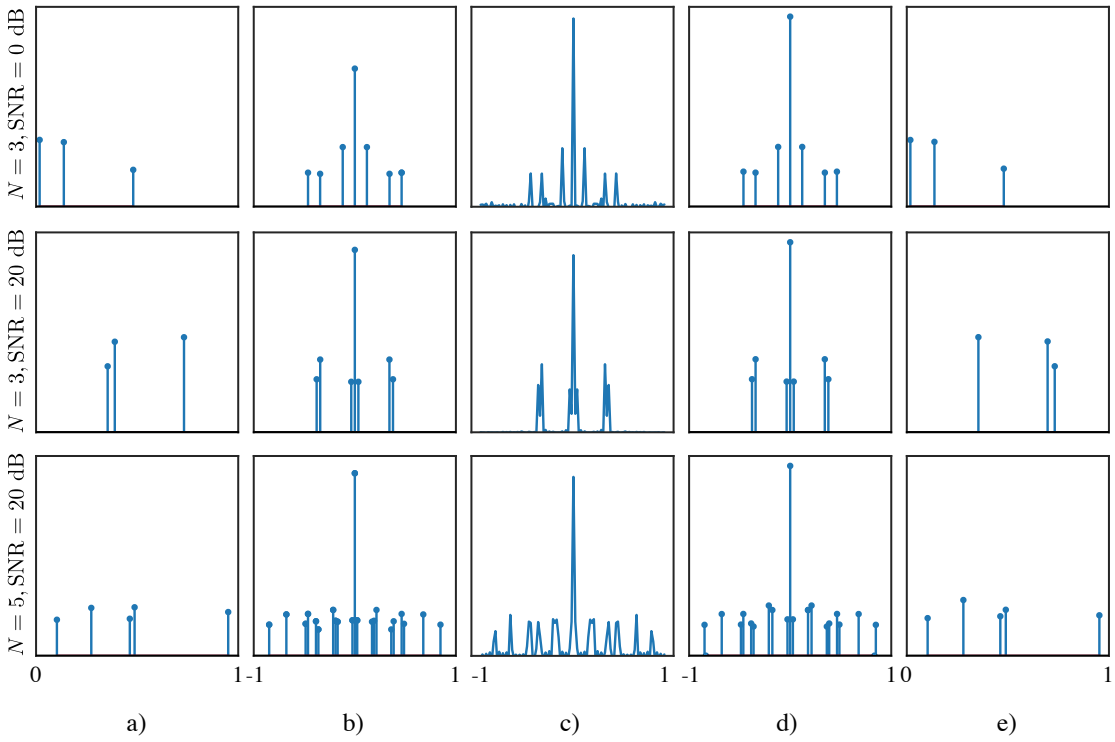


Figure 7.8: Examples of our algorithm in 1D for different values of N and different noise regimes: a) original points; b) corresponding continuous ACF; c) discrete noisy ACF with 100 samples (sinc sampling kernel used); d) output of the FRI-based super resolution algorithm (Section 7.5.4.1); e) result of the support and amplitude recovery algorithms (Sections 7.5.4.2 and 7.5.4.3).

Note that this solution assumes that $\mathbf{\Gamma}$ is symmetric; this might not be the case in a noisy setup, but we enforce it by replacing $\mathbf{\Gamma}$ with $\frac{1}{2}(\mathbf{\Gamma} + \mathbf{\Gamma}^\top)$.

Putting all pieces together, these three algorithms from Sections 7.5.4.1, 7.5.4.2 and 7.5.4.3 combine to enable the recovery of a continuous signal from its noisy sampled ACF; Fig. 7.8 illustrates a few examples of recoveries of trains of Diracs based on the combination of these three steps.

7.5.5 Complexity analysis

Algorithm 7.1 has N rounds. In each of these rounds, we go through all points in the existing solution set $\widehat{\mathcal{X}}_n$, and for each point we compute the difference with all the values in $\widetilde{\mathcal{D}}$. Since there are $\mathcal{O}(N)$ points in $\widehat{\mathcal{X}}_n$ and $\mathcal{O}(N^2)$ elements in $\widetilde{\mathcal{D}}$, this is done in $\mathcal{O}(N^3)$ operations. Furthermore, for each of these computed differences, we need to find the closest element in $\widetilde{\mathcal{D}}$, which requires additional $\mathcal{O}(N^2)$ comparisons. In total, the complexity of our algorithm is $\mathcal{O}(N^6)$. Even though this is high and limits the field of application to reasonable sizes, it compares favorably to an exhaustive search strategy, which grows exponentially with N.

It is possible to trade time complexity for storage complexity. Indeed, we observe that we

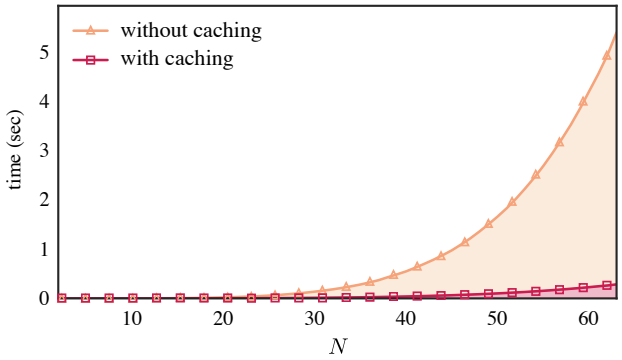


Figure 7.9: Comparison of the average run time of the original algorithm and its cached version. The times reported are the average of 100 runs of the algorithm. The dashed lines represent curves of the form CN^{α} that are fitted to the data. For the method without caching, we have $C = 4.25 \cdot 10^{-6}$ and $\alpha = 5.06$, while for the method with caching we have $C = 3.88 \cdot 10^{-6}$ and $\alpha = 4.37$. Remark how the caching is reducing the polynomial degree of the computational cost by approximately one.

compute at each round the following values

$$\widetilde{\boldsymbol{d}}_{i,j} = \underset{\widetilde{\boldsymbol{d}} \in \widetilde{\mathcal{D}}}{\arg\min} \, \|\boldsymbol{p}_j - \widehat{\boldsymbol{x}}_i - \widetilde{\boldsymbol{d}}\|^2, \tag{7.4}$$

for every point $\widehat{\boldsymbol{x}}_i \in \widehat{\mathcal{X}}_n$ and candidate $\boldsymbol{p}_j \in \mathcal{P}_n$. However, since we are just moving one element from $\mathcal{P}_n$ to $\widehat{\mathcal{X}}_{n+1}$ at each iteration, we propose to cache the values (7.4) in a lookup table to reduce the total computational cost. By doing so, we only need to update each $\widetilde{\boldsymbol{d}}_{i,j}$ when the corresponding candidate $\boldsymbol{p}_j$ is removed from $\mathcal{P}_n$ to be added to $\widehat{\mathcal{X}}_{n+1}$.

The theoretical complexity when caching $\widetilde{\boldsymbol{d}}_{i,j}$ is not trivial to analyze, but in practice we notice a significant improvement, as illustrated in Fig. 7.9.

7.5.6 Improving noise resilience

We now discuss strategies and variations of our support recovery algorithm aiming at improving the quality of the solution in noisy settings.

7.5.6.1 Deleting solutions from the set of differences

When a new point $\widehat{\boldsymbol{x}}_{n+1}$ is added to $\widehat{\mathcal{X}}_n$, Algorithm 7.1 ignores some useful information. Assuming that there are no collisions and no noise, we know that the values $\widehat{\mathcal{X}}_n - \widehat{\boldsymbol{x}}_{n+1}$ and $\widehat{\boldsymbol{x}}_{n+1} - \widehat{\mathcal{X}}_n$ in $\mathcal{D}$ cannot belong to the solution $\widehat{\mathcal{X}}$. Thus, as soon as $\widehat{\boldsymbol{x}}_{n+1}$ is added to the solution set, we can remove all values of the form $\widehat{\mathcal{X}}_n - \widehat{\boldsymbol{x}}_{n+1}$ and $\widehat{\boldsymbol{x}}_{n+1} - \widehat{\mathcal{X}}_n$ from $\mathcal{D}$.

The same reasoning applies to the noisy case, but we pick the closest values in $\widetilde{\mathcal{D}}$ as we do not have exact differences. More formally, when we add a new point $\widehat{\boldsymbol{x}}_{n+1}$ to the solution $\widehat{\mathcal{X}}_n$, we dispose of the following $2n$ elements of $\widetilde{\mathcal{D}}$,

$$\widetilde{\boldsymbol{d}}^* = \underset{\widetilde{\boldsymbol{d}} \in \widetilde{\mathcal{D}}}{\arg\min} \, \| \pm \widehat{\boldsymbol{x}} \mp \widehat{\boldsymbol{x}}_{n+1} - \widetilde{\boldsymbol{d}}\|^2, \quad \forall \widehat{\boldsymbol{x}} \in \widehat{\mathcal{X}}_n.$$

This approach results in two opposing effects. On one hand, we introduce the risk of erroneously discarding a point $\widetilde{\boldsymbol{d}}^*$ that belongs to the solution. On the other hand, we are pruning many elements out of $\widetilde{\mathcal{D}}$ and naturally reduce the risk of picking an erroneous candidate later on in the recovery process. As we will show in Section 7.5.6.4, the benefits out-weight the risks.

7.5.6.2 Symmetric cost function

Next, we replace the cost function (7.3) with a symmetric one to leverage the natural symmetry of the ACF.

In Algorithm 7.1, we search for the vectors in $\widetilde{\mathcal{D}}$ closest to the computed differences $\boldsymbol{p}-\widehat{\mathcal{X}}_n$ for each candidate $\boldsymbol{p}$. We strengthen its noise resilience by jointly searching for the vectors closest to $\mp\widehat{\mathcal{X}}_n \pm \boldsymbol{p}$ and choosing the candidate $\boldsymbol{p}$ that minimizes the sum of both errors. Specifically, we rewrite the cost function (7.3) as

$$\widehat{\boldsymbol{x}}_{n+1} = \operatorname*{arg\,min}_{\boldsymbol{p}\in\mathcal{P}_n} \sum_{\widehat{\boldsymbol{x}}\in\widehat{\mathcal{X}}_n} \min_{\widetilde{\boldsymbol{d}},\widetilde{\boldsymbol{d}}'\in\widetilde{\mathcal{D}}} \left\|\boldsymbol{p}-\widehat{\boldsymbol{x}}-\widetilde{\boldsymbol{d}}\right\|^2 + \left\|\widehat{\boldsymbol{x}}-\boldsymbol{p}-\widetilde{\boldsymbol{d}}'\right\|^2.$$

We stress that this improvement is compatible with the idea of caching introduced in Section 7.5.5. We can in fact cache the following pairs

$$(\widetilde{\boldsymbol{d}},\widetilde{\boldsymbol{d}}')_{i,j} = \operatorname*{arg\,min}_{\widetilde{\boldsymbol{d}},\widetilde{\boldsymbol{d}}'\in\widetilde{\mathcal{D}}} \|\boldsymbol{p}_j-\widehat{\boldsymbol{x}}_i-\widetilde{\boldsymbol{d}}\|^2 + \|\widehat{\boldsymbol{x}}_i-\boldsymbol{p}_j-\widetilde{\boldsymbol{d}}'\|^2,$$

for each $\widehat{\boldsymbol{x}}_i \in \widehat{\mathcal{X}}_n$ and $\boldsymbol{p}_j \in \mathcal{P}_n$ and recompute them when $\boldsymbol{p}_j$ gets added to the solution $\widehat{\mathcal{X}}_{n+1}$.

7.5.6.3 Denoising of partial solutions

At each iteration n of Algorithm 7.1, we have a partial solution $\widehat{\mathcal{X}}_{n+1}$ and, from (7.3), we identify for each pair $\widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{x}}_j \in \widehat{\mathcal{X}}_{n+1}$ a difference $\widehat{\boldsymbol{d}}_{i,j}$ that is the closest to $\widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_j$. In other words, we are simultaneously labeling the differences $\widehat{\boldsymbol{d}}_{i,j}$ using our current partial solution and arranging them in D CDMs $\{\boldsymbol{C}^d\}_{d=1}^D$ such that $\widehat{\boldsymbol{d}}_{i,j} = \begin{bmatrix} C^1_{i,j} & C^2_{i,j} & \dots & C^D_{i,j} \end{bmatrix}^\top$.

The labeling is completed as k reaches the final iteration $N-1$. The CDMs can be exploited to *denoise* the set $\widehat{\mathcal{X}}_{n+1}$ as they provide unused additional constraints and mitigate the error propagation between the iterations. More precisely, at every step we propose to replace $\widehat{\mathcal{X}}_{n+1}$ with a set of points $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^{n+1}$ that minimize the following cost function

$$J\big(\{\widehat{\boldsymbol{x}}_i\}_{i=1}^{n+1}\big) = \sum_{i,j} \|\widehat{\boldsymbol{d}}_{i,j} - (\widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_j)\|^2 \overset{(a)}{=} \sum_{i,j}\sum_{d} \|C^d_{i,j} - (\widehat{x}^d_i - \widehat{x}^d_j)\|^2. \tag{7.5}$$

This turns out to be the function minimized in the multidimensional CDM problem introduced in Section 6.4, where (a) follows from the fact that the frame $\boldsymbol{\Phi}_0$ is an orthonormal basis. The solution to (7.5) is derived in closed-form by setting its first derivative to 0. It results in the matrix form (6.11), where $\boldsymbol{B}$ indicates all pairs of points in the solution set $\widehat{\mathcal{X}}_{n+1}$, and $\boldsymbol{\Phi}_0$ contains the standard basis for D-dimensional space. As discussed in Chapter 6, the solution is optimal when the differences are corrupted by additive Gaussian noise, as it is the case here.

Observe that in every step we construct complete CDMs with one additional row and column for every new point in the solution set. Again referring to Chapter 6, we showed that for the

complete case, the point recovery is as simple as taking the average of the differences related to each point $\widehat{\boldsymbol{x}}_i \in \widehat{\mathcal{X}}_{n+1}$ as

$$\widehat{\boldsymbol{x}}_i = \frac{1}{n+1} \sum_{j=1}^{n+1} \widehat{\boldsymbol{d}}_{i,j}.$$

To see why this works, we separate the sum as

$$\frac{1}{n+1} \sum_{j=1}^{n+1} \widehat{\boldsymbol{d}}_{i,j} = \boldsymbol{x}_i - \frac{1}{n+1} \sum_{j=1}^{n+1} \boldsymbol{x}_j + \frac{1}{n+1} \sum_{j=1}^{n+1} \boldsymbol{z}_{i,j}.$$

The sum $-\frac{1}{n+1} \sum_{j=1}^{n+1} \boldsymbol{x}_j$ is the constant translation for all points $\widehat{\boldsymbol{x}}_i$. The consequence of this approach is that the total noise is reduced as we average its different realizations over $n+1$ values. Note that since Algorithm 7.1 assumes that $\widehat{\boldsymbol{x}}_1 = \mathbf{0}$ in $\widehat{\mathcal{X}}_n$, we also translate back all the points by $-\widehat{\boldsymbol{x}}_1$ after each update.

Unfortunately, the idea of caching the differences introduced in Section 7.5.5 is not compatible with the denoising of the partial solutions. As at each step we modify the partial solution set $\widehat{\mathcal{X}}_n$, the differences between $\widehat{\mathcal{X}}_n$ and $\widetilde{\mathcal{D}}$ change accordingly, which makes it impossible to cache them. Hence, there exists a hard trade-off between quality and complexity, and we should pick the right strategy depending on the requirements of each specific practical scenario.

7.5.6.4 Comparison of improvement strategies

Last, we evaluate the significance of our proposed improvements on Algorithm 7.1. We quantify the results using the index error, as well as the ℓ^2 error. We define the index error as a binary metric that is equal to 0 if the solution set $\widehat{\mathcal{X}}$ is of the form (7.2), and 1 otherwise. This error can be used to measure the probability of success of Algorithm 7.1. The ℓ^2 error we define as the ℓ^2-norm of the difference between the underlying points $\mathcal{X}$ and their estimation $\widehat{\mathcal{X}}$.[2]

The comparison of the different improvement strategies is illustrated in Fig. 7.10. In the experiment, we draw N one-dimensional points uniformly at random from the interval $[0, 1]$ and add Gaussian noise $\mathcal{N}(0, \sigma^2)$ on their pairwise differences. We perform two experiments, with $N = 6$ and $N = 10$. We run Algorithm 7.1 and the proposed improvements for different noise levels σ. It is clear that all the proposed strategies enhance the original algorithm, with respect to both the index error and the ℓ^2 error.

Moreover, we also observe that different strategies combine constructively: for example when $N = 6$, the symmetric cost function decreases the ℓ^2 error by 5% on average, while deleting solutions from the set of differences improves the results by 27% on average. When combined together, the average error decreases by 59%. Including the denoising further enhances the algorithm, as the average error decreases by 62%. Similarly, for the index error there is an evident shift between the phase transitions of the original algorithm with and without improvements. In Table 7.2 we quantify the performance of each of these enhancing methods for $N = 6$ and $N = 10$. We can conclude that different strategies combine constructively and improve the performance of the original algorithm in both cases.

[2]This requires to first align the two sets of points $\mathcal{X}$ and $\widehat{\mathcal{X}}$ by minimizing the ℓ^2-norm between their elements, subject to any shift and/or reflection.

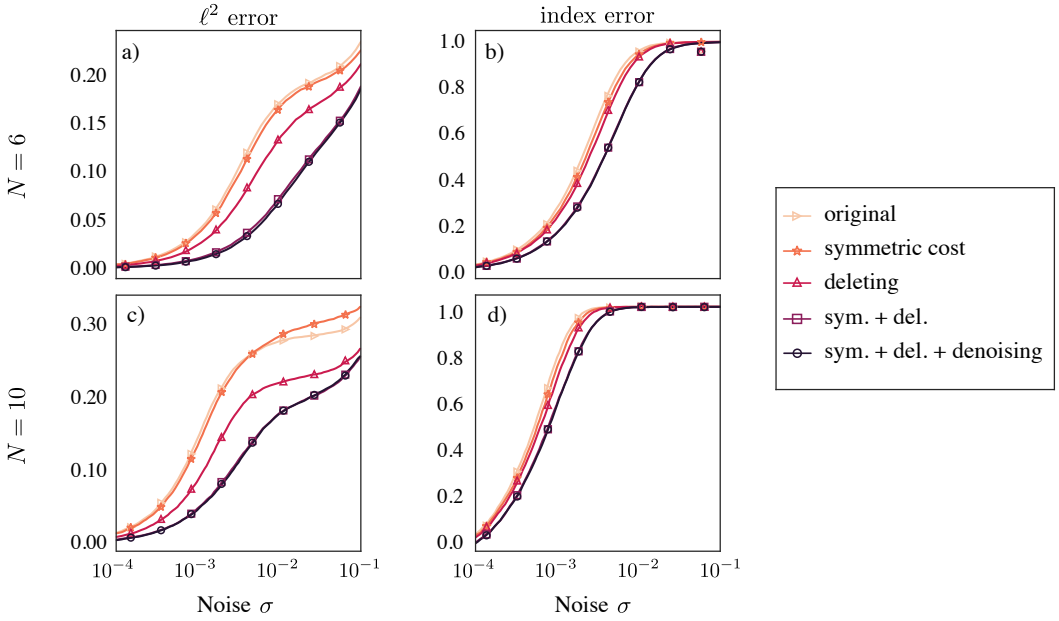


Figure 7.10: Average error for the different combinations of improvements of the algorithm. We create $\mathcal{X}$ from N 1D points chosen uniformly at random from the interval $[0, 1]$, create $\mathcal{D}$ accordingly and add Gaussian noise $\mathcal{N}(0, \sigma^2)$ to its elements. In a) and c) we plot the ℓ^2 error for different levels of noise σ and different improvements of the original algorithm. In b) and d) we show the index error.

	sym.	del.	sym. + del.	sym. + del. + denoising
$N = 6$	5%	27%	59%	62%
$N = 10$	1%	28%	48%	49%

Table 7.2: The average decrease in ℓ^2 error for different improvement strategies of the original Algorithm 7.1.

7.5.7 Influence of point locations

When running numerical simulations, we noticed that some configurations of points are easier to recover than others. In this section, we perform a small experiment to visualize the challenges posed by certain configurations.

We consider a low-complexity setup ($N = 4$, $D = 1$), fix the support boundaries, that is $x_1 = 0$ and $x_2 = 1$, and study the reconstruction error for various pairs $(x_3, x_4) \in [0, 1]^2$. We generate several instances of this problem and perturb the differences in $\mathcal{D}$ with additive Gaussian noise with zero mean and $\sigma = 0.01$. We measure the performance of Algorithm 7.1 (with all the improvements introduced in Section 7.5.6) using both the index and the ℓ^2 error. The average errors are then shown in Fig. 7.11, where we observe that there exist some combinations of points that lead to a significantly higher error.

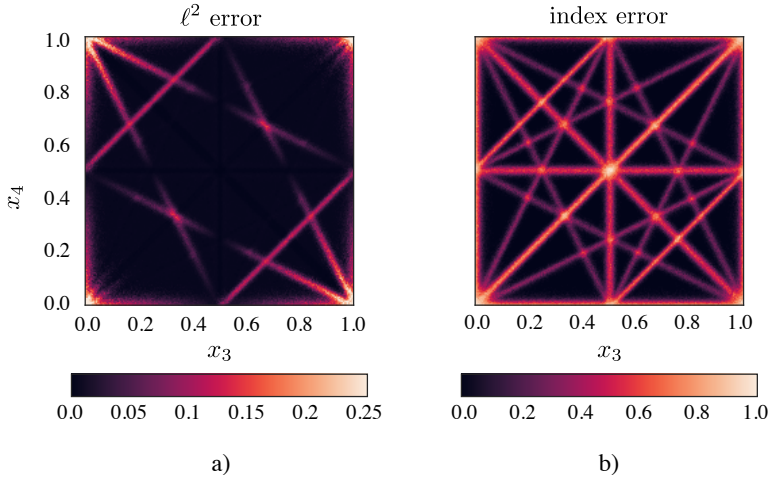


Figure 7.11: Influence of the points' locations on the estimation errors. We solve a 1D instance of the problem with $N = 4$, $x_1 = 0$, and $x_2 = 1$. The locations x_3 and x_4 vary along the x- and y-axis.

We now develop intuition about a few interesting cases that emerged from the previous experiment. For the sake of simplicity, we consider a noiseless setting where collisions in the ACF or non-uniqueness of the solution are the only causes of challenging configurations.

1. *Collision between a difference and a point.* When a difference and a point collide, it can happen that the difference is mistaken for the point. This does not influence the ℓ^2 error, but causes an index error. An example of such a case is when $x_3 = x_4$ (the main diagonal in Fig. 7.11): both the difference $x_4 - x_3$ and x_1 have value 0. As a consequence, the sets $\mathcal{X}' = \{x_1, x_2, x_3, x_4\}$ and $\mathcal{X}'' = \{x_4 - x_3, x_2, x_3, x_4\}$ are both equal to $\mathcal{X} = \{0, 1, x_3, x_3\}$, but $\mathcal{X}''$ is not of the form (7.2).

2. *Constant difference 0.5.* When $x_4 = x_3 \pm 0.5$, we can actually find more than one set of 4 points that map to a subset of the given differences. In the case $x_4 = x_3 + 0.5$, the differences are $\mathcal{D} = \pm\{0, 1, x_3, x_3 + 0.5, 1 - x_3, 0.5 - x_3, 0.5\}$; thus, $\mathcal{D}$ contains all pairwise difference from both $\mathcal{X}' = \{0, 1, x_3, x_3 + 0.5\}$ and $\mathcal{X}'' = \{0, 1, 0.5, x_3\}$. However, $\mathcal{X}''$ does not lead to a zero ℓ^2 error.

3. *Collision of differences when adding a new point to the solution set.* This is for example the case of $x_4 = 1-2x_3$ with $\mathcal{D} = \pm\{0, 1, x_3, 1-2x_3, 1-x_3, 2x_3, 1-3x_3\}$. The differences 0 and 1 are always selected in the first and the second step. In the third step, we could potentially add $2x_3$ to $\mathcal{X}_2 = \{0, 1\}$ and reduce the set of differences to $\mathcal{D} = \pm\{x_3, 1 - x_3, 1 - 3x_3\}$. Next, we select x_3 as a new point. We can verify that the differences of x_3 and the values in $\mathcal{X}_3 = \{0, 1, 2x_3\}$ exist in $\mathcal{D}$. However, in this verification we use the value x_3 in $\mathcal{D}$ twice: once as the difference between x_3 and 0, and once as the difference between x_3 and $2x_3$. The set of pairwise differences of $\mathcal{X}_4 = \{0, 1, 2x_3, x_3\}$ is indeed contained in the original $\mathcal{D}$, but neither its ℓ^2 error nor its index error is zero. Notice that if we swap the third and the fourth step, this confusion would be avoided as x_3 would be removed from the set of differences in the third step.

These three cases explain all the segments visible in Fig. 7.11. Such an analysis also applies to noisy regimes; the main difference is that we move from very localized configurations to blurrier areas where the solution is ambiguous. In fact, we introduced some noise into the experiment in Fig. 7.11 to enable the visualization of the *lines* identifying challenging patterns—a noiseless setting would have just led to infinitesimally thin lines. Such patterns become blurrier and wider as noise increases.

7.5.8 Conclusion

We presented a novel approach to solve the phase retrieval problem for sparse signals. While conventional algorithms operate in discretized space and recover the support of the points on a grid, the power of FRI sampling combined with the sparsity assumption on the signal model enables to recover the support of the points in continuous space. In this chapter, we focused on the support recovery algorithm which is in fact equivalent to labeling the entries of CDMs. We observed that while our algorithm runs in polynomial time with respect to the sparsity number of the signal, it remains relatively costly. To alleviate the computational costs without impacting the quality of the reconstruction, we proposed a caching layer to avoid repeating calculations. Furthermore, supported by the theoretical insights and algorithmic results on CDMs from Chapter 6, we introduced several improvements that contribute to enhance the quality of estimation in the presence of noise. In a more comprehensive coverage of this topic [93], we also derive the performance bound of the support recovery algorithm with a theoretical analysis and demonstrate the benefits of the proposed PR method for sparse signals via a comparison with Charge Flipping, a reference algorithm in crystallography.

Conclusion

> Il faut que je supporte deux ou trois chenilles si je veux connaître les papillons.[1]
>
> *The Little Prince*
> Antoine de Saint-Exupery

For everyone who read the previous chapters of this thesis, we hope that our contributions and discussions gratified your curiosity.

If you are an academic or entrepreneur interested in the field of simultaneous localization and mapping, we hope that our practical tools, algorithms and favorable results from Chapter 3 and Chapter 4 will encourage you to pursue your research using sonar robots with minimal sensing, and create even faster and more versatile methods. And when you do so, do not forget our theoretical results from Chapter 2 and Chapter 5 to be sure that your solution is unique.

If your interest happens to be in ranking, sensor localization or phase retrieval, we hope that our insights from Chapter 6 and Chapter 7 about the connection of these seemingly very different problems will help you find answers to your questions in the union of these distinct fields; each of them has attracted a vast literature, some of which focuses on theoretical aspects, and some of which addresses practical concerns.

Outlook

For adventurers looking for inspiration, below we propose a few research topics to be further investigated.

Active learning from pairwise comparisons In Chapter 7 we proposed an algorithm to estimate ranking of sport teams from their pairwise comparisons, which are given as a set of net scores in a finished tournament. What if every game has a cost, and we would like to find a "correct" ranking of the teams in the most efficient way? This problem is known as optimal data collection problem, or more specifically, optimal tournament design. In practice, there exist many different

[1] Well, I must endure the presence of a few caterpillars if I wish to become acquainted with butterflies.

kinds of tournaments. In a single elimination tournament with N teams and $N-1$ games, we expect that the "best" team wins the tournament, but it is difficult to rank the remaining teams in any reasonable way. At the other extreme, a round-robin tournament among N teams requires $\binom{N}{2}$ games, which may not be feasible when N is large. It is clear that there is a trade-off between the amount of pairwise data collected and the informativeness of the ranking, and the question is how to find an optimal tournament which maximizes the probability of a correct ranking for a given number of games?

In [133], the authors considered the ranking problem as the least squares estimation equal to our vector form (6.8) for CDMs. They formulated a bi-level optimization problem to maximally improve the informativeness of the ranking given the opportunity to collect a certain number of additional pairwise comparisons. In this bi-level optimization, the inner problem is to determine the ranking for a given set of scores and the outer problem is to identify data that maximizes the Fisher information of the ranking. They showed that for the least-squares estimate, the outer problem decouples from the inner problem. In other words, the optimality criteria for the Fisher information does not depend on the scores of the games.

Our preliminary results show that by maximizing the expected probability of a correct ranking rather than the Fisher information, the two problems do not decouple and the probability of estimating the correct ranking is higher than for the methods proposed in [133]. Even though theoretically efficient and promising, the tournaments created by our optimization algorithm are not very realistic: teams with similar rankings have to play against each other time and again, while the teams that stand further in the ranking never face each other. Hence, it would be interesting to see how adding the cost and constraints on the data collection, i.e. limiting the number of games between the same pairs, impacts different optimization strategies.

Optimal control of a mobile robot A similar question appears in the problem of simultaneous localization and mapping. In Chapter 4, we showed that when a robot performs a random motion, we can jointly reconstruct a room and estimate its trajectory by measuring distances from the nearest walls. For some applications it could however be crucial to localize the robot and obtain the geometry of the room in the most efficient way. In such a case, after every measurement the robot should decide where to move next to maximize the informativeness of the future measurements.

In [176], the authors were inspired by a similar question and addressed the problem of optimal robot motion for the audio source localization. They considered a mobile robot with a microphone array and one fixed source. They represented the likelihood of the source position on a discrete grid and proposed an algorithm to find the optimal robot trajectory by minimizing the entropy of the grid. Their experiments showed that optimal trajectories in different environments have similar shapes: the robots move towards the source along an arc, pass by the source and move around it.

To develop further these promising results, we suggest to consider the following questions: How does the performance of the algorithm change when the motion model is noisy? Can we move from the discrete grid to the continuous domain? What if the robot is equipped with a single omnidirectional microphone instead of the microphone array and we replace the angle of arrival measurements with point-to-plane distances? What if there are multiple sources and their labeling is unknown? We believe that answering these questions will enable to design an optimal control system for our echolocating robot, which would be of considerable value in time-sensitive applications, such as fire in buildings.

Odometry-free EchoSLAM In Chapter 3, we proposed an iterative algorithm to factor a point-to-plane distance matrix (PPDM) into a product of coordinate and projection matrices. In terms of simultaneous localization and mapping, it allows to jointly estimate a robot's trajectory and positions of the walls from the propagation times of first-order echoes. To obtain first-order echoes, we needed to distinguish them from other pulses in room impulse responses and to match them with the walls from which they originated. To that end, we leveraged the estimates of trajectories. We showed that such an approach is robust to noise in the estimates, but it does not work when the information about the trajectory is completely absent.

As the factorization of PPDMs does not require any knowledge about the trajectory (and walls), we would ideally have an algorithm with the same characteristics for the detection and labeling of the first-order echoes. In terms of PPDMs, this problem arises in the following question: Given a set of unordered entries from every row of a PPDM, along with some additional measurements, can we efficiently reconstruct a PPDM?

A positive answer would enable us to envision an odometry-free EchoSLAM algorithm and apply it not only to echolocating robots with inertial measurement units or odometry hardware, but also to any mobile device with a minimal sensing (one source and one receiver) and no information about its motion. As the increasingly present smart-home voice assistants constitute one category of such devices, the odometry-free EchoSLAM could have significant potential for variety of applications.

Closing remark One last time, I will refer to "The Little Prince", and conclude the thesis with the words that resonate with all of us:

> *I am not at all sure of success. One drawing goes along all right, and another has no resemblance to its subject. I make some errors, too, in the little prince's height: in one place he is too tall and in another too short. And I feel some doubts about the colour of his costume. So I fumble along as best I can, now good, now bad, and I hope generally fair-to-middling.*
>
> *The Little Prince*
> Antoine de Saint-Exupery

I slightly modified the paragraph to better resonate with mathematicians and other scientists:

> *I am not at all sure of success. One* derivation *goes along all right, and another has no resemblance to* its experimental result. *I make some errors, too, in the* noise model: *in one place* it *is too* large *and in another too* small. *And I feel some doubts about the* algorithm convergence. *So I fumble along as best I can, now good, now bad, and I hope generally fair-to-middling.*

While doing the research and writing this thesis, I often identified with this quote. Many times it made me realize that it is natural to be critical towards one's own work. But more importantly, it taught me to always keep persevering; accept the imperfections and hold the head up. Each individual contribution, no matter the size, is a significant and valuable piece in creating a greater common knowledge. I hope that this thesis will be a stepping stone for someone else just getting started.

Bibliography

[1] Y. Ahaggach, "Modular mobile robot for localization experiments," EPFL, Lausanne, Switzerland, Tech. Rep., 2019.

[2] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, "Solving Euclidean distance matrix completion problems via semidefinite programming," *Computational optimization and applications*, vol. 12, no. 1-3, pp. 13–30, 1999.

[3] B. Alipanahi, "New approaches to protein NMR automation," Ph.D. dissertation, University of Waterloo, 2011.

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

[5] F. Antonacci, J. Filos, M. R. Thomas, E. A. Habets, A. Sarti, P. A. Naylor, and S. Tubaro, "Inference of room geometry from acoustic impulse responses," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2683–2695, 2012.

[6] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[7] J. Azcarreta Ortiz, "Pyramic array: An FPGA based platform for many-channel audio acquisition," Master's thesis, EPFL, Lausanne, Switzerland and Universitat Politècnica de Catalunya, 2016.

[8] S. Bancroft, "An algebraic solution of the GPS equations," *IEEE Transactions on Aerospace and Electronic Systems*, no. 1, pp. 56–59, 1985.

[9] E. Barankin, "Locally best unbiased estimates," *The Annals of Mathematical Statistics*, pp. 477–501, 1949.

[10] Y. Barbotin, "Parametric estimation of sparse channels: theory and applications," Ph.D. dissertation, EPFL, Lausanne, 2014.

[11] Y. Barbotin and M. Vetterli, "Fast and robust parametric estimation of jointly sparse channels," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 402–412, 2012.

[12] R. Beinert and G. Plonka, "Ambiguities in one-dimensional phase retrieval of structured functions," *Proceedings in Applied Mathematics and Mechanics*, vol. 15, no. 1, pp. 653–654, 2015.

[13] ——, "Sparse phase retrieval of one-dimensional signals by Prony's method," *Frontiers in Applied Mathematics and Statistics*, vol. 3, p. 5, 2017.

[14] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*. Springer Science & Business Media, 2008, vol. 1.

[15] J. C. Bezdek and R. J. Hathaway, "Some notes on alternating optimization," in Proc. *AFSS International Conference on Fuzzy Systems*, pp. 288–300. Springer, 2002.

[16] J. C. Bezdek, R. J. Hathaway, R. E. Howard, C. A. Wilson, and M. P. Windham, "Local convergence analysis of a grouped variable version of coordinate descent," *Journal of Optimization Theory and Applications*, vol. 54, no. 3, pp. 471–477, 1987.

[17] S. J. 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," *Annals of Operations Research*, vol. 271, no. 1, pp. 161–203, 2018.

[18] S. T. Birchfield and A. Subramanya, "Microphone array position calibration by basis-point classical multidimensional scaling," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1025–1034, 2005.

[19] P. Biswas, "Semidefinite programming approaches to distance geometry problems," Ph.D. dissertation, Stanford University, 2007.

[20] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in Proc. *3rd International Symposium on Information Processing in Sensor Networks*, pp. 46–54. ACM, 2004.

[21] R. Biswas and S. Thrun, "A passive approach to sensor network localization," in Proc. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 1544–1549. IEEE, 2004.

[22] T. Blu, P. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot, "Sparse sampling of signal innovations," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 31–40, 2008.

[23] D. A. Bohn, "Environmental effects on the speed of sound," *Journal of the Audio Engineering Society*, vol. 36, no. 4, 1988.

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

[25] O. Bottema and B. Roth, *Theoretical kinematics*. Courier Corporation, 1990, vol. 24.

[26] M. Boutin and G. Kemper, "A drone can hear the shape of a room," *arXiv preprint arXiv:1901.10472*, 2019.

[27] W. Braun, C. Bösch, L. R. Brown, N. Gō, and K. Wüthrich, "Combined use of proton-proton Overhauser enhancements and a distance geometry algorithm for determination of polypeptide conformations. Application to micelle-bound glucagon," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 667, no. 2, pp. 377 – 396, 1981.

[28] W. Braun and N. Gō, "Calculation of protein conformations by proton-proton distance constraints: A new efficient algorithm," *Journal of Molecular Biology*, vol. 186, no. 3, pp. 611 – 626, 1985.

[29] J. Bruck, J. Gao, and A. A. Jiang, "Localization and routing in sensor networks by local angle information," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, p. 7, 2009.

[30] C. Brunner, T. Peynot, and T. Vidal-Calleja, "Combining multiple sensor modalities for a localisation robust to smoke," in Proc. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2489–2496, 2011.

[31] A. M. Buchanan and A. W. Fitzgibbon, "Damped newton algorithms for matrix factorization with missing data," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 316–322, 2005.

[32] F. Buekenhout and M. Parker, "The number of nets of the regular convex polytopes in dimension ≤ 4," *Discrete Mathematics*, vol. 186, no. 1-3, pp. 69–94, 1998.

[33] T. Callaghan, P. J. Mucha, and M. A. Porter, "Random walker ranking for NCAA division I-A football," *The American Mathematical Monthly*, vol. 114, no. 9, pp. 761–777, 2007.

[34] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.

[35] M. Cieliebak, S. Eidenbenz, and P. Penna, "Partial digest is hard to solve for erroneous input data," *Theoretical Computer Science*, vol. 349, no. 3, pp. 361–381, 2005.

[36] W. N. Colley, "Colley's bias free college football ranking method: The Colley matrix explained," *Princeton University, Princeton*, 2002.

[37] H. Cramér, *Mathematical methods of statistics*. Princeton university press, 1999, vol. 9.

[38] M. Crocco, A. Del Bue, and V. Murino, "A bilinear approach to the position self-calibration of multiple sensors," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 660–673, 2011.

[39] M. Crocco, A. Trucco, V. Murino, and A. Del Bue, "Towards fully uncalibrated room reconstruction with sound," in Proc. *22nd European Signal Processing Conference (EUSIPCO)*, pp. 910–914. IEEE, 2014.

[40] G. A. Croes, "A method for solving traveling-salesman problems," *Operations research*, vol. 6, no. 6, pp. 791–812, 1958.

[41] T. Dakić, *On the Turnpike Problem*. Simon Fraser University BC, Canada, 2000.

[42] J. De Leeuw, "Applications of convex analysis to multidimensional scaling," *Department of Statistics, UCLA*, 2005.

[43] L. Doherty, L. El Ghaoui, and K. Pister, "Convex position estimation in wireless sensor networks," in Proc. *IEEE Conference on Computer Communications. 20th Annual Joint Conference of the IEEE Computer and Communications Society*, vol. 3, pp. 1655–1663, 2001.

[44] I. Dokmanić, "Listening to distances and hearing shapes," Ph.D. dissertation, Ecole Polytechnique Federale de Lausanne (EPFL), 2015.

[45] I. Dokmanić, L. Daudet, and M. Vetterli, "How to localize ten microphones in one finger snap," in Proc. *22nd European Signal Processing Conference (EUSIPCO)*, pp. 2275–2279, 2014.

[46] ——, "From acoustic room reconstruction to SLAM," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6345–6349, 2016.

[47] I. Dokmanić, Y. M. Lu, and M. Vetterli, "Can one hear the shape of a room: The 2-D polygonal case," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 321–324. IEEE, 2011.

[48] I. Dokmanić, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean distance matrices: Essential theory, algorithms, and applications," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 12–30, 2015.

[49] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape," *Proceedings of the National Academy of Sciences*, vol. 110, no. 30, pp. 12 186–12 191, 2013.

[50] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, p. 3, 2009.

[51] P. L. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1741–1757, 2007.

[52] P. L. Dragotti and F. Homann, "Sampling signals with finite rate of innovation in the presence of noise," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2941–2944. IEEE, 2009.

[53] T. Du, S. Qu, Q. Guo, and L. Zhu, "A simple efficient anchor-free node localization algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 13, no. 4, p. 1550147717705784, 2017.

[54] N. Duric, P. Littrup, L. Poulo, A. Babkin, R. Pevzner, E. Holsapple, O. Rama, and C. Glide, "Detection of breast cancer with ultrasound tomography: First results with the computed ultrasound risk evaluation (CURE) prototype," *Medical Physics*, vol. 34, no. 2, pp. 773–785, 2007.

[55] J. Edwards, "Signal processing improves autonomous vehicle navigation accuracy: Guidance innovations promise safer and more reliable autonomous vehicle operation [special reports]," *IEEE Signal Processing Magazine*, vol. 36, no. 2, pp. 15–18, 2019.

[56] Y. C. Eldar, P. Sidorenko, D. G. Mixon, S. Barel, and O. Cohen, "Sparse phase retrieval from short-time Fourier measurements," *IEEE Signal Processing Letters*, vol. 22, no. 5, pp. 638–642, 2015.

[57] C. Evers, A. H. Moore, and P. A. Naylor, "Acoustic simultaneous localization and mapping (a-SLAM) of a moving microphone array and its surrounding speakers," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6–10. IEEE, 2016.

[58] C. Evers and P. A. Naylor, "Acoustic SLAM," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1484–1498, 2018.

[59] 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," *Journal of the American Chemical Society*, vol. 119, no. 29, pp. 6853–6857, 1997.

[60] B. P. Flanagan and K. L. Bell, "Array self-calibration with large sensor position errors," *Signal Processing*, vol. 81, no. 10, pp. 2201–2214, 2001.

[61] N. Gaffke and R. Mathar, "A cyclic projection algorithm via duality," *Metrika*, vol. 36, no. 1, pp. 29–54, 1989.

[62] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 4, pp. 692–730, 2017.

[63] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens, "Auto-localization in ad-hoc microphone arrays," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 106–110, 2013.

[64] S. Geršgorine, "Uber die Abgrenzung der Eigenwerte einer Matrix," *lzv. Akad. Nauk. USSR. Otd. Fiz-Mat. Nauk*, vol. 7, pp. 749–754, 1931.

[65] W. Givens, "Computation of plain unitary rotations transforming a general matrix to triangular form," *Journal of the Society for Industrial and Applied Mathematics*, vol. 6, no. 1, pp. 26–50, 1958.

[66] W. Glunt, T. L. Hayden, and M. Raydan, "Molecular conformations from distance matrices," *Journal of Computational Chemistry*, vol. 14, no. 1, pp. 114–120, 1993.

[67] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: a survey and extensions," *Mathematical Methods of Operations Research*, vol. 66, no. 3, pp. 373–407, 2007.

[68] A. Y. Govan, A. N. Langville, and C. D. Meyer, "Offense-defense approach to ranking team sports," *Journal of Quantitative Analysis in Sports*, vol. 5, no. 1, 2009.

[69] P. Güntert, C. Mumenthaler, and K. Wüthrich, "Torsion angle dynamics for NMR structure calculation with the new program Dyana," *Journal of Molecular Biology*, vol. 273, no. 1, pp. 283–298, 1997.

[70] P. Güntert, *Automated NMR Structure Calculation With CYANA*. Totowa, NJ: Humana Press, 2004, pp. 353–378.

[71] L. Guttman, "A general nonmetric technique for finding the smallest coordinate space for a configuration of points," *Psychometrika*, vol. 33, no. 4, pp. 469–506, 1968.

[72] K. M. Hall, "An r-dimensional quadratic placement algorithm," *Management Science*, vol. 17, no. 3, pp. 219–229, 1970.

[73] G. W. Hart, *Multidimensional Analysis - Algebras and Systems for Science and Engineering*, 1st ed. Springer-Verlag New York, Inc., 1995.

[74] R. Hartley and F. Schaffalitzky, "Powerfactorization: 3D reconstruction with missing or uncertain data," *Australia-Japan Advanced Workshop on Computer Vision*, vol. 74, pp. 76–85, 2003.

[75] T. F. Havel and K. Wüthrich, "An evaluation of the combined use of nuclear magnetic resonance and distance geometry for the determination of protein conformations in solution," in *NMR In Structural Biology: A Collection of Papers by Kurt Wüthrich.* World Scientific, 1995, pp. 305–318.

[76] Y. Hioka, M. Kingan, G. Schmid, and K. A. Stol, "Speech enhancement using a microphone array mounted on an unmanned aerial vehicle," in Proc. *International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 1–5. IEEE, 2016.

[77] J.-S. Hu, C.-Y. Chan, C.-K. Wang, M.-T. Lee, and C.-Y. Kuo, "Simultaneous localization of a mobile robot and multiple sound sources using a microphone array," *Advanced Robotics*, vol. 25, no. 1-2, pp. 135–152, 2011.

[78] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Application-specific sensor network for environmental monitoring," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, p. 17, 2010.

[79] D. Jacobs, "Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 206–212, 1997.

[80] K. Jaganathan, S. Oymak, and B. Hassibi, "Sparse phase retrieval: Uniqueness guarantees and recovery algorithms," *IEEE Transactions on Signal Processing*, vol. 65, no. 9, pp. 2402–2410, 2017.

[81] I. Jager, R. Heusdens, and N. D. Gaubitch, "Room geometry estimation from acoustic echoes using graph-based echo labeling," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2016.

[82] T. Jech, "The ranking of incomplete tournaments: A mathematician's guide to popular sports," *The American Mathematical Monthly*, vol. 90, no. 4, pp. 246–266, 1983.

[83] C. R. Johnson, "Inverse M-matrices," *Linear Algebra and its Applications*, vol. 47, pp. 195–216, 1982.

[84] J. P. Keener, "The Perron-Frobenius theorem and the ranking of football teams," *SIAM review*, vol. 35, no. 1, pp. 80–93, 1993.

[85] J. Kietlinski-Zaleski, "Ultra-wideband positioning using reflections from known indoor features," Ph.D. dissertation, University of Pisa, 2011.

[86] J. Kietlinski-Zaleski and T. Yamazato, "TDoA UWB positioning with three receivers using known indoor features," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 94, no. 3, pp. 964–971, 2011.

[87] H. Kim and M. Viberg, "Two decades of array signal processing research," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[88] K. T. Knox, "Image retrieval from astronomical speckle patterns," *Journal of the Optical Society of America.*, vol. 66, no. 11, pp. 1236–1239, 1976.

[89] K. Kolev, P. Tanskanen, P. Speciale, and M. Pollefeys, "Turning mobile phones into 3D scanners," in Proc. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3946–3953, 2014.

[90] M. Kreković, G. Baechler, I. Dokmanić, and M. Vetterli, "Structure from sound with incomplete data," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3539–3543. IEEE, 2018.

[91] M. Kreković, G. Baechler, F. Dümbgen, G. Elhami, R. Scheibler, A. J. Scholefield, and M. Vetterli, "Combining range and direction for improved localization," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3484–3488. IEEE, 2018.

[92] M. Kreković, G. Baechler, F. Dümbgen, G. Elhami, and M. Vetterli, "Coordinate difference matrices," *SIAM Journal on Matrix Analysis and Applications*, 2018, manuscript under review.

[93] M. Kreković, G. Baechler, J. Ranieri, A. Chebira, Y. M. Lu, and M. Vetterli, "Super resolution phase retrieval for sparse signals," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4839–4854, 2019.

[94] M. Kreković, I. Dokmanić, and M. Vetterli, "EchoSLAM: Simultaneous localization and mapping with acoustic echoes," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11–15. IEEE, 2016.

[95] ——, "Look, no beacons! Optimal all-in-one EchoSLAM," in Proc. *Asilomar Conference on Signals, Systems, and Computers*, 2016.

[96] ——, "Omnidirectional bats, point-to-plane distances, and the price of uniqueness," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3261–3265. IEEE, 2017.

[97] ——, "Shapes from echoes: Uniqueness from point-to-plane distance matrices," *IEEE Transactions on Signal Processing*, 2019, manuscript under review.

[98] M. Kreković, M. Kolundžija, I. Dokmanić, and M. Vetterli, "Point-to-plane distance matrices in practice: Application to room reconstruction," *IEEE Transactions on Signal Processing*, 2019, manuscript to be sumibtted.

[99] N. Krislock and H. Wolkowicz, "Euclidean distance matrices and applications," in *Handbook on semidefinite, conic and polynomial optimization.* Springer, 2012, pp. 879–914.

[100] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[101] Y. Kuang, E. Ask, S. Burgess, and K. Åström, "Understanding TOA and TDOA network calibration using far field approximation as initial estimate," *International Conference on Pattern Recognition Applications and Methods*, pp. 590–596, 2012.

[102] Y. Kuang, K. Åström, and F. Tufvesson, "Single antenna anchor-free UWB positioning based on multipath propagation," in Proc. *International Conference on Communications (ICC)*, pp. 5814–5818. IEEE, 2013.

[103] J. Kubatko, D. Oliver, K. Pelton, and D. T. Rosenbaum, "A starting point for analyzing basketball statistics," *Journal of Quantitative Analysis in Sports*, vol. 3, no. 3, 2007.

[104] V. Kubelka, L. Oswald, F. Pomerleau, F. Colas, T. Svoboda, and M. Reinstein, "Robust data fusion of multimodal sensory information for mobile robots," *Journal of Field Robotics*, vol. 32, no. 4, pp. 447–473, 2015.

[105] P. Kułakowski, J. Vales-Alonso, E. Egea-López, W. Ludwin, and J. García-Haro, "Angle-of-arrival localization based on antenna arrays for wireless sensor networks," *Computers & Electrical Engineering*, vol. 36, no. 6, pp. 1181–1186, 2010.

[106] A. N. Langville and C. D. Meyer, *Who's #1?: The science of rating and ranking.* Princeton University Press, 2012.

[107] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson, and M. Z. Win, "A belief propagation algorithm for multipath-based SLAM," *Corr*, vol. arXiv:1801.04463v3, 2018.

[108] E. Leitinger, S. Grebien, X. Li, F. Tufvesson, and K. Witrisal, "On the use of MPC amplitude information in radio signal based SLAM," in Proc. *Statistical Signal Processing Workshop (SSP)*, pp. 633–637. IEEE, 2018.

[109] E. Leitinger, P. Meissner, M. Lafer, and K. Witrisal, "Simultaneous localization and mapping using multipath channel information," in Proc. *IEEE International Conference on Communication Workshop*, pp. 754–760, 2015.

[110] E. Leitinger, P. Meissner, C. Rüdisser, G. Dumphart, and K. Witrisal, "Evaluation of position-related information in multipath components for indoor positioning," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2313–2328, 2015.

[111] E. Leitinger, F. Meyer, P. Meissner, K. Witrisal, and F. Hlawatsch, "Belief propagation based joint probabilistic data association for multipath-assisted indoor navigation and tracking," in Proc. *International Conference on Localization and GNSS (ICL-GNSS)*, pp. 1–6. IEEE, 2016.

[112] E. Leitinger, F. Meyer, F. Tufvesson, and K. Witrisal, "Factor graph based simultaneous localization and mapping using multipath channel information," in Proc. *International Conference on Communications Workshops (ICC Workshops)*, pp. 652–658. IEEE, 2017.

[113] X. Li and V. Voroninski, "Sparse signal recovery from quadratic measurements via convex programming," *SIAM Journal on Mathematical Analysis*, vol. 45, no. 5, pp. 3019–3033, 2013.

[114] Y. E. Li and L. Demanet, "Phase and amplitude tracking for seismic event separation," *Geophysics*, vol. 80, no. 6, pp. WD59–WD72, 2015.

[115] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, "Euclidean distance geometry and applications," *SIAM Review*, vol. 56, no. 1, pp. 3–69, 2014.

[116] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 6, pp. 1067 –1080, 2007.

[117] D. Macagnano and G. T. F. De Abreu, "Algebraic approach for robust localization with heterogeneous information," *IEEE Transactions on Wireless Communications*, vol. 12, no. 10, pp. 5334–5345, 2013.

[118] D. Malioutov, M. Cetin, and A. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[119] I. Maravić and M. Vetterli, "Exact sampling results for some classes of parametric nonbandlimited 2-D signals," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 175–189, 2004.

[120] P. Marziliano and M. Vetterli, "Reconstruction of irregularly sampled discrete-time bandlimited signals with unknown sampling locations," *IEEE Transactions on Signal Processing*, vol. 48, no. 12, pp. 3462–3471, 2000.

[121] K. Massey, "Statistical models applied to the rating of sports teams," *Bluefield College*, 1997.

[122] R. Mautz, "Indoor positioning technologies," Ph.D. dissertation, ETH Zurich, Zurich, 2012.

[123] P. Meissner, "Multipath-assisted indoor positioning," Ph.D. dissertation, Graz University of Technology, 2014.

[124] R. Mendrzik, H. Wymeersch, G. Bauch, and Z. Abu-Shaban, "Harnessing NLOS components for position and orientation estimation in 5G millimeter wave MIMO," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 93–107, 2018.

[125] M. Milford, A. Jacobson, Z. Chen, and G. Wyeth, "RatSLAM: Using models of rodent hippocampus for robot navigation and beyond," in *Robotics Research*. Springer, 2016, pp. 467–485.

[126] R. P. Millane, "Phase retrieval in crystallography and optics," *Journal of the Optical Society of America.*, vol. 7, no. 3, pp. 394–411, 1990.

[127] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 593–598, 2002.

[128] A. H. Moore, M. Brookes, and P. A. Naylor, "Room geometry estimation from a single channel acoustic impulse response," in Proc. *21st European Signal Processing Conference (EUSIPCO)*, pp. 1–5. IEEE, 2013.

[129] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, *Distance geometry: theory, methods, and applications*. Springer Science & Business Media, 2012.

[130] K. P. Murphy, "Bayesian map learning in dynamic environments," in Proc. *Advances in Neural Information Processing Systems*, pp. 1015–1021, 2000.

[131] P. Nicolas and G. Vezzosi, "Localization of far-field sources with an array of unknown geometry," in *Underwater Acoustic Data Processing*. Springer, 1989, pp. 503–509.

[132] N. Ono, H. Kohno, N. Ito, and S. Sagayama, "Blind alignment of asynchronously recorded signals for distributed microphone array," in Proc. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 161–164, 2009.

[133] B. Osting, C. Brune, and S. J. Osher, "Optimal data collection for informative rankings expose well-connected graphs," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2981–3012, 2014.

[134] M. Pacholska, B. B. Haro, A. Scholefield, and M. Vetterli, "Sampling at unknown locations, with an application in surface retrieval," *IEEE International Conference on Sampling Theory and Applications*, pp. 364–368, 2017.

[135] H. Pan, T. Blu, and M. Vetterli, "Towards generalized FRI sampling with an application to source resolution in radioastronomy," *IEEE Transactions on Signal Processing*, vol. 65, no. 4, pp. 821–835, 2017.

[136] ——, "Efficient multidimensional diracs estimation with linear sample complexity," *IEEE Transactions on Signal Processing*, vol. 66, no. 17, pp. 4642–4656, 2018.

[137] R. Parhizkar, "Euclidean distance matrices: Properties, algorithms and applications," Ph.D. dissertation, Ecole Polytechnique Federale de Lausanne (EPFL), 2013.

[138] J. Parsons, J. B. Holmes, J. M. Rojas, J. Tsai, and C. E. M. Strauss, "Practical conversion from torsion space to Cartesian space for in silico protein synthesis," *Journal of computational chemistry*, vol. 26, 2005.

[139] S. Patole and M. Torlak, "Two dimensional array imaging with beam steered data," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5181–5189, 2013.

[140] F. Peng, T. Wang, and B. Chen, "Room shape reconstruction with a single mobile acoustic sensor," in Proc. *IEEE Global Conference on Signal and Information Processing*, pp. 1116–1120, 2015.

[141] R. Peng and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks," *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 1, pp. 374–382, 2006.

[142] G. Poole and T. Boullion, "A survey on M-matrices," *SIAM Review*, vol. 16, no. 4, pp. 419–427, 1974.

[143] S. Pradhan, G. Baig, W. Mao, L. Qiu, G. Chen, and B. Yang, "Smartphone-based acoustic indoor space mapping," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, p. 75, 2018.

[144] T. Rajapaksha, X. Qiu, E. Cheng, and I. Burnett, "Geometrical room geometry estimation from room impulse responses," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 331–335. IEEE, 2016.

[145] J. Ranieri, A. Chebira, Y. M. Lu, and M. Vetterli, "Phase retrieval for sparse signals: Uniqueness conditions," *arXiv preprint arXiv:1308.3058*, 2013.

[146] L. Remaggi, P. J. Jackson, W. Wang, and J. A. Chambers, "A 3d model for room boundary estimation," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 514–518. IEEE, 2015.

[147] F. Ribeiro, D. Florencio, D. Ba, and C. Zhang, "Geometrically constrained room modeling with compact microphone arrays," *Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1449–1460, 2011.

[148] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003, vol. 82.

[149] M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual SLAM and structure from motion in dynamic environments: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, p. 37, 2018.

[150] R. Scheibler, J. Azcarreta, R. Beuchat, and C. Ferry, "Pyramic: Full stack open microphone array architecture and dataset," in Proc. *16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 226–230, 2018.

[151] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 351–355. IEEE, 2018.

[152] T. Schlick, *Molecular modeling and simulation: an interdisciplinary guide*. Springer Science & Business Media, 2010, vol. 21.

[153] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *Adaptive Antennas for Wireless Communications*, no. 3, pp. 190–194, 2009.

[154] P. H. Schönemann, "On metric multidimensional unfolding," *Psychometrika*, vol. 35, no. 3, pp. 349–366, 1970.

[155] G. Schouten and J. Steckel, "Principles of biological echolocation applied to radar sensing: Applying biomimetic sensors to achieve autonomous navigation," *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 98–111, 2019.

[156] A. Shahmansoori, G. E. Garcia, G. Destino, G. Seco-Granados, and H. Wymeersch, "Position and orientation estimation through millimeter-wave MIMO in 5G systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1822–1835, 2017.

[157] Y. Shechtman, A. Beck, and Y. C. Eldar, "GESPAR: efficient phase retrieval of sparse signals," *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 928–938, 2014.

[158] O. Shih and A. Rowe, "Can a phone hear the shape of a room?" in Proc. *18th International Conference on Information Processing in Sensor Networks*, pp. 277–288. ACM, 2019.

[159] S. Shiraishi, T. Obata, and M. Daigo, "Properties of a positive reciprocal matrix and their application to AHP," *Journal of the Operations Research Society of Japan*, vol. 41, no. 3, pp. 404–414, 1998.

[160] S. S. Skiena, W. D. Smith, and P. Lemke, "Reconstructing sets from interpoint distances," in Proc. *6th Annual Symposium on Computational Geometry*, pp. 332–339. ACM, 1990.

[161] D. A. Spielman and S.-H. Teng, "Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 3, pp. 835–885, 2014.

[162] J. Steckel and H. Peremans, "BatSLAM: Simultaneous localization and mapping using biomimetic sonar," *PloS one*, vol. 8, no. 1, p. e54076, 2013.

[163] C. H. Suh and C. W. Radcliffe, *Kinematics and mechanisms design*. Wiley, 1987.

[164] Y. Takane, F. W. Young, and J. De Leeuw, "Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features," *Psychometrika*, vol. 42, no. 1, pp. 7–67, 1977.

[165] A. F. Tchango, "Behavioral recognition and multi-target tracking in partially observed environments," Ph.D. dissertation, Université de Lorraine, 2015.

[166] S. Tervo and T. Tossavainen, "3d room geometry estimation from measured impulse responses," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 513–516. IEEE, 2012.

[167] S. Thrun, "Affine structure from sound," *Advances in Neural Information Processing Systems*, pp. 1353–1360, 2006.

[168] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[169] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[170] L. Torresani, A. Hertzmann, and C. Bregler, "Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 878–892, 2008.

[171] J.-M. Valin, J. Rouat, and F. Michaud, "Enhanced robot audition based on microphone array source separation with post-filter," in Proc. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2123–2128. IEEE, 2004.

[172] B. van de Rotten and S. V. Lunel, "A limited memory Broyden method to solve high-dimensional systems of nonlinear equations," in *EQUADIFF 2003*. World Scientific, 2005, pp. 196–201.

[173] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.

[174] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of Signal Processing*. Cambridge University Press, 2014.

[175] R. Vidal and R. Hartley, "Motion segmentation with missing data using powerfactorization and GPCA," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 310–316, 2004.

[176] E. Vincent, A. Sini, and F. Charpillet, "Audio source localization by optimal control of a mobile robot," in Proc. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5630–5634. IEEE, 2015.

[177] J. Weissbock, "Forecasting success in the National Hockey League using in-game statistics and textual data," Ph.D. dissertation, University of Ottawa, 2014.

[178] J. Wendeberg, T. Janson, and C. Schindelhauer, "Self-localization based on ambient signals," *Theoretical Computer Science*, vol. 453, pp. 98–109, 2012.

[179] K. Witrisal, P. Meissner, E. Leitinger, Y. Shen, C. Gustafson, F. Tufvesson, K. Haneda, D. Dardari, A. F. Molisch, A. Conti, *et al.*, "High-accuracy localization for assisted living: 5G systems will turn multipath channels from foe to friend," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 59–70, 2016.

[180] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," *European Conference on Computer Vision*, pp. 94–106, 2006.

[181] B. Zhou, M. Elbadry, R. Gao, and F. Ye, "BatMapper: Acoustic sensing based indoor floor plan construction using smartphones," in Proc. *15th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 42–55, 2017.

[182] A. Zimmermann, S. Moorthy, and Z. Shi, "Predicting college basketball match outcomes using machine learning techniques: some results and lessons learned," *arXiv preprint arXiv:1310.3607*, 2013.

Miranda **KREKOVIĆ**

Im Sihlhof 2
8134 Adliswil (ZH), Switzerland
Tel: +41 78 812 75 32
miranda.krekovic@gmail.com

Croatian citizen
Born June 12th 1991

ACADEMIC EDUCATION

2014 – 2019 **Ph.D. in Computer and Communication Sciences**
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
Advisors: Prof. Martin Vetterli and Prof. Ivan Dokmanić

2012 – 2014 **Master of Science in Information and Communication Technology**
Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia
- Major GPA: 5.000 / 5.000, Rank: 1 / 150.
  Graduated Summa Cum Laude (with highest honor)

2009 – 2012 **Bachelor of Science in Computing, Information Processing and Multimedia Systems**
Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia
- Major GPA: 4.977 / 5.000, Rank: 2 / 385

2005 – 2009 **XV Gymnasium**
Jordanovac 8, Zagreb, Croatia

WORK EXPERIENCE

Starting Feb 2020 **Google Inc. – Zürich**, Switzerland
Software Engineer
VideoAds Quality team

Jul 2018 – Oct 2018 **Google Inc. – Zürich**, Switzerland
Software Engineering Intern
Joined the VideoAds Quality team and worked on two projects: automation of an internal service and classification of video campaigns.

Sep 2016 – ongoing **GirlsCoding.org – Lausanne**, Switzerland
Co-founder
I embarked on a project on encouraging girls to study computer science and increase the involvement of women with technical degrees. With my colleague, I co-founded a social enterprise to inspire girls of age 9 to 16 to embrace technologies and learn to code through hands-on workshops.

Sep 2016 – Dec 2016 **Google Inc. – Mountain View, CA**, United States
Software Engineering Intern
Joined the team Sound Understanding (Machine Perception) and worked on a research project for a blind source separation.

Jul 2014 – Aug 2014 **LCAV, EPFL – Lausanne**, Switzerland
Summer Research Intern
Joined Audiovisual Communications Laboratory as a part of Summer@EPFL program and worked on the inverse problems in room acoustics and echolocation.

Feb 2014 – May 2014	**Pinecone – Zagreb**, Croatia *Software Developer* Implemented the back-end of the system (including database design and program architecture) using Java.
Jul 2013 – Aug 2013	**Infobip – Vodnjan**, Croatia *Summer Intern* Worked on several projects (programming in Java, front-end development, Android applications, integration with social networks and Infobip products).
Jul 2012 – Aug 2012	**Ericsson Nikola Tesla d.d. – Zagreb**, Croatia *Summer Intern* Development of a novel system for handling emergency calls (Java, SQL database design, front-end).

RESEARCH EXPERIENCE

Sep 2014 – Sep 2019	**Indoor localization and room reconstruction from acoustic echoes** *Doctoral projects* My PHD work covered my interests in the problems of room reconstruction, source localization, source separation, and simultaneous localization and mapping (SLAM) based on acoustic signals.
Feb 2014 – Jun 2014	**Perspective correction of the text before the optical character recognition process** *MSc thesis* To improve the OCR systems, I proposed a new algorithm for correcting the perspective of the text in the image.
Sep 2013 – Feb 2014	**A system for automating ophthalmologic examination** *Graduate Project* To automate ophthalmologic examinations, we developed a healthcare system designed for general practitioners.
Feb 2013 – Jun 2013	**A system for mapping physical movements into sound synthesis parameters** *Graduate Seminar* We presented an interactive multimedia system intended for mapping gestural features into sound synthesis parameters. The system was employed in a contemporary dance performance Doors of Perception.
Feb 2012 – Jun 2012	**Segmentation of blood vessels and detection of microaneurysms in fundus images** *BSc thesis* I proposed algorithms for the detection of blood vessels and microaneurysms in retinal images.

TEACHING AND SIGNIFICANT COURSE ACHIEVEMENTS (RANKS IN PARENTHESIS)

2017 – 2019	**Organizer and teacher** of several coding workshops and camps for children at age 9-16.
2015 – 2019	**Teacher assistant** in the courses: Statistical Signal and Data Processing through Applications, Audio Signal Processing and Virtual Acoustics, and Signal Processing for Communications.

2014	**Courses**: Neural Networks (1/46), Biomedical Informatics (1/45), Business Intelligence (2/76), Bioinformatics (2/76).
2013	**Teacher assistant** in the course Signals and Systems and Multimedia Systems. **Courses**: Linear Algebra (1/72), Digital Image Processing and Analysis (1/46), Multimedia Communications (1/75), Mobility in Network (1/47), Discrete Mathematics (1/315), Digital Speech Processing (1/16), Digital Signal Processing (1/88), Random Processes in Systems (1/78), Communication Protocols (1/72).
2012	**Courses**: Open Computing (1/83), Multimedia Technologies (1/85), Information Processing (1/40), Information, Logic and Languages (2/88), Information Theory (2/524), Communication Networks (3/331).
2011	**Teacher assistant** in the course Communication Networks, Probability and Statistics and Mathematics 3R. **Courses**: Probability and Statistics (1/594), Introduction to Theoretical Computer Science (2/518), Databases (2/371), Mathematics 3R (3/495), Physics 2 (4/621).
2010	**Teacher assistant** in a course Mathematics 1 and Mathematics 2 and Electronics. **Courses**: Algorithms and Data Structures (2/685), Computer Architecture 1 (3/788), Mathematics 1 (4/869), Mathematics 2 (4/654).
2009	**Teacher assistant** in the course Digital logic.

HONORS, AWARDS AND SCHOLARSHIPS

2018	Finalist of Women Techmakers Scholars Program
2014 – 2015	EPFL EDIC Fellowship
2014	Bronze Plaque ”Josip Lončar”[1] Special Rector’s Award for multimedia project titled The Doors of Perception. Won the 2nd place in the App Start Contest. The contest included 56 teams.
2013	Dean’s Award ”Josip Lončar”[2] Won the 5th place and the best presentation award in the App Start Contest, a student competition in development of mobile and web applications. The contest included 76 teams from 30 different universities.
2012	Special Dean’s Award ”Josip Lončar”[3]
2011 – 2014	The ”City of Zagreb” University Scholarship
2011	Dean’s Award ”Josip Lončar”[2]
2010	Dean’s Award ”Josip Lončar”[2]
2007 – 2011	The ”City of Zagreb” High School Scholarship

PUBLICATIONS

Journal Articles

[11] M. Kreković, I. Dokmanić, and M. Vetterli, “Shapes from echoes: Uniqueness from point-to-plane distance matrices”, *IEEE Transactions on Signal Processing*, 2019, Manuscript under review.

[1]Bronze plaque ”Josip Lončar” is an award for outstanding performance during the entire Master’s program at the University of Zagreb (top 1% of all students).

[2]Dean’s Award ”Josip Lončar” is an award for excellent achievements (top 1% of all students) in the 1st and 2nd year of Bachelors program, and the 1st year of Master’s program at the University of Zagreb.

[3]Special Dean’s Award ”Josip Lončar” is an award for excellent achievements (top 1% of all students) in the entire Bachelors program at the University of Zagreb.

[10] M. Kreković, M. Kolundžija, I. Dokmanić, and M. Vetterli, “Point-to-plane distance matrices in practice: Application to room reconstruction”, *IEEE Transactions on Signal Processing*, 2019, Manuscript to be sumibtted.

[9] M. Kreković, G. Baechler, F. Dümbgen, G. Elhami, and M. Vetterli, “Coordinate difference matrices”, *SIAM Journal on Matrix Analysis and Applications*, 2018, Manuscript under review.

[8] M. Kreković, G. Baechler, J. Ranieri, A. Chebira, Y. M. Lu, and M. Vetterli, “Super resolution phase retrieval for sparse signals”, *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4839–4854, 2019.

Conference Papers

[7] M. Kreković, G. Baechler, F. Dümbgen, G. Elhami, R. Scheibler, A. J. Scholefield, and M. Vetterli, “Combining range and direction for improved localization”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 3484–3488.

[6] M. Kreković, G. Baechler, I. Dokmanić, and M. Vetterli, “Structure from sound with incomplete data”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 3539–3543.

[5] M. Kreković, I. Dokmanić, and M. Vetterli, “Omnidirectional bats, point-to-plane distances, and the price of uniqueness”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, pp. 3261–3265.

[4] ——, “EchoSLAM: Simultaneous localization and mapping with acoustic echoes”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 11–15.

[3] ——, “Look, no beacons! Optimal all-in-one EchoSLAM”, in *Asilomar Conference on Signals, Systems, and Computers*, 2016.

[2] M. Kreković, G. Kreković, and F. Grbac, “Sound my vision: Real-time video analysis on mobile platforms for controlling multimedia performances”, in *International Conference on Sound and Music Computing*, 2015, pp. 235–240.

[1] M. Kreković, I. Dokmanić, and M. Vetterli, “A method for real-time detection of human fall from video”, in *International Convention MIPRO*, 2012, pp. 1709–1712.