

Euclidean Distance Matrices: Properties, Algorithms and Applications

THÈSE N° 5971 (2013)

PRÉSENTÉE LE 7 FÉVRIER 2014

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE COMMUNICATIONS AUDIOVISUELLES
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Reza PARHIZKAR

acceptée sur proposition du jury:

Prof. E. Telatar, président du jury
Prof. M. Vetterli, directeur de thèse
Prof. H. Bölcskei, rapporteur
Dr F. Quick, rapporteur
Prof. R. Urbanke, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2013

Abstract

Euclidean distance matrices (EDMs) are central players in many diverse fields including psychometrics, NMR spectroscopy, machine learning and sensor networks. However, they are not often exploited in signal processing. In this thesis, we analyze attributes of EDMs and derive new key properties of them. These analyses allow us to propose algorithms to approximate EDMs and provide analytic bounds on the performance of our methods. We use these techniques to suggest new solutions for several practical problems in signal processing. Together with these properties, algorithms and applications, EDMs can thus be considered as a fundamental toolbox to be used in signal processing.

In more detail, we start by introducing the structure and properties of EDMs. In particular, we focus on their rank property; the rank of an EDM is at most the dimension of the set of points generating it plus 2. Using this property, we introduce the use of low rank matrix completion methods for approximating and completing noisy and partially revealed EDMs. We apply this algorithm to the problem of sensor position calibration in ultrasound tomography devices. By adapting the matrix completion framework, in addition to proposing a self calibration process for these devices, we also provide analytic bounds for the calibration error.

We then study the problem of sensor localization using distance information by minimizing a non-linear cost function known as the s-stress function in the multidimensional scaling (MDS) community. We derive key properties of this cost function that can be used to reduce the search domain for finding its global minimum. We provide an efficient, low cost and distributed algorithm for minimizing this cost function for incomplete networks and noisy measurements. In randomized experiments, the proposed method converges to the global minimum of the s-stress in more than 99% of the cases. We also address the open problem of existence of non-global minimizers of the s-stress and reduce this problem to a hypothesis. If the hypothesis is true then the cost function has only global minimizers, otherwise, it has non-global minimizers.

Using the rank property of EDMs and the proposed minimization algorithm for approximating them, we address an interesting and practical problem in acoustics. We show that using five microphones and one loudspeaker, we can hear the shape of a room. We reformulate this problem as finding the locations of the image sources of the loudspeaker with respect to the walls. We propose an algorithm to find these positions only using first-order echoes. We prove that the reconstruction of the room is almost surely unique. We further introduce a new algorithm for locating a microphone inside a known room using only one loudspeaker. Our experimental evaluations conducted on the EPFL campus and also in the Lausanne cathedral, confirm the robustness and accuracy of the proposed methods.

By integrating further properties of EDMs into the matrix completion framework, we propose a new method for calibrating microphone arrays in a diffuse noise field. We use a specific characterization of diffuse noise fields to relate the coherence of recorded signals by two microphones to their mutual distance. As this model is not reliable for large distances between microphones,

we use matrix completion coupled with other properties of EDMs to estimate these distances and calibrate the microphone array. Evaluation of our algorithm using real data measurements demonstrates, for the first time, the possibility of accurately calibrating large ad-hoc microphone arrays in a diffuse noise field.

The last part of the thesis addresses a central problem in signal processing; the design of discrete-time filters (equivalently window functions) that are compact both in time and frequency. By properly adapting the definitions of compactness in the continuous time to discrete time, we formulate the search for maximally compact sequences as solving a semi-definite program. We show that the spectra of maximally compact sequences are a special class of Mathieu's cosine functions. Using the asymptotic behavior of these functions, we provide a tight bound for the time-frequency spread of discrete-time sequences. Our analysis shows that the Heisenberg uncertainty bound on the time-frequency spread of sequences is not tight and the lower bound depends on the frequency spread, unlike in the continuous time case.

Keywords: Euclidean Distance Matrices, Calibration, Sensor Localization, S-stress, Multidimensional Scaling, Acoustics, Heisenberg Uncertainty Principle, Maximally Compact Sequences

Résumé

Pièces maîtresses de nombreux domaines tel que la psychométrie, la spectroscopie RMN, l'apprentissage automatique, et réseaux de senseurs, les matrices de distance Euclidiennes (EDMs) se sont révélées utiles dans de nombreuses applications. Néanmoins, elles n'ont pas été souvent exploitées en traitement des signaux. Dans cette thèse, nous analysons les caractéristiques des EDMs et en déduisons d'importantes nouvelles propriétés. Ces analyses nous permettent de proposer de nouveaux algorithmes pour approcher les EDMs, ainsi que de donner des bornes analytiques sur la performance de ces méthodes. Ces techniques sont ensuite appliquées pour suggérer de nouvelles solutions à plusieurs problèmes pratiques en traitement des signaux. Munies de ces nouvelles propriétés, algorithmes et applications, les EDMs peuvent être considérées un outil fondamental à utiliser en traitement des signaux.

Plus en détail, nous commençons par présenter la structure et les propriétés des EDMs. En particulier, nous nous concentrons sur la propriété du rang; le rang d'une EDM est au plus la dimension de l'ensemble des points la générant plus 2. Cette propriété nous permet d'appliquer les méthodes de complétion de matrice de faible rang à l'approximation et la complétion d'EDMs seulement partiellement révélées. Nous appliquons cet algorithme au problème de l'étalonnage de la position des senseurs dans les appareils de tomographie ultrasonique. Nous présentons également non seulement une adaptation de cette méthode au processus d'auto-étalonnage de ces appareils, mais aussi des bornes analytiques sur l'erreur d'étalonnage.

Nous étudions ensuite le problème de la localisation de senseur exploitant l'information de distance par la minimisation d'une fonction de coût non-linéaire connue sous le nom de fonction *s-stress* dans la communauté du positionnement multidimensionnel (MDS). Nous déduisons des propriétés majeures de cette fonction de coût nous permettant de réduire le domaine de recherche pour trouver son minimum global. Nous produisons un algorithme distribué, efficace et peu coûteux, pour minimiser la fonction de coût pour des réseaux incomplets ainsi que des mesures bruitées. Nous montrons que dans des expériences aléatoires, la méthode proposée converge vers le minimum global du *s-stress* dans 99% des cas. Nous abordons le problème ouvert de l'existence de minimiseurs non-globaux du *s-stress* et réduisons ce problème à une hypothèse, dont la satisfaction ou non-satisfaction détermine l'existence ou l'absence, respectivement, de minimiseurs non-globaux de la fonction de coût.

Nous mettons à contribution la propriété du rang des EDMs, ainsi que l'algorithme de minimisation proposé pour leur approximation, pour aborder un problème d'acoustique aussi intéressant que pratique. Nous montrons qu'avec cinq microphones et un haut-parleur, il est possible d'entendre la forme d'une pièce. Nous reformulons ce problème en celui de trouver la position des images de la source sonore, le haut-parleur, par rapport aux murs. Nous proposons un algorithme pour trouver ces positions utilisant seulement les échos d'ordre premier. Nous prouvons que la reconstruction de la pièce est presque sûrement unique. Nous présentons en outre un nouvel algorithme pour la localisation d'un microphone à l'intérieur d'une pièce de

forme connue utilisant seulement un haut-parleur. Notre évaluation expérimentale, conduites sur le campus de l'EPFL ainsi que dans la cathédrale de Lausanne, confirme la robustesse et la précisions des méthodes proposées.

Par l'intégration de propriétés supplémentaires des EDMs dans nos algorithmes de complétion de matrice, nous proposons une nouvelle méthode pour l'étalonnage de réseaux de microphones dans un champ de bruit diffus. Nous utilisons une caractéristique spécifique des champs de bruit diffus afin de lier la cohérence de signaux enregistrés par deux microphones à leur distance mutuelle. Ce modèle n'étant pas fiable pour de longues distances entre les microphones, nous utilisons la complétion de matrice couplée à d'autres propriétés des EDMs afin d'évaluer ces distances et d'étalonner le réseau de microphones. L'évaluation de notre algorithmes avec des données réelles démontre pour la première fois la possibilité d'étalonner précisément de grands réseaux de microphones dans un champs de bruit diffus.

La partie finale de cette thèse traite d'un problème central en traitement des signaux; la conception de filtres à temps discrets (ou également fenêtre d'observation) à la fois compact en temps et en fréquence. Une adaptation pertinente de la définition de compacité en temps continu pour les filtres à temps discret nous permet de formuler la recherche de séquences maximalement compactes en un problème d'optimisation semi-définie positive. Nous montrons que les spectres des séquences maximalement compactes sont une classe spéciale des fonctions en cosinus de Mathieu. Exploitant le comportement asymptotique de ces fonctions, nous donnons une borne inférieure atteignable pour l'étalement temps-fréquence des séquences à temps discret. Notre analyse montre que le minorant donné par le principe d'incertitude d'Heisenberg pour l'étalement temps-fréquence des séquences n'est pas atteignable.

Mots-clés: Matrices de Distance Euclidiennes, Étalonnage, Localisation de Senseur, S-stress, Positionnement Multidimensionnel, Acoustiques, Principe d'Incertitude d'Heisenberg, Séquences Maximalement Compactes

Acknowledgments

I would like to express my deepest gratitude to my thesis advisor, Prof. Martin Vetterli. If it were not for him, I would not be studying in this school in the first place, and this thesis would not exist. When I received the admission letter for the M.Sc. program from EPFL in Jan. 2007, I immediately sent an e-mail to the only professor I knew there, Martin, asking for a research assistant position in his group. It was one of those typical one-page e-mails with lots of naïve self-praising phrases. It amazed me that he took the time to answer me saying, “Dear Sir, thank you for your interest. It seems you have been admitted to the MSc program, with a fellowship, thus my recommendation is that you enroll in the MSc program, and possibly take a project in our lab, so we get to know each other. Then, we can see what to do next. I hope this answers your question, and best regards martin”. I did not know that I had been offered the fellowship, and Martin’s message made my day. Thanks, Martin! Thanks for your unstoppable flow of new ideas, persistent inspiration, wise guidance, true friendship and most importantly for your trust.

I would like to thank my committee members, Prof. Helmut Bölcskei, Dr. Frank Quick, Prof. Emre Telatar and Prof. Rüdiger Urbanke, for accepting to assess my thesis and taking the time to read it.

I would like to offer my special thanks to the past and current members of LCAV for creating such a rich environment for research. I have enjoyed every second of our invaluable collaborations, wonderful coffee breaks, and fun ski and hiking days.

I have been very fortunate to know and to live with two incredible friends: Hamed Hassani and Farid Movahedi Naini. Farid and I came to Switzerland six years ago on the same flight and Hamed joined us a few weeks later. Since then, we have been living together. I would not hesitate for a second to call them brothers. I would like to also thank my wonderful friends Vahid Aref, Saeid Haghghatshoar and Masih Nilchian to whom I owe many of the lessons I have learned in life.

Words do not do justice to the expression of my debt to and appreciation of my family, but I will try anyways. I would like to start with my lovely sister, Mahsa. Thanks for always being there. I remember vividly the hours you would spend writing several different compositions for us, in case one would be the topic of the exam. Thanks for always being the light on the road to future. You have been and will be *our* role model. I am also very grateful to my brother-in-law, Nima for his true friendship and for being beside my parents while both their sons are far away.

My foremost gratitude and recognition goes to my father and mother, Ali and Soudabeh, for their unconditional and unflinching love, support and dedication. If you had experienced their love, you would not believe it, otherwise you would not be able to imagine it. I do not know any parents who have sacrificed so much for their children. If one day I can be even half as good to my children, as they were to us, I would be already a wonderful parent. I thank them for teaching us the true definition of love, and for giving me the best possible gift in the world that one could wish for.

You might have noticed that in the above words I used “our” or “us” several times. My twin brother, Mohammad has been and will be an inseparable part of me. People normally say that I look older, but the truth is Mohammad has always been the big brother. No matter how childish, selfish or foolish I was, he has never given up on me. Thank you Mohammad! You do not stop surprising me with how much wisdom and life experience you have. As if you were a decade older than me! I would not hesitate to trade this thesis for a fraction of your wisdom.

This thesis is dedicated with love to my parents.

Contents

Abstract	iii
Résumé	v
Acknowledgments	vii
1 Introduction	1
1.1 Motivation	1
1.2 Euclidean Distance Matrices	3
1.2.1 Invariance Under Rigid Transformation	6
1.2.2 Embedding Dimension and the Rank of an EDM	7
1.2.3 The EDM Cone	8
1.3 Multidimensional Scaling	9
1.3.1 Classic Multi-dimensional Scaling	10
1.3.2 Stress Function Minimization	11
1.3.3 S-stress Function Minimization	12
1.4 Sensor Localization, Calibration and Uncertainties	12
1.5 Thesis Outline and Contributions	13
2 Sensor Calibration for Ultrasound Tomography Devices	15
2.1 Introduction	15
2.2 Calibration as a Dual to Sensor Localization	17
2.3 Circular Time of Flight Tomography	18
2.3.1 Homogeneous Medium and Dimensionality Reduction	19
2.3.2 Time-of-Flight Estimation	19
2.4 Background	20
2.4.1 Matrix Completion	20
2.4.2 Reconstruction Error Measure	22
2.5 Mathematical Formulation	22
2.6 Calibration Algorithm and Theoretical Bounds	24
2.6.1 Computational Complexity	26
2.7 Numerical Evaluations	26
2.8 Summary	29
2.A Analysis	30

3	Distributed Low-Complexity Sensor Localization	39
3.1	Introduction	39
3.2	The S-stress Criterion For Sensor Localization	40
3.3	Distributed Sensor Localization With the S-stress Criterion	41
3.4	Algebraic Properties of the S-stress Function	45
3.5	Does the S-stress Function Have Non-Global Minimizers?	53
3.6	An Optimal English Keyboard Design	56
3.7	Summary	59
3.A	Analysis	61
4	Room Geometry Estimation and In-Room Localization Using EDMs	63
4.1	Introduction	63
4.2	Room Geometry Estimation Using EDMs	64
4.2.1	Modeling the Room	64
4.2.2	Echo Labeling Using EDMs	66
4.2.3	Practical Room Geometry Estimation Algorithm	68
4.2.4	Uniqueness	69
4.2.5	Experimental Evaluations	69
4.3	In-Room Microphone Localization Using EDMs	71
4.3.1	Echo Labeling for Microphone Localization	72
4.3.2	Practical Echo Labeling for Microphone Calibration	73
4.3.3	Experimental Evaluations	73
4.4	Summary	74
4.A	Analysis	76
5	Microphone Array Calibration with Diffuse Noise	79
5.1	Introduction	79
5.2	Distance Estimation in a Diffuse Noise Field	80
5.2.1	Coherence in Diffuse Noise Fields	80
5.2.2	Distance Estimation from Coherence Information	82
5.3	Microphone Calibration in Diffuse Noise Field	83
5.3.1	Cadzow Projection to EDM Properties	84
5.4	Experimental Evaluations	85
5.4.1	Diffuse Noise Recording Set-up	85
5.4.2	Microphone Calibration with Simulated Data	86
5.4.3	Microphone Calibration with Real Data	86
5.5	Summary	87
6	Sequences with Minimal Time-Frequency Spreads	89
6.1	Introduction	89
6.1.1	Uncertainty principles for sequences	90
6.1.2	Contribution	93
6.1.3	Related Work	94
6.2	Maximally Compact Sequences	94
6.2.1	Properties of Maximally Compact Sequences	95
6.2.2	Design of Maximally Compact Sequences and Their Uncertainty Bounds	96

Contents	xi
<hr/>	
6.3 Numerical Analysis	99
6.4 Summary	99
6.A Analysis	101
7 Conclusion	111
Bibliography	113
Curriculum Vitæ	123

Chapter 1

Introduction

It is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail.

Abraham Harold Maslow

1.1 Motivation

Euclidean distance matrices (EDMs) are probably one of the most useful and at the same time under-appreciated tools in signal processing. Studies on the properties of these objects do not go back further than a century. In this thesis, by studying the properties of these objects and introducing new algorithms and applications, we will try to convince the reader that EDMs deserve more attention in the signal processing and communications communities.

Although the notions of metric spaces and distances were introduced long before by Fréchet [40], it was not until 1935 that Schoenberg in [114] studied the properties of Euclidean distance matrices in detail. Further properties were developed by Young and Householder in 1938 [137]. In 1952, Togerson introduced the concept of multidimensional scaling (MDS) as the problem (or the procedure) of finding a set of points that produce a given inter-distance matrix [126]. This distance matrix could be generated from a Euclidean geometry or a set of dissimilarities. However, the relation between the used distances and the non-metric dissimilarities was defined vaguely. Later in 1964, Kruskal suggested the notion of *stress* as a goodness of fit for non-metric data [73]. The most significant practical impact of EDMs until that time can be considered the successful test of Transit, the first satellite navigation system used by the United States Navy in 1960. Later, in a series of papers [47, 48], Gower showed further properties of Euclidean geometry and approaches for the approximations of EDMs. In his 1985 paper [48], Gower rigorously presented the relation of the rank of EDMs to their embedding dimension.

While the mathematicians were busy proving abstract properties of EDMs, biologists started using them to estimate the shape of proteins. In early 1980s, Williamson, Havel and Wüthrich developed the idea of using nuclear magnetic resonance (NMR) to extract inter-distances of hydrogen atoms in proteins and use these distances to reconstruct the shape of the molecule [53, 135]¹. In 1990, Glunt et al. in [42] and Hayden et al. in [54], provided some insights on the structure of the cone of EDMs. It took the multidimensional scaling community a few more years to come up with a solution for the molecular conformation through Trosset in 1998 [128]. Since 2000, the practical advantages of EDMs started to appear also in the machine learning community; examples include papers by Tenenbaum et al. in 2000 [123] for image understanding and handwriting recognition, Jain et al. in 2004 [59], for speech and music and Weinberger et al. also in 2004 [134], for learning image manifolds. Also starting 1999, as the interest in sensor networks increased, several approaches based on EDM properties were proposed for sensor localization [3, 12, 29].

Let us give a short description of two of the EDM applications mentioned above. Many other applications will be introduced throughout the thesis.

Protein Structure Prediction

Knowing the structure of proteins is crucial for understanding their physical and chemical properties and interactions. It is also useful for drug design. One of the methods to estimate the structure of a protein is nuclear magnetic resonance spectroscopy (usually abbreviated as protein NMR). The process of determining the protein structure by NMR consists of measuring many short inter-distances between hydrogen atoms and restraining the protein structure with these distances. Large distance measurements (larger than 5 Å [52]) are normally unreliable and thus a network of short distances is available for reconstructing the structure of the protein. This produces an incomplete and noisy EDM from which the location of the hydrogen atoms must be estimated. Many approaches including distance matrix completion [17] and methods based on molecular dynamics and simulated annealing [92] are proposed to solve this problem. In Figure 1.1 we show an example reconstruction of the protein 2E8O, SAM domain, with a method based on semi-definite programming called SPROS [4]. The blue structure shows the estimated protein structure using SPROS and the red one is the reference structure.

Dimensionality Reduction in Machine Learning

In many applications of machine learning (such as face recognition [130] or handwriting recognition [58]) the high-dimensional measurements lie on a low-dimensional but *non-linear* manifold. In order to be able to analyze these data efficiently, we need to find a low dimensional embedding of these data points. One of the approaches to this problem is called the isometric feature mapping or *Isomap* [123]. Isomap finds the paths between the neighboring nodes and guesses larger distances by finding the shortest paths between the corresponding nodes. Then, using these geodesic distances and applying simple localization algorithms, the method estimates the new node positions in the low-dimensional space. An example of the application of the algorithm on the Swiss roll data-set is shown in Figure 1.2. Several nodes are lying on a 2-dimensional manifold in 3D (see Figure 1.2(a)). We can observe that for two selected nodes on

1. Wüthrich received the Chemistry Nobel Prize in 2002 for “his development of nuclear magnetic resonance spectroscopy for determining the three-dimensional structure of biological macromolecules in solution”

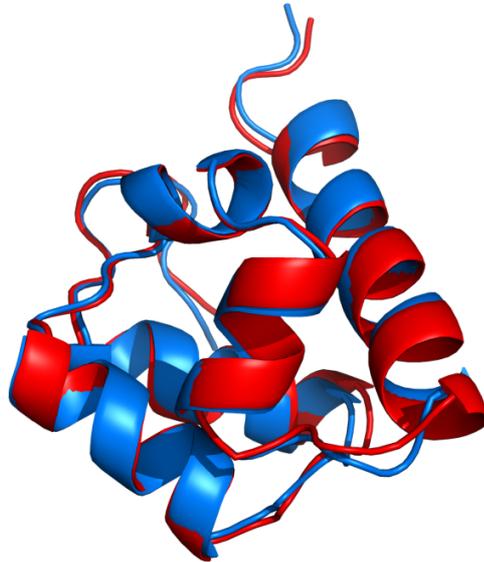


Figure 1.1: Structure estimated by SPROS (red) and the reference structure (blue) for the protein 2E8O, SAM domain. Image taken from [4] with permission.

the non-linear manifold, their Euclidean distance in 3D (blue line) may not accurately show their similarity, as measured by geodesic distance along the low-dimensional manifold (red curve). Using Isomap, we can find a non-linear embedding of the data-set in 2D (Figure 1.2(b)). In the low-dimensional embedding, the Euclidean distance between data points represents more accurately their geodesic distance.

When we go through the history of EDMs and their applications, we unfortunately do not witness many results from the signal processing community with an EDM flavor². This lack of appreciation, has also held the community back from developing efficient algorithms that fit specific applications. In these pages, we revisit some of the properties of EDMs, propose new algorithms to approximate them and test them through several applications in signal processing.

Let us continue with introducing some basic properties of EDMs which will be used in this thesis.

1.2 Euclidean Distance Matrices

Consider a list of points $\{\mathbf{x}_i, i = 1, \dots, n\}$ in the Euclidean space \mathbb{R}^η of dimension η . A matrix $\mathbf{D} \in \mathbb{R}_+^{n \times n}$ is called a Euclidean distance matrix (EDM), when its entries, $d_{i,j}^2$ are the

2. Of course one might argue that we can call many things as “signal processing” and many of the mentioned applications lie in signal processing!

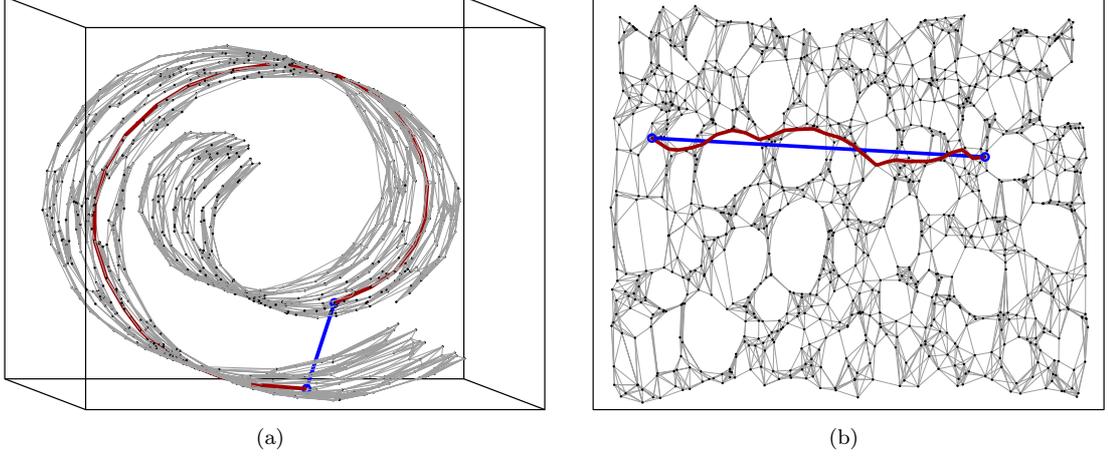


Figure 1.2: The Swiss roll data-set in a 3-dimensional space (a), and its non-linear embedding in 2D using Isomap (b). The images are reproduced from [123] and the data-set provided in <http://isomap.stanford.edu>.

Euclidean distance-squares between pairs of \mathbf{x}_i and \mathbf{x}_j , i.e.,

$$\begin{aligned} D[i, j] &= d_{i,j}^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \langle \mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_i - \mathbf{x}_j \rangle \\ &= \langle \mathbf{x}_i, \mathbf{x}_i \rangle + \langle \mathbf{x}_j, \mathbf{x}_j \rangle - 2 \langle \mathbf{x}_i, \mathbf{x}_j \rangle. \end{aligned} \quad (1.1)$$

As a result, any element of an EDM must satisfy the basic Euclidean metric properties [71]:

M1. Non-negativity

$$d_{i,j} \geq 0.$$

M2. Self-distance

$$d_{i,j} = 0 \iff \mathbf{x}_i = \mathbf{x}_j.$$

M3. Symmetry

$$d_{i,j} = d_{j,i}.$$

M4. Triangle inequality

$$d_{i,j} \leq d_{i,k} + d_{k,j}.$$

Note that these properties are necessary but not sufficient for a matrix to be an EDM.

Example 1.1

Consider the points $\mathbf{x}_1 = [1, 1, 0]^T$, $\mathbf{x}_2 = [5, 1, 0]^T$, $\mathbf{x}_3 = [1, 4, 0]^T$ and $\mathbf{x}_4 = [1, 1, 1]^T$ in \mathbb{R}^3 . We can find their corresponding distance matrix as \mathbf{D}_1 ,

$$\mathbf{D}_1 = \begin{bmatrix} 0 & 16 & 9 & 1 \\ 16 & 0 & 25 & 17 \\ 9 & 25 & 0 & 10 \\ 1 & 17 & 10 & 0 \end{bmatrix}, \quad \mathbf{D}_2 = \begin{bmatrix} 0 & 16 & 9 & 36 \\ 16 & 0 & 25 & 17 \\ 9 & 25 & 0 & 10 \\ 36 & 17 & 10 & 0 \end{bmatrix}.$$

Clearly, by construction \mathbf{D}_1 is an EDM and satisfies all the metric properties. Consider now \mathbf{D}_2 as a simple modification of \mathbf{D}_1 . The new matrix \mathbf{D}_2 satisfies all the metric properties M1–M4. However, it is not an EDM (we will see shortly how to check if a matrix is an EDM).

Let us now describe the necessary and sufficient conditions for a matrix to be an EDM. In this regard, we need to define the following notions:

Definition 1.1 (Symmetric hollow subspace)

Denoted by \mathbb{S}_h^n , the symmetric hollow subspace is a proper subspace of symmetric matrices \mathbb{S}^n with a zero diagonal.

$$\mathbb{S}_h^n \stackrel{\text{def}}{=} \{\mathbf{A} \in \mathbb{S}^n \mid \text{diag}(\mathbf{A}) = \mathbf{0}\},$$

where $\text{diag}(\cdot)$ denotes a column vector with the diagonal entries of its input matrix.

Definition 1.2 (Positive semi-definite cone)

Denoted by \mathbb{S}_+^n , the positive semi-definite cone is the set of all symmetric positive semi-definite matrices of dimension $n \times n$.

$$\mathbb{S}_+^n \stackrel{\text{def}}{=} \{\mathbf{A} \in \mathbb{S}^n \mid \mathbf{A} \succeq \mathbf{0}\}.$$

Let us also define the geometric centering matrix \mathbf{L} as

$$\mathbf{L} \stackrel{\text{def}}{=} \mathbf{I} - \frac{1}{n} \mathbf{1} \mathbf{1}^T, \quad (1.2)$$

where \mathbf{I} is the $n \times n$ identity matrix and $\mathbf{1}$ is the all one column vector in \mathbb{R}^n .

Theorem 1.1 (Schoenberg [114])

$$\mathbf{D} \text{ is an EDM} \iff \begin{cases} -\mathbf{L} \mathbf{D} \mathbf{L} \in \mathbb{S}_+^n \\ \mathbf{D} \in \mathbb{S}_h^n \end{cases} \quad (1.3)$$

Theorem 1.1 is very important in the sense that it provides a necessary and sufficient condition for a matrix to be an EDM, while the metric properties M1–M4 do not provide such functionality.

In the following we provide yet another important property of EDMs.

1.2.1 Invariance Under Rigid Transformation

Let us collect the coordinates of the set of points in a matrix \mathbf{X} as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} \in \mathbb{R}^{n \times \eta},$$

with $\mathbf{x}_i \in \mathbb{R}^\eta$. This, for example, can be the collection of n sensors thrown in a 3-dimensional ($\eta = 3$) field.

An EDM \mathbf{D} must be expressible as a function of some \mathbf{X} . Let us call that function D . From (1.1), we can write

$$D(\mathbf{X}) = \text{diag}(\mathbf{X}\mathbf{X}^T)\mathbf{1}^T + \mathbf{1}\text{diag}(\mathbf{X}\mathbf{X}^T)^T - 2\mathbf{X}\mathbf{X}^T. \quad (1.4)$$

In words, for every set of points in \mathbb{R}^η , the function $D(\mathbf{X})$ outputs the EDM corresponding to that point set. Given an EDM \mathbf{D} , there are in general infinitely many position matrices \mathbf{X} that generate \mathbf{D} . They are related with rigid transformations (also called isometric transformations): translation, rotation and reflection.

Translation Invariance

A translation in \mathbb{R}^η is represented by

$$\mathbf{Y} = \mathbf{X} - \mathbf{1}\mathbf{t}^T,$$

where $\mathbf{t} \in \mathbb{R}^\eta$ is the translation vector. It is easy to see from (1.4) that for every $\mathbf{t} \in \mathbb{R}^\eta$,

$$D(\mathbf{X} - \mathbf{1}\mathbf{t}^T) = D(\mathbf{X}).$$

Example 1.2 (Centering a set of points at the origin)

If a set of n points are listed in $\mathbf{X} \in \mathbb{R}^{n \times \eta}$, their geometric center is calculated as [26]:

$$\boldsymbol{\mu} = \frac{1}{n}\mathbf{X}^T\mathbf{1}.$$

Centering the set of points at the origin is equivalent to

$$\begin{aligned} \mathbf{X}_c &= \mathbf{X} - \mathbf{1}\boldsymbol{\mu}^T \\ &= \mathbf{X} - \mathbf{1}\left(\frac{1}{n}\mathbf{1}^T\mathbf{X}\right) \\ &= \left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\mathbf{X} \\ &= \mathbf{L}\mathbf{X}. \end{aligned}$$

This is why we called the matrix \mathbf{L} in (1.2) the geometric centering matrix.

It is easy to verify that

$$D(\mathbf{L}\mathbf{X}) = D(\mathbf{X}).$$

Rotation and Reflection Invariance

The rotation and/or reflection of a set of points \mathbf{X} around a point $\mathbf{z} \in \mathbb{R}^\eta$ is achieved by

$$\mathbf{X}_r = (\mathbf{X} - \mathbf{1} \mathbf{z}^T) \mathbf{Q},$$

where $\mathbf{Q} \in \mathbb{R}^{\eta \times \eta}$ is an orthogonal matrix.

We can verify that

$$\begin{aligned} D(\mathbf{X}_r) &= D((\mathbf{X} - \mathbf{1} \mathbf{z}^T) \mathbf{Q}) \\ &= D(\mathbf{X} \mathbf{Q} - \mathbf{1} \mathbf{w}^T) \\ &= D(\mathbf{X} \mathbf{Q}) \\ &\stackrel{(a)}{=} D(\mathbf{X}), \end{aligned}$$

where (a) follows from (1.4) using $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$.

1.2.2 Embedding Dimension and the Rank of an EDM

An important notion in the Euclidean distance geometry is called the embedding (or affine) dimension of an EDM.

Definition 1.3 (*Embedding dimension* [48])

If a matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ is an EDM, its embedding or affine dimension is the rank of \mathbf{X} with the least rank that generates \mathbf{D} .

In other words, the embedding dimension of an EDM is the dimension of the smallest affine set in \mathbb{R}^n containing \mathbf{X} , the set of points that generate it.

Theorem 1.2 (*EDM rank vs. embedding dimension* [48])

For a Euclidean distance matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ with embedding dimension r , we have

$$\text{rank}(\mathbf{D}) \leq r + 2. \tag{1.5}$$

Further, $\text{rank}(\mathbf{D}) = r + 1$, if and only if the points generating \mathbf{D} lie on the relative boundary of an r -dimensional hypersphere.

Proof.

In order to prove the upper bound in (1.5), we use the formulation of $D(\mathbf{X})$ in (1.4),

$$\begin{aligned} \text{rank}(D(\mathbf{X})) &\leq \text{rank}(\text{diag}(\mathbf{X} \mathbf{X}^T) \mathbf{1}^T) + \text{rank}(\mathbf{1} \text{diag}(\mathbf{X} \mathbf{X}^T)^T) + \text{rank}(2\mathbf{X} \mathbf{X}^T) \\ &\leq 1 + 1 + r, \end{aligned}$$

where we used the fact that $\text{rank}(\mathbf{A} + \mathbf{B}) \leq \text{rank}(\mathbf{A}) + \text{rank}(\mathbf{B})$. The rest of the proof can be found in [48]. \square

Note that the rank of an EDM is independent of its dimensions (i.e. the number of points generating that EDM). The result of Theorem 1.2 lies at the heart of this thesis and is very central to the obtained results.

An important question is how to find the embedding dimension of a distance matrix? The answer is simple [48]; From (1.4), we can see that

$$-LDL = 2LXX^T L,$$

because $L\mathbf{1} = \mathbf{0}$. Now

$$\begin{aligned} r &= \text{rank}(\mathbf{X}) \\ &= \text{rank}(\mathbf{LX}) \\ &\stackrel{(a)}{=} \text{rank}(\mathbf{LDL}). \end{aligned} \tag{1.6}$$

For (a) we have used the fact that $\text{rank}(\mathbf{A}^T \mathbf{A}) = \text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^T)$. Thus, in order to find the embedding dimension of an EDM \mathbf{D} , it is enough to compute $\text{rank}(\mathbf{LDL})$.

1.2.3 The EDM Cone

Let EDM^n denote the set of all EDMs of dimension $n \times n$. The set EDM^n forms a closed convex cone; using (1.3), for any $\lambda_1, \lambda_2 \geq 0$:

$$\begin{aligned} LD_1L \succeq 0 \quad LD_2L \succeq 0 \\ D_1 \in \mathbb{S}_h^n \quad D_2 \in \mathbb{S}_h^n \end{aligned} \implies \begin{aligned} \lambda_1 LD_1L + \lambda_2 LD_2L \succeq 0 \\ \lambda_1 D_1 + \lambda_2 D_2 \in \mathbb{S}_h^n \end{aligned}$$

Note that EDM^n contains all the EDMs in \mathbb{S}_h^n with embedding dimensions ranging from 0 to $n - 1$ (since $D(\mathbf{X})$ is invariant under translation, the maximum embedding dimension of an $n \times n$ EDM is $n - 1$ as the dimension of the points can be reduced by at least one).

Now the question is which portion of this cone belongs to EDMs with certain embedding dimensions. Recall from (1.6) that $r = \text{rank}(\mathbf{LDL})$. We have the following lemma.

Lemma 1.1 (The EDM cone and the embedding dimension [26])

The EDM cone relative interior comprises

$$\text{rel int } \text{EDM}^n = \{\mathbf{D} \in \text{EDM}^n \mid \text{rank}(\mathbf{LDL}) = n - 1\},$$

which is a convex cone. Also the relative boundary of the EDM cone is

$$\text{rel } \partial \text{EDM}^n = \bigcup_{r=0}^{n-2} \{\mathbf{D} \in \text{EDM}^n \mid \text{rank}(\mathbf{LDL}) = r\}.$$

None of these sets are necessarily convex.

Lemma 1.1 states that all the interior of the EDM cone consists only of EDMs with embedding dimension $n - 1$ and the rest of EDMs lie only on the relative boundary of the cone. This lemma plays a crucial role in showing the hardness of problems that try to find the best EDM approximations to a matrix with a certain embedding dimension. The set of such EDMs is not convex anymore. Further, Hayden et al. in [54] show that for $n > 3$, EDM^n is not a circular cone anymore. These properties show why the approximation is not straight forward.

One particular problem of interest is completing a partially revealed EDM. This problem can be formulated as

$$\begin{aligned} \text{find} \quad & \mathbf{D}_{\text{est}} \in \text{EDM}^n \\ \text{s.t.} \quad & \mathbf{D}_{\text{est}}[i, j] = \mathbf{D}[i, j] \quad \text{for } (i, j) \text{ revealed indices.} \end{aligned} \tag{1.7}$$

If \mathbf{D} is a partial EDM in \mathbb{R}^r with embedding dimension r , we are interested in completing it so that the result has also an embedding dimension r . Then problem (1.7) can be further restricted to

$$\begin{aligned} \text{find } & \mathbf{D}_{\text{est}} \in \text{EDM}^n \\ \text{s.t. } & \mathbf{D}_{\text{est}}[i, j] = \mathbf{D}[i, j] \quad \text{for } (i, j) \text{ revealed indices} \\ & \text{emb dim}(\mathbf{D}_{\text{est}}) = r. \end{aligned}$$

This problem is NP-hard.

Theorem 1.3 ([72, 113])

The problem of one-embeddability of graphs³ with integer weights is NP-complete.

For this reason, all the solutions for Euclidean distance matrix completion or approximation problems are relaxed in order to make the solutions tractable (e.g. [12, 60]). In Chapter 2 we see a relaxed formulation for the completion problem. Also in Chapter 3 we provide an optimization framework to tackle the problem locally.

One of the earliest implicit applications of EDMs is for data visualization [126]. This task of down-scaling a set of high dimensional data, with a given distance matrix, into a lower dimension (which might have multi dimensions) and visualizing them in the lower dimensional space is called multidimensional scaling (MDS). This process is in essence the same as sensor localization from mutual distances and similar results are developed in both fields in parallel. In the following section we briefly introduce multidimensional scaling and some of the common tools for solving related problems in this field.

1.3 Multidimensional Scaling

Multidimensional scaling (MDS) was originally proposed in psychometrics [73, 126] to visualize the (dis-)similarities between objects (or stimuli, like colors). It is defined as the *problem* of finding n points (normally in a certain dimension) whose inter-point distances represent (dis-)similarities between objects. The term “multidimensional scaling” is often used to refer to methods for solving MDS (e.g. [14]). The following example is borrowed from [14] to showcase one of the initial problems in MDS.

Example 1.3 (MDS and Color Similarities)

Ekman in 1954 [36] used 14 colors differing only in their wavelength. Different pairs of colors were projected on a screen and the average score of 31 subjects for their similarities were taken (0: not similar, 1: identical). The measured similarities are shown in Table 1.1. With the simplest algorithm called the classic MDS — which will be introduced in the following lines — we can associate a set of points in two dimensions to these wavelengths. The reconstructed locations are shown in Figure 1.3. The figure suggests close correlation of the visualization with the well-known color circle (or color wheel) [61].

Although the MDS problem originated for different set of purposes and has been widely used in a separate research community, it addresses the same fundamental question in sensor

3. One-embeddability of graphs is a special case of the k -embeddability problem which accounts for finding an embedding of a weighted graph in k dimensions that preserves the weights.

λ (nm)	434	445	465	472	490	504	537	555	584	600	610	628	651	674
434	1.0	.86	.42	.42	.18	.06	.07	.04	.02	.07	.09	.12	.13	.16
445	.86	1.0	.50	.44	.22	.09	.07	.07	.02	.04	.07	.11	.13	.14
465	.42	.50	1.0	.81	.47	.17	.10	.08	.02	.01	.02	.01	.05	.03
472	.42	.44	.81	1.0	.54	.25	.10	.09	.02	.01	.00	.01	.02	.04
490	.18	.22	.47	.54	1.0	.61	.31	.26	.07	.02	.02	.01	.02	.00
504	.06	.09	.17	.25	.61	1.0	.62	.45	.14	.08	.02	.02	.02	.01
537	.07	.07	.10	.10	.31	.62	1.0	.73	.22	.14	.05	.02	.02	.00
555	.04	.07	.08	.09	.26	.45	.73	1.0	.33	.19	.04	.03	.02	.02
584	.02	.02	.02	.02	.07	.14	.22	.33	1.0	.58	.37	.27	.20	.23
600	.07	.04	.01	.01	.02	.08	.14	.19	.58	1.0	.74	.50	.41	.28
610	.09	.07	.02	.00	.02	.02	.05	.04	.37	.74	1.0	.76	.62	.55
628	.12	.11	.01	.01	.01	.02	.02	.03	.27	.50	.76	1.0	.85	.68
651	.13	.13	.05	.02	.02	.02	.02	.02	.20	.41	.62	.85	1.0	.76
674	.16	.14	.03	.04	.00	.01	.00	.02	.23	.28	.55	.68	.76	1.0

Table 1.1: Average similarity scores of colors with wavelengths from 434 to 674 nm [36] used in Example 1.3.

localization and signal processing: finding a set of points with a certain dimension that generate a given set of (possibly noisy) distances.

The MDS literature offers many formulations and algorithms to estimate the set of points \mathbf{X} from \mathbf{D} . In the following lines we provide a brief description of three of the most famous approaches to this problem.

1.3.1 Classic Multi-dimensional Scaling

If all the pairwise distances are measured without error, then a naïve algorithm called classic MDS exactly recovers the correct configuration of points [34, 75, 115]. Let us now explain this algorithm.

Let \mathbf{L} be as in (1.2). Recall from (1.4), that

$$-\mathbf{LDL} = 2\mathbf{LXX}^T\mathbf{L}.$$

Then, given the singular value decomposition (SVD) of the symmetric and positive semi-definite matrix $(-1/2)\mathbf{LDL}$ as $(-1/2)\mathbf{LDL} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$, we have

$$\text{MDS}_\eta(\mathbf{D}) \triangleq \mathbf{U}_\eta \mathbf{\Sigma}_\eta^{1/2},$$

where \mathbf{U}_η denotes the $n \times \eta$ left singular matrix corresponding to the η largest singular values and $\mathbf{\Sigma}_\eta$ denotes the $\eta \times \eta$ diagonal matrix with η largest singular values in the diagonal. Note that as we showed in Example 1.2, the operation \mathbf{LX} only shifts the center of the set of the points to the origin. This is also known as the MDSLOCALIZE algorithm in [34]. The algorithm is summarized in Algorithm 1.1.

Although the algorithm provides exact solutions in noiseless cases, there is no guarantee for optimality of the solution in the noisy case.

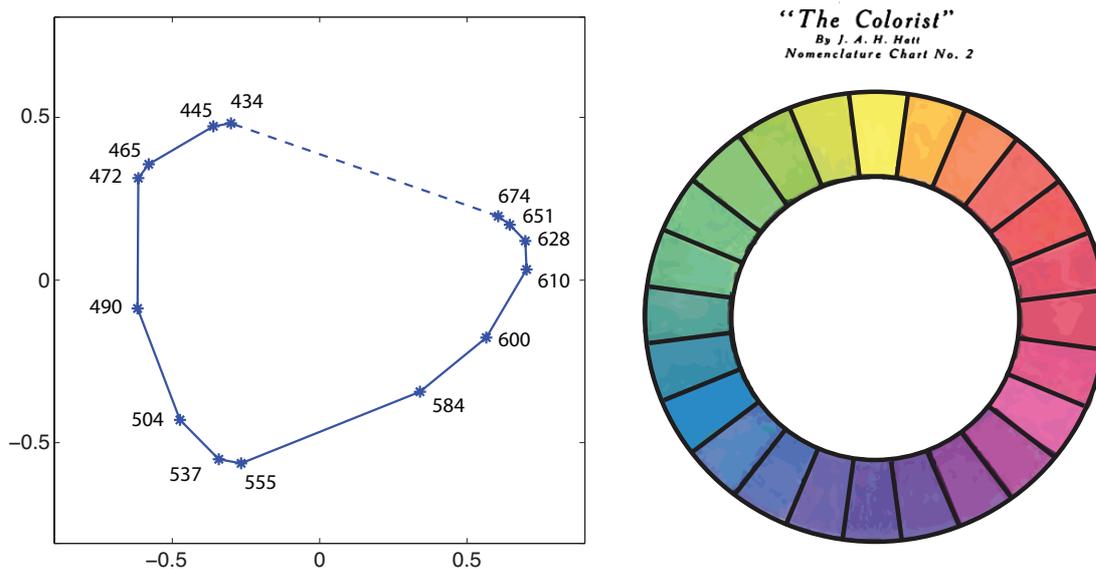


Figure 1.3: The output of metric MDS on the color similarity data of [36] with embedding dimension 2. The numbers on the plot represent the wavelength of each color. The plot resembles the color wheel (also called the color circle) [61]. Image taken from Wikipedia (http://en.wikipedia.org/wiki/Color_wheel).

Algorithm 1.1 Classic MDS (MDSLOCALIZE) [115].

Input: Dimension η , estimated squared distance matrix D

Output: Estimated positions $\text{MDS}_\eta(D)$

1. Compute $(-1/2)\mathbf{L}\mathbf{D}\mathbf{L}$;
 2. Compute the best rank- η approximation $\mathbf{U}_\eta \mathbf{\Sigma}_\eta \mathbf{U}_\eta^T$ of $(-1/2)\mathbf{L}\mathbf{D}\mathbf{L}$;
 3. Return $\text{MDS}_\eta(D) \triangleq \mathbf{U}_\eta \mathbf{\Sigma}_\eta^{1/2}$.
-

1.3.2 Stress Function Minimization

Another way to solve the MDS problem is to use optimization methods. From Lemma 1.1 we know that for a fixed embedding dimension, the target set is not convex. Thus, the existing optimization methods are not only non-linear, but also non-convex. If we consider the distances instead of the squared distances, the optimization problem is called the *raw Stress* [74]:

$$\underset{\mathbf{X} \in \mathbb{R}^{n \times \eta}}{\text{minimize}} \quad \sum_{i,j} w_{i,j} \left(\sqrt{D(\mathbf{X})[i,j]} - \sqrt{D[i,j]} \right)^2. \quad (1.8)$$

The weights $w_{i,j}$ are zero if the measurement (i,j) is not known and $w_{i,j} \geq 0$ for the rest of measurements. As this cost function is not globally differentiable, optimization methods for solving it are more involved. There are several approaches for this problem such as iterative majorization [27, 28], methods using convex analysis [86, 87] and steepest descent methods [50, 74].

1.3.3 S-stress Function Minimization

The s-stress cost function was first introduced by Takane, Young and De Leeuw [122]:

$$\underset{\mathbf{X} \in \mathbb{R}^{n \times \eta}}{\text{minimize}} \quad \sum_{i,j} w_{i,j} (D(\mathbf{X})[i,j] - D[i,j])^2 .$$

Note that, contrary to the Stress function, the s-stress criterion is differentiable everywhere. However, one disadvantage of the s-stress functions is the fact that it favors larger distances over small ones [122]. Gaffke and Mathar [41] propose an algorithm to find the global minimum of the s-stress function for embedding dimension $\eta = n - 1$. We bear in mind that EDMs with this embedding dimension constitute a convex set (Lemma 1.1). We are rather interested in cases where the embedding dimension is significantly smaller than n .

In this thesis, the s-stress function is considered in great detail. We study many of its properties in Chapter 3 and introduce a new low complexity algorithm to minimize it in a distributed manner. We also show applications of this cost function and the minimization algorithm in Chapters 4 and 5.

1.4 Sensor Localization, Calibration and Uncertainties

Independent of the specific applications, if you have a network of sensors that collaboratively measure a quantity, knowing their location is of interest. This network of sensors could be a few microphones deployed in a room to record sound or used for teleconferencing [39], it can be a network of sensors dropped from an airplane to measure the presence of enemies in a large field [77], a sensor network to estimate the temperature map in the Alps [57] or hundreds of ultrasound sensors used for the diagnosis of breast cancer [35]. In almost all applications of sensor networks, it is not possible to design the system in such a way that the position of the sensors is known and fixed; the obvious case is when the sensors are dropped from an airplane and the non-obvious one is when a ring of ultrasound transducers is designed for cancer imaging. In the latter, although designed with high precision, the positions are sometimes found using an X-ray scan of the ring to determine the exact position of the sensors.

In this thesis, we are mostly focused on finding the position of sensors using their pair-wise distance information. If all the distances between the sensors were known and noiseless, the task of finding the sensor positions would be very simple (using the classic MDS introduced in Section 1.3.1). In many applications however, not all the distance information is available and even the known distances are noisy. These uncertainties make the localization procedure challenging.

Normally in a sensor network, the missing distance information correspond to sensors which are far from each other (because of energy constraints for communication). In some applications, however, the distance measurements for close-by sensors are missing. This is in particular true for the calibration of sensors in ultrasound tomography devices, explained in Chapter 2. In these cases, the classic methods for sensor localization need to be properly adapted for accurate positioning of sensors.

The main part of this thesis focuses on the applications of EDMs and sensor positioning. Heisenberg uncertainty principle on the other hand states that accurate location yields inaccurate momentum and vice versa. In Chapter 6 we study the uncertainty principle for discrete sequences. Although the chapter might seem unrelated to the context of this thesis, there is no doubt that the fundamental question of time-frequency uncertainty for discrete sequences plays an important

role in all fields of signal processing. It suffices to note that in all the applications presented in this thesis, some sort of windowing has been performed on the signals of interest. The design of optimal window functions and their frequency characteristics are considered in Chapter 6.

1.5 Thesis Outline and Contributions

Each result presented in this thesis has originated from and was motivated by a practical problem in the field of signal processing. The problems at hand have defined the tools to be used in this work. Thus, the contributions of the thesis are presented by their corresponding applications, i.e. in each chapter we start by explaining the practical problem at hand, and then we present the tools to solve such a problem.

In the following we present a brief summary of each chapter and its contributions.

Calibration Through Matrix Completion

In Chapter 2, we study the calibration of ultrasound tomography devices using time-of-flight (ToF) measurements. In particular, we study circular tomography devices where the sensors do not lie on a perfect circle. After the manufacturing process, either the sensors are assumed to lie on a circle, or an X-ray image provides an estimate of their positions. We aim at finding a self calibration process for such devices with no extra equipment. In the calibration process we face however some obstacles: The distance information for close-by transducers are missing due the beam shape of the ultrasound sensors (it is very hard to build small omni-directional ultrasound transducers). Further, due to malfunction of sensors or excessive noise, some measurements are not reliable and thus useless in the distance matrix.

We use the rank property of EDMs (Theorem 1.2) and a state-of-the-art low rank matrix completion, OPTSPACE, to complete the distance matrix. We then use the classic MDS algorithm (Section 1.3.1) to estimate the sensor positions. With this two-step process, we not only introduce a self calibrating mechanism for ultrasound tomography devices, but we also provide analytic upper bounds on the calibration error in the presence of noise and missing entries. The analysis enables us to also find bounds on the performance of OPTSPACE in the presence of structured missing entries.

Distributed Low-Complexity Sensor Localization

In Chapter 3, we consider the s-stress function shown in Section 1.3.3 and study its properties in great detail. By decomposing this cost function, we provide an algorithm based on alternate coordinate descent to minimize it. The proposed algorithm is implementable in a distributed fashion and we observe that in practice the number of iteration until convergence is linear with respect to the number of sensors. We show — through randomized experiments — that the proposed method converges to the global minimum of the s-stress in more than 99% of cases. This raises the question if all the minimizers of the s-stress function are global or not. Interestingly, no one has been able to find an analytic non-global minimizer of the s-stress so far. By proving several algebraic properties of the s-stress function, we reduce the question of existence of its non-global minimizers to a hypothesis. If the hypothesis is true, then the s-stress function has only global minimizers, otherwise it also has local minimizers. Thanks to the structure of the

chosen cost function, we later provide a toy example of designing an optimal English keyboard based on the average pair-wise distances of alphabet characters in English words.

Room Geometry Estimation and In-Room Localization

In Chapter 4, we address the interesting problem of hearing the shape of a room. Imagine that you are blindfolded inside an unknown room. How can you hear the shape of the room with a single finger snap? Using the image source model (assuming that the room is convex and polyhedral), we reformulate the problem of room geometry estimation as the problem of positioning image sources. We use the fact that echoes coming from an image source to four microphones can build a row of an EDM. Thanks to the rank property of EDMs (Theorem 1.2), we assign correct echoes to the image sources, i.e., we label the echoes. We prove that such assignment is unique almost surely. In the noisy case where the rank test is not reliable, we use the s -stress criterion (Section 1.3.3) with the algorithm proposed in Chapter 3 to assign scores to every echo combination. We provide experiments performed in a classroom on the EPFL campus and also in the Lausanne Cathedral. Using the same principles, we also provide a novel algorithm for indoor localization when the room geometry is known. Using this method, we can estimate the location of a microphone inside a room on the EPFL campus with 5 mm accuracy.

Microphone Array Calibration With Diffuse Noise

In Chapter 5, we aim at solving the problem of microphone array calibration in a diffuse noise field. We use a property of the diffuse noise fields that the coherence between recordings of two microphones is directly related to their mutual distances. This model, however, performs poorly when the microphones are separated by a large distance inside a room. Thus, using a Cadzow-like method we estimate the large distances in the corresponding EDM of the microphones. Through this process, we alternatively impose the rank property of EDMs (Theorem 1.2) and their other properties (symmetry, zero diagonal and non-negativity). This alternating procedure, although heuristic, provides a better approximation framework for noisy and incomplete EDMs. Using real experiments we compare the calibration results with different localization methods and show, for the first time, the possibility of calibrating large ad-hoc microphone arrays in a diffuse field.

Sequences With Minimal Time-Frequency Uncertainty

In Chapter 6, we study a central problem in signal processing and communications: design signals (or equivalently window functions or filters) that are compact both in time and frequency. The variance is accepted as a good measure of compactness, and with this definition Heisenberg states that a given function cannot be arbitrarily compact both in time and frequency, defining an “uncertainty” lower bound. In continuous-time, Gaussian functions reach this bound. For sequences, however, this is not true; it is known that the Heisenberg’s bound is generally unachievable. We study the proper choice of variance in the periodic frequency domain. Then, for a chosen frequency variance, we formulate the search for maximally compact sequences as an exactly and efficiently solved convex optimization problem, thus providing a sharp uncertainty principle for sequences. The optimization formulation also reveals that maximally compact sequences are derived from Mathieu’s harmonic cosine function of order zero. Using asymptotic behavior of Mathieu’s functions, we provide analytic uncertainty bounds in the asymptotical regimes.

Chapter 2

Sensor Calibration for Ultrasound Tomography Devices

Essentially, all models are wrong, but some are useful.

George Edward Pelham Box

2.1 Introduction

In most applications that involve sensing, finding the correct positions of the sensors is of crucial importance for obtaining reliable results. This is particularly true in the case of inverse problems which can be very sensitive to incorrect sensor placement. This requirement can be satisfied in two ways; We might put the effort in the construction of the instruments and try to place the sensors exactly in the desired positions, or use a method to find the exact positions after the construction of the device. In this chapter we consider the latter and we call the procedure of obtaining the sensor positions *calibration*. Note that even in the former case, due to the limited precision of the construction instruments, a calibration is needed afterwards to determine the exact sensor positions.

This chapter¹ focuses on the calibration problem in circular ultrasound tomography devices, in particular, the ones manufactured and deployed in [35, 65]. These devices consist of a circular ring surrounding an object and scanning horizontal planes. Ultrasound sensors are placed on the interior boundary of the ring and act both as transmitters and receivers. In such tomography devices, the sensors are not exactly placed on a perfect circle. This uncertainty in the positions

1. This chapter is the result of a collaboration with A. Karbasi, S. Oh and M. Vetterli [95].

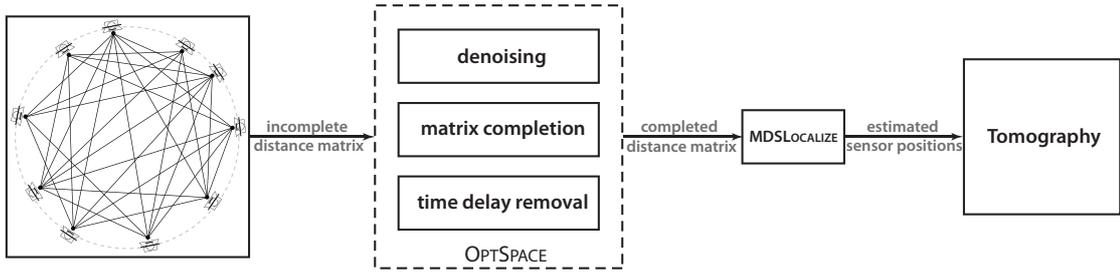


Figure 2.1: Block diagram for the calibration procedure prior to ultrasound tomography. The incomplete distance matrix is passed through the OPTSPACE algorithm which denoises it, estimates the missing entries and removes the unknown time delay. The calibration is finished then by applying the classic MDS algorithm on the completed matrix which estimates the actual sensor positions.

of the sensors acts as a source of error in the reconstruction algorithms that are used to obtain the characteristics of the enclosed object. We aim at finding a simple method for calibrating the system at low cost and without using any extra calibrating instrument.

We use the time-of-flight (ToF) between each pair of sensors to estimate their mutual distances. In a *homogeneous* medium where the sound speed is fixed, the ToFs between sensors have a simple relation to their mutual distances. If we have all the noiseless ToF measurements, we can use the classic MDS method, introduced in Section 1.3, to estimate the sensor positions. In practical settings, however, there are a number of challenges; these measurements are noisy, not all the measurements are available, and there is an unknown delay added to the ToF measurements. We categorize the missing measurements into two classes; *structured missing entries* caused by inability of the sensors to compute the ToF with their close-by neighbors, and *random missing entries* due to malfunctioning of the sensors and uncertainties of the measurement procedure (explained further later).

In general, it is a difficult task to infer missing entries of a matrix. However, it has recently been established that if the matrix is low rank, a small random subset of its entries permits an exact reconstruction [20]. We use the rank property of Euclidean distance matrices from Theorem 1.2 to show that an element-wise transform of the ToF matrix has low rank. Thus, its missing entries can be accurately estimated using matrix completion algorithms. We use OPTSPACE, a robust matrix completion algorithm developed by Keshavan et al. [68] to estimate the missing entries of this matrix. After completing the ToF matrix, we use the classic MDS algorithm (MDSLOCALIZE) shown in Algorithm 1.1, to estimate the position of the sensors.

Besides providing a complete framework for self-calibration of circular ultrasound tomography devices, one of the main contributions of this chapter is the theoretical bound on the calibration error. Such results, as we see in the next section, are not only valid for ultrasound tomography devices, but also benefit the analysis of sensor localization algorithms for which local connectivity information is missing.

We also present a heuristic algorithm to estimate the unknown time delay, which is due to the unknown piezoelectric impulse response and lack of the time-origin in the measurements. The block diagram shown in Figure 2.1 summarizes the calibration procedure taking place prior to tomography.

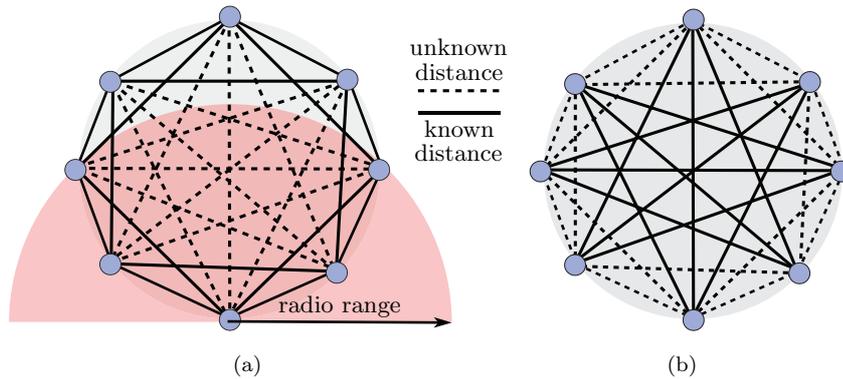


Figure 2.2: Calibration with missing entries (a) in sensor localization local connectivity information is available. (b) in calibration the opposite is true.

2.2 Calibration as a Dual to Sensor Localization

Calibration for circular tomography devices is a variant of sensor localization [21, 55]. In the classic sensor localization, given the *local connectivity*, the objective is to infer the global position of the sensors [79, 91, 112]. In practice, each node in the sensor network has a small communication range compared to the field size they are installed. This situation is depicted in Figure 2.2(a).

In our problem, however, the local connectivity is precisely the kind of information that is missing (see Section 2.3.2). This situation is demonstrated in Figure 2.2(b). By comparing these two scenarios, one can think of the calibration problem for ultrasound sensors as the *dual* problem of sensor localization. As a result, all sensor localization algorithms that rely on local information/connectivity are doomed to fail in our scenario. To confirm this fact, in Section 2.7 through numerical simulations we compare the performance of our proposed method with the state-of-the-art algorithms for sensor localization applied in our setting.

The first sensor localization algorithm we consider is MDS-MAP [115]. In this method the distance of sensors that are not in each others vicinity is approximated by the shortest path between them. We can easily see that given the distances of faraway sensors, the shortest path is a very coarse estimate of the distance between the close-by sensors. This makes MDS-MAP perform poorly in our setting. Further, we compare our results to one of the most prominent algorithms for centralized sensor localization, based on semi-definite programming (SDP). The main problem of SDP-based methods is their heavy computational load. According to [13], the sensor localization for more than 200 sensors is computationally prohibitive. Theoretical guarantees of such methods were provided recently by Javanmard et al. [60].

In the core of our proposed method is matrix completion (explained in Section 2.4.1). Based on the rank property of EDMs, Drineas et al. suggest using matrix completion for inferring the unknown distances [34] in the distance-squared matrix. However, their analysis relies on the assumption that even for faraway sensors, there is a nonzero probability of communication. In our setting, this assumption implies that the pairwise distances of nearby transmitters/receivers can be obtained with a nonzero probability, which does not hold.

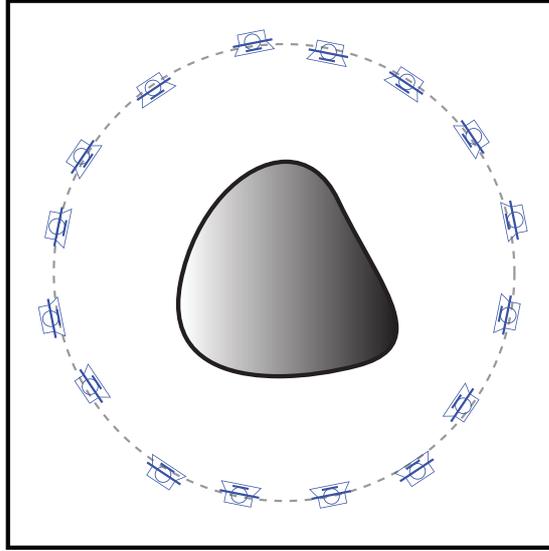


Figure 2.3: Circular setup for ultrasound tomography considered in this chapter. Ultrasound transducers are distributed on the edge of a circular ring and the object with unknown characteristics is put inside.

In a series of papers [67–69], Keshavan et al. study an efficient implementation of a matrix completion algorithm called OPTSPACE. We show that OPTSPACE is also capable of finding the missing nearby distances in our scenario and hence provide us with their corresponding ToFs. To the best of our knowledge, all the above work, as well as the recent matrix completion algorithms [104, 105], only deal with *random* missing entries. However, in our case, we have *structured* missing entries in addition to random ones (see Section 2.3.2), an aspect that was absent from the previous work. Therefore, one of our contributions is to provide analytic bounds on the error of OPTSPACE in the presence of structured missing entries.

2.3 Circular Time of Flight Tomography

In a circular ultrasound tomography setup, n ultrasound transmitters and receivers are installed on the interior edge of a circular ring and an object with unknown acoustic characteristics is placed inside the ring. The general configuration for such a tomography device is depicted in Figure 2.3. At each time instance a transmitter is fired, sending ultrasound signals with frequencies ranging from hundreds to thousands of kHz, while the rest of the sensors record the received signals. The same process is repeated for all the transmitters. Each one of the n sensors on the ring is capable of transmitting and receiving ultrasound signals. By employing these measurements, an inverse problem is constructed, whose solution provides the acoustic characteristics of the enclosed object (e.g., sound speed, sound attenuation, etc.). In order to solve the inverse problem, a very precise estimate of the sensor positions is needed. In most applications (e.g., [64, 90]) it is assumed that the sensors are positioned equidistant apart on a circle and no later calibration is performed to find the exact sensor positions.

2.3.1 Homogeneous Medium and Dimensionality Reduction

Assume that the squared mutual ToFs are stored in a matrix \mathbf{T} . In a *homogeneous* medium, entries of \mathbf{T} represent the time travelled by sound in a straight line between each pair of a transmitter and receiver.

Knowing the temperature and the characteristics of the medium inside the ring, we can accurately estimate the constant sound speed c_0 . Thus, it is reasonable to assume that c_0 is fixed and known. We can construct a Euclidean distance matrix \mathbf{D} consisting of squares of the mutual distances between the sensors as

$$\mathbf{D} = [d_{i,j}^2] = c_0^2 \mathbf{T}, \quad \mathbf{T} = [t_{i,j}^2], \quad i, j \in \{1, \dots, n\}$$

where $t_{i,j}$ is the ToF between sensors i and j and n is the total number of sensors around the circular ring. Notice that the only difference between the ToF matrix \mathbf{T} , and distance matrix \mathbf{D} , is the constant c_0 . Thus in the following, we focus mainly on the distance matrix rather than the actual measured matrix \mathbf{T} .

The matrix \mathbf{T} is a symmetric matrix with zeros on the diagonal and so is the matrix \mathbf{D} . We saw in Theorem 1.2 that the rank of an EDM does not depend on the number of the sensors, but rather on the dimension of the space that the sensors span. We assume that the sensors lie on a 2-dimensional ring. Thus, the EDM \mathbf{D} has rank at most 4 (We assume that $\eta = 2$). According to Theorem 1.2, if the sensors were on a perfect circle, the rank of the matrix would be 3.

2.3.2 Time-of-Flight Estimation

Several methods for ToF estimation (also known as time-delay estimation in acoustics [23]) are proposed in the signal processing community [63, 78]. Normally the received signal is compared to a reference signal (ideally the sent signal), and the relative delay between the two signals is estimated. As the sent signal is not available in most cases (due to unknown impulse response of the transducers), the received signal through the object is compared to the received signal when the underlying medium is homogeneous. This assumption is not true in our case. In the calibration phase, we have only signals passed through the homogeneous medium. Thus, there is no reference signal to find the relative time-of-flights.

Because of the above limitations, we are forced to estimate the *absolute* ToFs. For this purpose, we use the first arrival method [138] to estimate the ToFs.

In practical screening systems, to record measurements for one fired transmitter, all the sensors are turned on simultaneously and after some unknown transition time (which is caused by the system structure, different sensor responses, etc.), the transmitter is fed with the electrical signal and the receivers start recording the signal. This unknown time may change for each pair of transmitters and receivers. We will see that this unknown time delay plays an important role in calibration.

The beam width of the transducers and the transition behavior of the ultrasound sensors prevent the sensors to have a reliable ToF measurement for close-by neighbors. This causes incorrect ToF values for the sensors positioned close to each other. Therefore, numbering the sensors on the ring by their angles from 1 to n , in the ToF matrix \mathbf{T} , there are no measurements on a certain band around the main diagonal and on the lower left and upper right parts as well. We call these missing entries as *structured missing entries*. This is illustrated in Figure 2.4. The

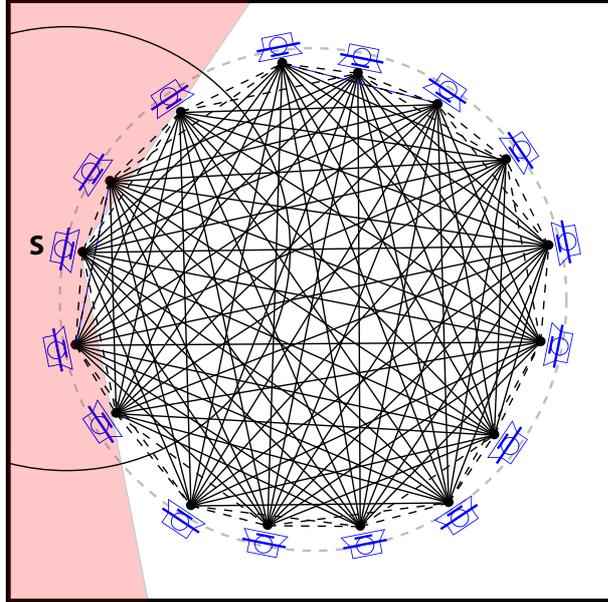


Figure 2.4: The beam width of the transmitter prohibits the neighboring sensors to have reliable ToF measurements. This is shown by dashed lines and results in the structured missing entries.

links shown by dashed lines do not contribute to the ToF measurements, because the beam for the transmitter does not cover the red part.

Compared to other measuring sensors such as X-ray and electromagnetic imaging devices, ultrasound sensors are more prone to malfunctions due to continuous contact with the measurement liquid (water, gel or oil). Moreover, as these devices are inside liquid, a momentary presence of an air bubble can cause an error in the sensor measurements. Studies also highlight the possibility of malfunction of ultrasound sensors [109]. Thus, during the measurement procedure, it might happen that some sensors produce outliers. One can perform a post processing on the measurements, in which a smoothness criterion is defined and the measurements not satisfying this criterion are removed from the ToF matrix. We call these entries *random missing entries*. An instance of the ToF matrix with the structured and random missing entries is shown in Figure 2.5, where the gray entries correspond to the missing entries.

Furthermore, in practice, the measurements are corrupted by noise. Thus, we have an incomplete and noisy matrix \mathbf{T} , which cannot be used for position reconstruction, unless the time delay effect is removed, the unknown entries are estimated, and the noise is smoothed.

2.4 Background

2.4.1 Matrix Completion

OPTSPACE, introduced in [68], is an algorithm for recovering a low-rank matrix from noisy data with missing entries. The steps are shown in Algorithm 2.1. Let \mathbf{M} be a rank- q matrix of

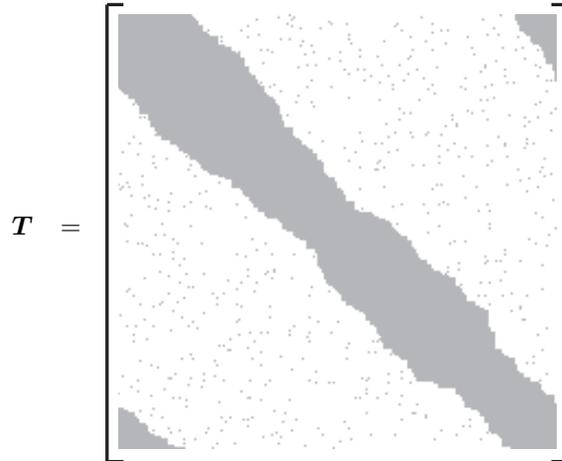


Figure 2.5: An example of an incomplete ToF matrix with structured and random missing entries. Missing entries are colored in gray.

dimensions $n \times n$, \mathbf{Z} the measurement noise, and E the set of indices of the measured entries. Then, the measured noisy and incomplete matrix is $\mathbf{M}^E = \mathcal{P}_E(\mathbf{M} + \mathbf{Z})$.

Algorithm 2.1 OPTSPACE [68]

Input: Observed matrix $\mathbf{M}^E = \mathcal{P}_E(\mathbf{M} + \mathbf{Z})$.

Output: Estimate \mathbf{M} .

1. Trimming: remove over-represented columns/rows;
 2. Rank- q projection on the space of rank- q matrices according to (2.1);
 3. Gradient descent: Minimize a cost function $F(\cdot)$ defined in (2.2);
-

A row or a column is over-represented if it contains more samples than twice the average number of samples per row or column. These rows or columns can dominate the spectral characteristics of the observed matrix \mathbf{M}^E . Thus, some of their entries are removed uniformly at random from the observed matrix. Let $\widetilde{\mathbf{M}}^E$ be the resulting matrix of this trimming step. The trimming step is presented here for completeness, but in the case when p (refer to Section 2.5) is larger than some fixed constant (like in our case), $\mathbf{M}^E = \widetilde{\mathbf{M}}^E$ with high probability and the trimming step can be omitted.

In the second step, we compute the singular value decomposition of $\widetilde{\mathbf{M}}^E$.

$$\widetilde{\mathbf{M}}^E = \sum_{i=1}^n \sigma_i(\widetilde{\mathbf{M}}^E) u_i v_i^T,$$

where $\sigma_i(\cdot)$ denotes the i -th singular value of a matrix. Then, the rank- q projection returns the matrix

$$\mathcal{P}_q(\widetilde{\mathbf{M}}^E) = (1/p_n) \sum_{i=1}^q \sigma_i(\widetilde{\mathbf{M}}^E) u_i v_i^T, \quad (2.1)$$

obtained by setting to 0 all but the q largest singular values.

Starting from the initial guess provided by the rank- q projection $\mathcal{P}_q(\widetilde{\mathbf{M}}^E)$, the final step solves a minimization problem stated as the following [68]:

Given $\mathbf{X} \in \mathbb{R}^{n \times q}$, $\mathbf{Y} \in \mathbb{R}^{n \times q}$ with $\mathbf{X}^T \mathbf{X} = \mathbf{I}$ and $\mathbf{Y}^T \mathbf{Y} = \mathbf{I}$, define

$$F(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{S} \in \mathbb{R}^{q \times q}} \frac{1}{2} \sum_{(i,j) \in E} (M_{i,j} - (\mathbf{X} \mathbf{S} \mathbf{Y}^T)_{i,j})^2. \quad (2.2)$$

Values for \mathbf{X} and \mathbf{Y} are computed by minimizing $F(\mathbf{X}, \mathbf{Y})$. This consists of writing $\mathcal{P}_q(\widetilde{\mathbf{M}}^E) = \mathbf{X}_0 \mathbf{S}_0 \mathbf{Y}_0^T$ and minimizing $F(\mathbf{X}, \mathbf{Y})$ locally with initial condition $\mathbf{X} = \mathbf{X}_0$ and $\mathbf{Y} = \mathbf{Y}_0$. This last step tries to get us as close as possible to the correct low rank matrix \mathbf{M} .

2.4.2 Reconstruction Error Measure

As mentioned earlier, for calibration, we perform the classic MDS method (MDSLOCALIZE) to recover the sensor locations from the completed distance matrix. Recall from Section 1.2.1 that EDMs are invariant under rigid transformation. Thus, any method that tries to estimate the positions from mutual distances will also be invariant under rigid transformation. Strictly speaking, we only recover the configuration and not the absolute positions, in the sense that $\text{MDS}_\eta(\mathbf{D})$ is one version of infinitely many solutions that (approximately) match the distance measurements \mathbf{D} .

Let $\widehat{\mathbf{X}} \in \mathbb{R}^{n \times \eta}$ denote an estimate for \mathbf{X} . Then, we need to define a metric for the distance between the original position matrix \mathbf{X} and the estimation $\widehat{\mathbf{X}}$ which is invariant under rigid transformations of \mathbf{X} or $\widehat{\mathbf{X}}$.

Lemma 2.1 ([34, 93, 115])

Let matrix \mathbf{L} be as in (1.2). Also let \mathbf{X} and $\widehat{\mathbf{X}}$ be two position matrices with dimension $n \times \eta$. Then, we can show that

- $\mathbf{L} \mathbf{X} \mathbf{X}^T \mathbf{L}$ is invariant under rigid transformation.
- $\mathbf{L} \mathbf{X} \mathbf{X}^T \mathbf{L} = \mathbf{L} \widehat{\mathbf{X}} \widehat{\mathbf{X}}^T \mathbf{L}$ implies that \mathbf{X} and $\widehat{\mathbf{X}}$ are equal up to a rigid transformation.

This naturally defines the following distance between \mathbf{X} and $\widehat{\mathbf{X}}$.

$$\text{dist}(\mathbf{X}, \widehat{\mathbf{X}}) = \frac{1}{n} \left\| \mathbf{L} \mathbf{X} \mathbf{X}^T \mathbf{L} - \mathbf{L} \widehat{\mathbf{X}} \widehat{\mathbf{X}}^T \mathbf{L} \right\|_F, \quad (2.3)$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

According to Lemma 2.1, this distance is invariant to rigid transformation of \mathbf{X} and $\widehat{\mathbf{X}}$. Furthermore, $\text{dist}(\mathbf{X}, \widehat{\mathbf{X}}) = 0$ implies that \mathbf{X} and $\widehat{\mathbf{X}}$ are equal up to a rigid transformation. We later state our theoretical results in terms of the distance defined in (2.3).

2.5 Mathematical Formulation

For simplicity, we will assume that the unknown time delay is constant for all the transmitters. Specifically, all the transmitters send the electrical signal after some fixed but unknown delay t_0 . Hence, we can rewrite the distance matrix as

$$\sqrt{\widetilde{\mathbf{D}}} = \sqrt{\mathbf{D}} + d_0 \mathbf{J} + \mathbf{Z}, \quad (2.4)$$

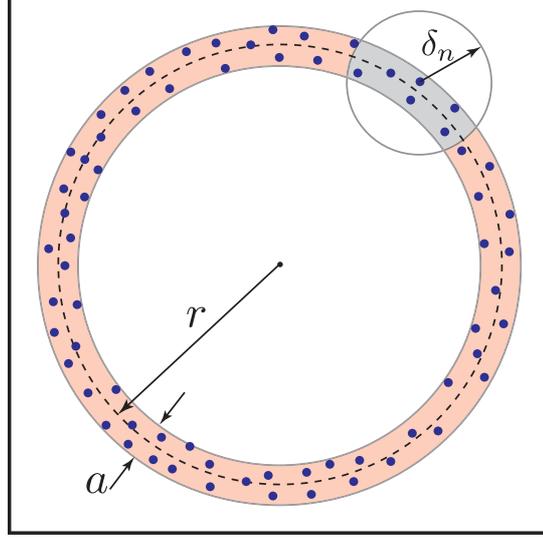


Figure 2.6: Sensors are distributed around a circle of radius r with small deviations from the circumference.

where $\mathbf{D} = c_0^2 \mathbf{T}$, $d_0 = c_0 t_0$, $\mathbf{Z} = c_0 \mathbf{Z}_0$ and

$$\mathbf{J}[i, j] = \begin{cases} 1 & \text{if } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

The matrix \mathbf{Z}_0 models the noise in the ToF measurements. In practice, the sensor positions deviate from the circumference of a ring and our ultimate goal is to estimate these deviations or equivalently the correct positions (see Figure 2.6). The general positions taken by sensors are denoted by the set of vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$.

As described earlier, there are two types of missing entries; structured and random. To model the structured missing entries, we assume that any measurement between sensors of distance less than δ_n is missing (see Figure 2.6). Hence, the number of structured missing entries depends on δ_n^2 . We are interested in the regime where we have a small number of structured missing entries per row in the large system limit. Accordingly, a typical range for δ_n is $\delta_n = \Theta(r\sqrt{\log n/n})$. A random set of structured missing indices $S \subseteq [n] \times [n]$ is defined from $\{\mathbf{x}_i\}$ and δ_n , by

$$S = \{(i, j) : d_{i,j} \leq \delta_n \text{ and } i \neq j\},$$

where $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$. Then, the structured missing entries are denoted by a matrix

$$\mathbf{D}^s[i, j] = \begin{cases} \mathbf{D}[i, j] & \text{if } (i, j) \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the matrix $\mathbf{D}^{\bar{s}} = \mathbf{D} - \mathbf{D}^s$ captures the noiseless distance measurements that are not affected by structured missing entries. This way, we can interpret the matrix \mathbf{D}^s as additive

noise in our model. Likewise, for the constant additive time delay we can define

$$\mathbf{J}^{\bar{s}}[i, j] = \begin{cases} \mathbf{J}[i, j] & \text{if } (i, j) \in S^\perp, \\ 0 & \text{otherwise,} \end{cases}$$

where S^\perp denotes the complementary set of S . Next, to model the noise we add a random noise matrix $\mathbf{Z}^{\bar{s}}$.

$$\mathbf{Z}^{\bar{s}}[i, j] = \begin{cases} \mathbf{Z}[i, j] & \text{if } (i, j) \in S^\perp, \\ 0 & \text{otherwise.} \end{cases}$$

We do not assume a prior distribution on \mathbf{Z} , and the main theorem is stated for any general noise matrix \mathbf{Z} , deterministic or random.

Finally, to model the random missing entries, we assume that each entry of $\sqrt{\mathbf{D}^{\bar{s}}} + d_0\mathbf{J}^{\bar{s}} + \mathbf{Z}^{\bar{s}}$ is sampled with probability $p < 1$. Let $E \subseteq [n] \times [n]$ denote the subset of indices which are not erased randomly. Then a projection operator $\mathcal{P}_E : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ is defined as

$$\mathcal{P}_E(\mathbf{M})[i, j] = \begin{cases} \mathbf{M}[i, j] & \text{if } (i, j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

We denote the observed measurement matrix by

$$\mathbf{N}^E = \mathcal{P}_E \left((\sqrt{\mathbf{D}^{\bar{s}}} + d_0\mathbf{J}^{\bar{s}} + \mathbf{Z}^{\bar{s}})^2 \right),$$

where $(\cdot)^2$ represents element-wise squaring. Notice that the matrix \mathbf{N}^E has the same shape as \mathbf{T} shown schematically in Figure 2.5. Now we can state the goal of our calibration problem:

Given the observed matrix \mathbf{N}^E and the missing indices $S \cup E^\perp$, we want to estimate a matrix $\hat{\mathbf{D}}$ which is close to the correct distance matrix \mathbf{D} . Then by using $\hat{\mathbf{D}}$ we would like to estimate the sensor positions.

In order to achieve this goal, there are two obstacles we need to overcome. First, we need to estimate the missing entries of \mathbf{N}^E and second, we want to find the sensor positions given approximate pairwise distances. The former is done by employing the matrix completion algorithm OPTSPACE [69] and the latter by using MDSLICALIZE, in Algorithm 1.1 [34].

2.6 Calibration Algorithm and Theoretical Bounds

We mentioned earlier that the OPTSPACE algorithm is not directly applicable to the squared distance matrix because of the unknown delay. Since \mathbf{J} in (2.4) is a full rank matrix, the matrix $\tilde{\mathbf{D}}$ no longer has rank four. Moreover, as the measurements are noisy, one cannot hope for estimating the exact value for d_0 . Therefore, in the following we will provide error bounds on the reconstruction of the positions assuming that the time delay (equivalently d_0) is known. Afterwards, a heuristic method is proposed to estimate the value of d_0 .

Algorithm 2.2 Finding d_0 .**Input:** Matrix \mathbf{N}^E ;**Output:** Estimate d_0 ;

1. Construct the candidate set $\mathcal{C}_d = \{d_0^{(1)}, \dots, d_0^{(M)}\}$ containing discrete values for d_0 .
2. **for** $k = 1$ to M **do**
3. Set $\sqrt{\mathbf{N}_{(k)}^E} = \sqrt{\mathbf{N}^E} - d_0^{(k)} \mathbf{J}^E$;
4. Apply OPTSPACE on $\mathbf{N}_{(k)}^E$ and call the output $\widehat{\mathbf{N}}^{(k)}$;
5. Apply MDSLICALIZE and let $\mathbf{X}^{(k)} = \text{MDS}_2(\widehat{\mathbf{N}}^{(k)})$;
6. Find $c^{(k)}$

$$c^{(k)} = \sum_{(i,j) \in E \cap S^\perp} (d_0^{(k)} + \|\mathbf{X}_i^{(k)} - \mathbf{X}_j^{(k)}\| - \sqrt{\mathbf{N}^E[i,j]})^2$$
;
7. **end for**
8. Choose d_0 satisfying

$$d_0 = d_0^{(l)}, \quad l = \arg \min_k c^{(k)}$$
;

Theorem 2.1

Assume n sensors are distributed independently and uniformly at random on a circular ring of width a with central radius r_0 as in Figure 2.4. The resulting distance matrix \mathbf{D} is corrupted by structured missing entries \mathbf{D}^s and measurement noise \mathbf{Z}^s . Further, the entries are missing randomly with probability p . Let \mathbf{N}^E denote the observed distance-squared matrix. Assume $\delta_n = \delta r_0 \sqrt{\log n/n}$. Then, there exist constants C_1 and C_2 , such that the output of OPTSPACE $\widehat{\mathbf{D}}$ achieves

$$\frac{1}{n} \|\mathbf{D} - \widehat{\mathbf{D}}\|_F \leq C_1 \left(\sqrt{\frac{\log n}{n}} \right)^3 + C_2 \frac{\|\mathcal{P}_E(\mathbf{Y}^s)\|_2}{pn}, \quad (2.5)$$

with probability larger than $1 - n^{-3}$, provided that the right hand side is less than $\sigma_4(\mathbf{D})/n$. In (2.5), $\mathbf{Y}^s[i,j] = \mathbf{Z}^s[i,j]^2 + 2\mathbf{Z}^s[i,j]\sqrt{\mathbf{D}^s[i,j]}$.

The proof is given in Appendix 2.A.1. The above theorem, in great generality, holds for any noise matrix \mathbf{Z} , deterministic or random. The above guarantees only hold up to numerical constants. To evaluate the performance of this approach in practice we provide simulation results in Section 2.7.

Theorem 2.2

Applying the MDSLICALIZE algorithm on $\widehat{\mathbf{D}}$, the error on the resulting coordinates will be bounded as follows

$$\text{dist}(\mathbf{X}, \widehat{\mathbf{X}}) \leq C_1 \left(\sqrt{\frac{\log n}{n}} \right)^3 + C_2 \frac{\|\mathcal{P}_E(\mathbf{Y}^s)\|_2}{pn},$$

with probability larger than $1 - 1/n^3$. The proof is given in Appendix 2.A.2

Finally we can use a heuristic algorithm to estimate the unknown value of d_0 . It simply checks for which value of d_0 , the reconstructed positions produce a distance matrix closest to the recorded measurements when the effect of d_0 is removed. This is shown in Algorithm 2.2.

In fact, this algorithm guarantees that after removing the effect of the time delay, we have found the best rank 4 approximation of the distance squared matrix. In other words, if we remove exactly the mismatch d_0 , we will have an incomplete version of a rank 4 matrix and after reconstruction, the measured values will be closest to the reconstructed ones.

2.6.1 Computational Complexity

Note that the calibration phase can be performed in the resting stage of the device; it does not introduce any difficulties or extra procedures to the functionalities of the device. It is also easy to compute the complexity of the calibration phase. We first need to measure the time of flights. This step is also performed in the actual ultrasound tomography and does not add computational complexity compared to real tomography. Then, we need to complete the distance matrix. The complexity of this step is $O(|E| \log n)$, where $|E|$ is the number of observed entries and n is the number of sensors [67]. After the distance matrix is complete, one needs to perform the MDSLICALIZE algorithm. This step requires a singular value decomposition which has a complexity of $O(n^3)$.

2.7 Numerical Evaluations

In order to evaluate the functionality of the calibration process, three sets of experiments are conducted. First, the distance matrix is assumed to be noiseless and the value of d_0 is set to zero. The position estimation error is derived for different values of n and the ring width a . The value of r_0 is set to 10 cm, on average 5 percent of entries are missing randomly, and δ in Theorem 2.1 is assumed to be 1. For each value of a and n , the experiment is repeated 10 times, and the average error is taken. The results are reported in Figure 2.7. As expected from Theorem 2.2, the general trend in all the curves is that the error decreases as n grows. Moreover, the larger a , the bigger the reconstruction error, which is also coherent with the results of Theorem 2.2.

To examine the stability of the estimation algorithm under noise, under the same settings as in the previous experiment, we added to each entry of the matrix \sqrt{D} a centered white Gaussian noise of different standard deviations. For each n and standard deviation of the noise, the experiments are repeated 10 times and the average is taken. The results are depicted in Figure 2.8. As the variance of the noise increases, the position estimation error grows, but in general the error decreases for larger n .

As we discussed in Section 2.2, one might treat the calibration problem as a special case of the sensor localization problem. In order to compare the performance of sensor localization methods based on local information with the proposed methods, a set of simulations are performed. We compared the localization results of our method to the ones of MDS-MAP [115], SDP-based [13] and also SVD-RECONSTRUCT [34]. Figure 2.9 demonstrates the position reconstruction error (defined in (2.3)) of our algorithm compared to these methods for different number of sensors, n .

In these simulations, we set the values of a to 1 cm, δ to 1, r_0 to 10 cm, t_0 to zero, and the percentage of the random missing entries to 5. The distance measurements are corrupted with a white Gaussian noise of standard deviation 0.6 mm. For each method and each n , the experiment is performed 10 times for different positions and different noises, and the average error is taken. For the SDP-based method, we use the algorithm presented in [13] and the code published by the same authors. For MDS-MAP, we estimate the shortest paths using Johnson's

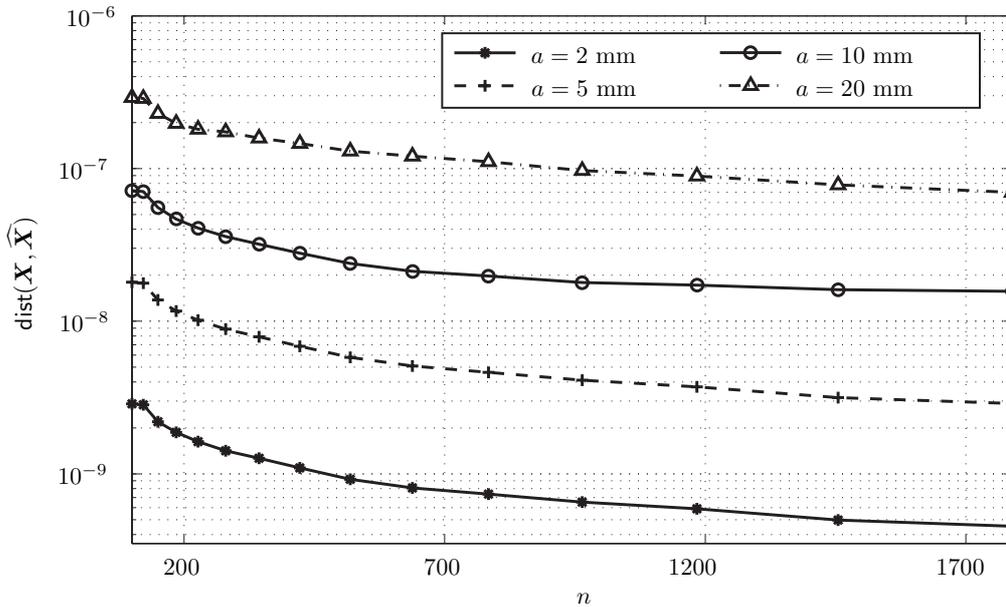


Figure 2.7: Error in position estimation in the noiseless case for different values of a . As n increases, the reconstruction error tends to zero. The estimation error increases for larger values of a , which confirms the results of Theorem 2.2.

algorithm [62]. Finally for SVD-RECONSTRUCT, we use the algorithm in [34]. In order to adapt the measurements with the assumptions of the method, we assumed that $p_{ij} = 1 - 0.05 = 0.95$ for the measured points (note that 0.05 is on average the probability of having a random missing entry) and $\gamma_{ij} = 0$.

As the results in Figure 2.9 suggest, MDS-MAP and SVD-RECONSTRUCT perform poorer compared to the other two methods. The poor performance of MDS-MAP is for the fact that it highly relies on the presence of local distance information, whereas in our case, these measurements are in fact missing. Also note that as the simulation results show, the estimation error might increase as n grows.

For SVD-RECONSTRUCT, the unrealistic assumption that all the sensors have a non-zero probability of being connected causes the bad results of the method. In our case, the probability that the close-by sensors are connected is zero because of the structured missing entries. In fact, since p_{ij} is high, one could see this method as simply applying the classic MDS on the incomplete distance matrix. The surprising observation about the performance of this method is that the estimation error does not change much with n .

In contrast to the two aforementioned algorithms, the SDP-based method performs very well for estimating the sensor positions and the reconstruction error is very close to the one of the proposed method. This is due to the fact that this method does not directly rely on the local distance information. As the number of sensors grows, however, the number of constraints for the semi-definite program grow, which causes the algorithm to fail for n larger than 150 in our case. The same limitation is also reported by the authors of the method.

In summary, taking the computational cost and reconstruction accuracy of the algorithms

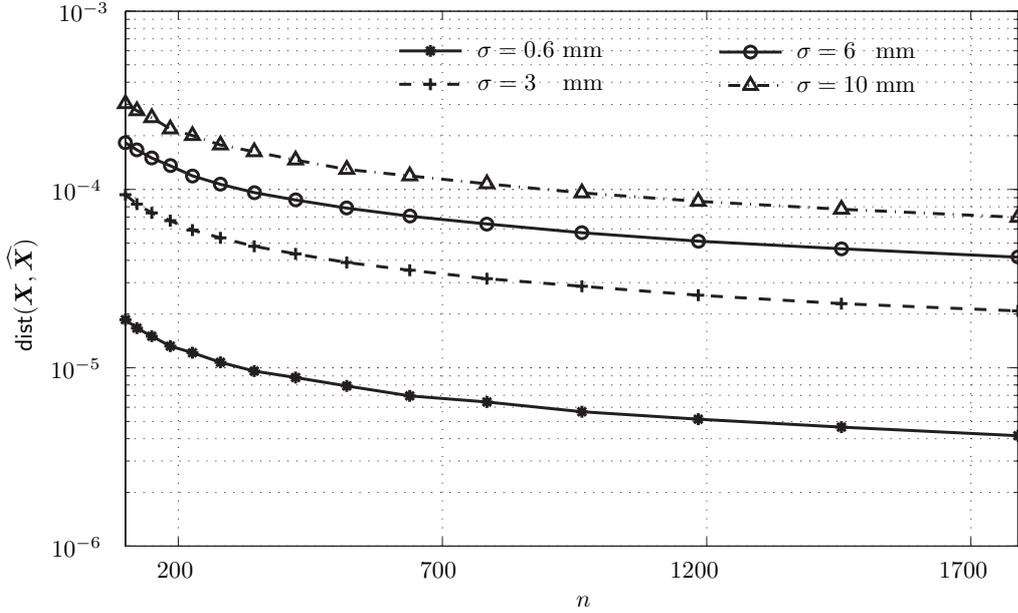


Figure 2.8: Error in position estimation for the case with centered white Gaussian noise of different standard deviations, σ .

into account, the proposed method performs significantly better.

Moreover, to show the importance of calibration in an ultrasound scanning device, a simple simulation is performed. Early diagnosis of breast cancer requires sub-millimeter precision in the imaging device. This implicitly poses restrictions on the calibration of the measuring device. If the ToF measurements correspond to the exact positions of sensors without time delay t_0 , reconstruction of *water* will lead to a homogeneous region with values equal to the water sound speed, whereas wrong assumptions on the sensor positions and t_0 cause the inverse method to give incorrect values of the sound speed to compensate the effect of position mismatch and time delay.

In a simple experiment, we simulated the reconstruction of water sound speed ($c_0 = 1500 \text{ m/s}$) using the ToF measurements. In this setup, 200 sensors are distributed around a circle with radius $r_0 = 10 \text{ cm}$, and they deviate at most 5 mm from the circumference. Also, the ToF measurements are added by $t_0 = 10 \mu\text{s}$. The incomplete distance matrix is shown in Figure 2.10(a). The value for t_0 is estimated as $10 \mu\text{s}$ using the proposed heuristic method, which is exactly as set in the simulation. The output of OPTSPACE algorithm is the completed matrix \hat{D} , which is shown in Figure 2.10(b).

Using the completed distance matrix and the MDS method, the positions are reconstructed and fed to an inverse tomography algorithm to reconstruct the water sound speed. The results of the reconstruction are shown in Figure 2.11. In Figure 2.11(a), the ToF matrix is not complete, it contains the time delay t_0 , and the positions are not calibrated. The dark ring is caused by the non-zero time delay in the ToF measurements. In Figure 2.11(b), the time mismatch is resolved using the proposed algorithm, but the sensor positions are not calibrated and the ToF matrix is

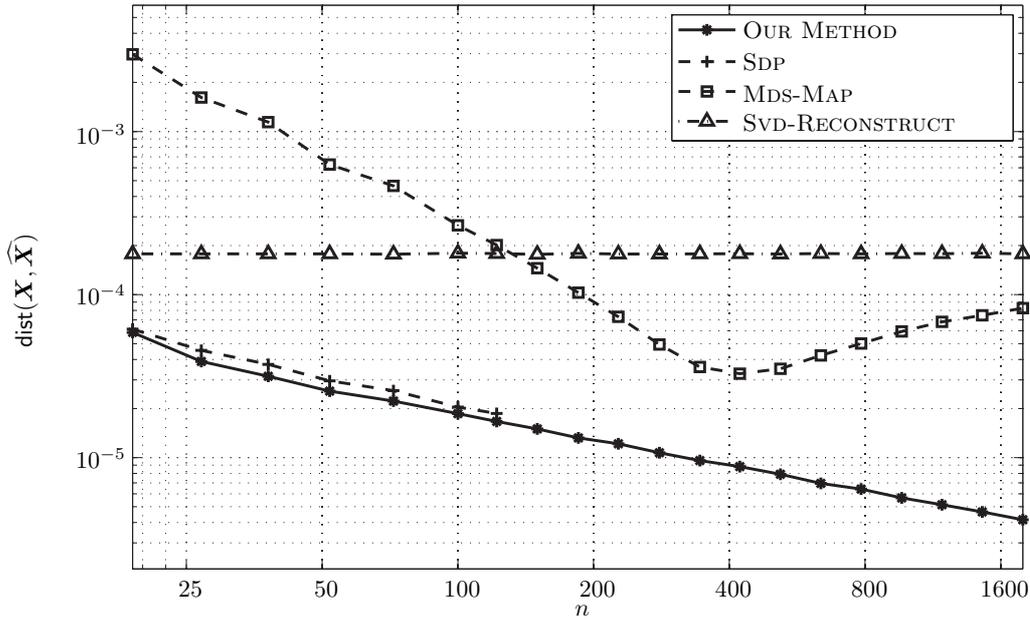


Figure 2.9: Error in position estimation versus the number of sensors for different methods.

still not complete. This figure shows clearly that finding the unknown time delay improves the reconstruction image significantly. Figure 2.11(c), shows the reconstructed medium when the ToF matrix is completed and time mismatch is removed, but the sensor positions are not yet calibrated. From this figure, it is confirmed that accurate time-of-flights are necessary but not sufficient to have a good reconstruction of the inclosed object. Finally, Figure 2.11(d) shows the reconstruction when the positions are also calibrated. Notice the change in the dynamic range of the figure for the last case.

2.8 Summary

We started the chapter by introducing a common problem in sensing devices. We focused on specific kind of measuring devices with ultrasound sensors on a circular ring. We used the rank property of EDMs to propose an algorithm for calibrating the sensor positions in presence of several uncertainties in the measurement process. Using the analysis of matrix completion literature, we were able to provide theoretical bounds on the reconstruction error of the sensor positions. We also tested our method through extensive simulations to demonstrate its functionality in practice. We compared the algorithm with the state-of-the-art centralized sensor localization methods and showed that our method outperforms them in estimating the correct sensor positions.

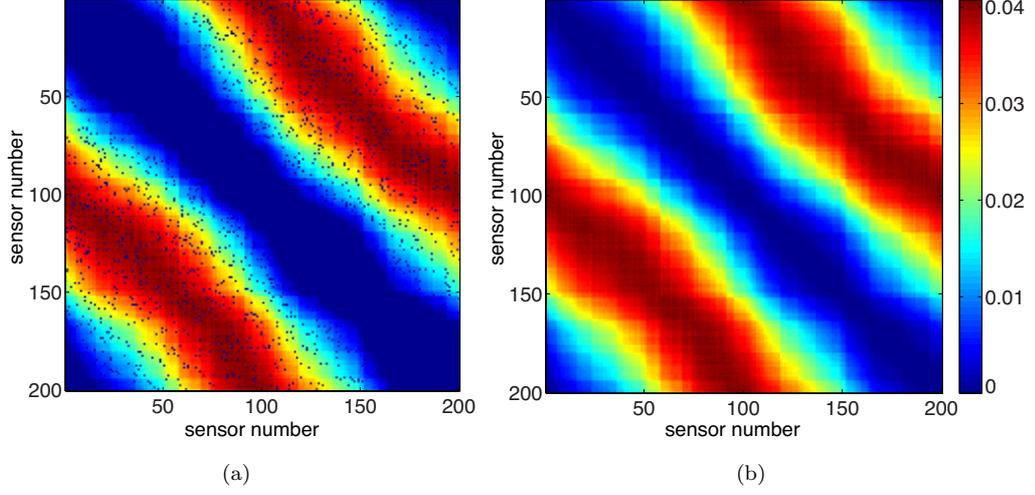


Figure 2.10: Input and output of OPTSPACE algorithm. Colors represent the value of the squared distances. (a) The incomplete distance squared matrix \mathbf{D} , with 5 percent of entries randomly missing, $t_0 = 10 \mu s$ and $\delta_n = 3 cm$. (b) The completed matrix with estimated $t_0 = 10 \mu s$.

2.A Analysis

2.A.1 Proof of Theorem 2.1

In order to prove our main result, we apply Theorem 1.2 of [67] to the rank-4 matrix \mathbf{D} and the observed matrix \mathbf{N}^E .

First, let us recall the *incoherence* property of \mathbf{D} from [67]. A rank-4 symmetric matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ is said to be μ -*incoherent* if the following conditions hold. Let $\mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$ be the singular value decomposition of \mathbf{D} .

A0. For all $i \in [n]$, we have $\sum_{k=1}^4 \mathbf{U}[i, k]^2 \leq 4\mu/n$.

A1. For all $i \in [n], j \in [n]$, we have $|\mathbf{D}[i, j]/\sigma_1(\mathbf{D})| \leq \sqrt{4\mu/n}$.

The extra $1/n$ terms in the right hand side are due to the fact that here, we assume that the singular vectors are normalized to unit norm, whereas in [67] the singular vectors are normalized to have norm \sqrt{n} .

Theorem 1.2 of [67] states that if a rank-4 matrix \mathbf{D} is μ -*incoherent* then the following is true with probability at least $1 - 1/n^3$. Let $\sigma_i(\mathbf{D})$ be the i th singular value of \mathbf{D} and $\kappa(\mathbf{D}) = \sigma_1(\mathbf{D})/\sigma_4(\mathbf{D})$ be the condition number of \mathbf{D} . Also, let $\hat{\mathbf{D}}$ denote the estimation returned by OPTSPACE with input \mathbf{N}^E . Then, there exists numerical constants C_1 and C_2 such that

$$\frac{1}{n} \left\| \mathbf{D} - \hat{\mathbf{D}} \right\|_F \leq C_1 \frac{\|\mathcal{P}_E(\mathbf{D}^s)\|_2 + \|\mathcal{P}_E(\mathbf{Y}^s)\|_2}{pn}, \quad (2.6)$$

provided that

$$np \geq C_2 \mu^2 \kappa(\mathbf{D})^6 \log n, \quad (2.7)$$

and

$$C_1 \frac{\|\mathcal{P}_E(\mathbf{D}^s)\|_2 + \|\mathcal{P}_E(\mathbf{Y}^s)\|_2}{pn} \leq \frac{\sigma_4(\mathbf{D})}{n}. \quad (2.8)$$

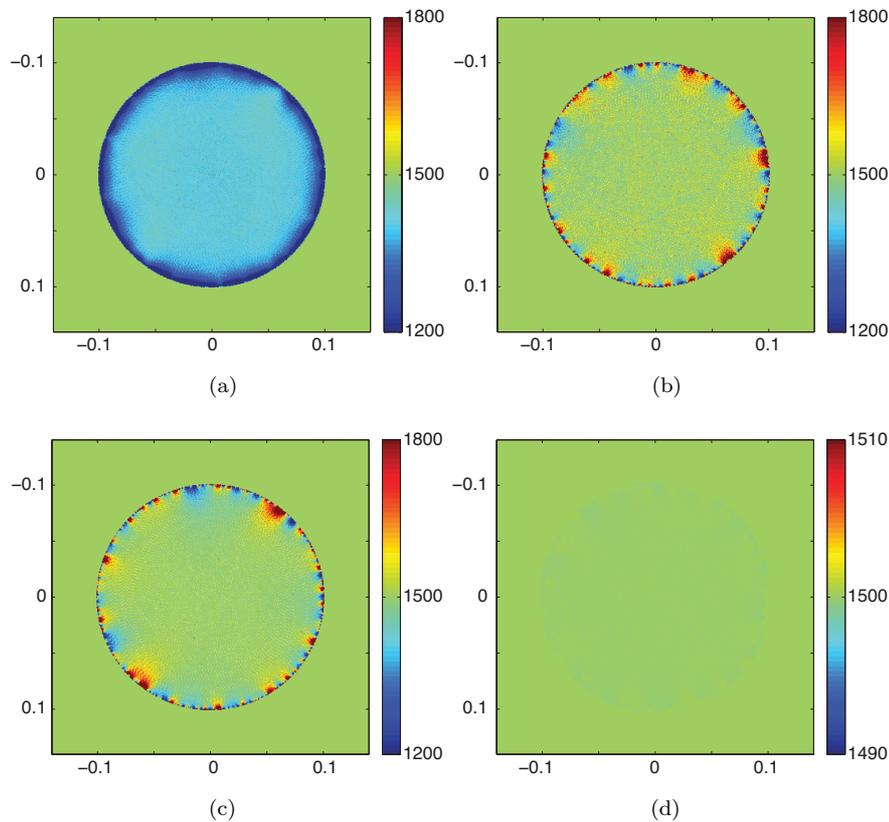


Figure 2.11: Results of the inversion procedure for finding the sound speed inside the ring with only water inside. (a) Reconstruction when no calibration is performed. (b) Reconstruction after t_0 is removed from the ToF matrix, but the matrix is still incomplete and the positions are not calibrated. (c) Reconstruction when the matrix is also completed, but the positions are not yet calibrated. (d) Reconstruction with completed ToF matrix and calibrated positions.

First, using Lemma 2.2, we show that the bound in (2.6) gives the desired bound in the theorem. Then, it is enough to show that there exists a numerical constant N such that the conditions in (2.7) and (2.8) are satisfied with high probability for $n \geq N$.

Lemma 2.2

In the model defined in Section 2.5, n sensors are distributed independently and uniformly at random on a circular ring of width a with central radius r_0 . Then, with probability larger than $1 - n^{-3}$, there exists a constant c such that

$$\|\mathcal{P}_E(\mathbf{D}^s)\|_2 \leq c\delta^3(r_0 + a)^2 \left(\sqrt{\frac{\log n}{n}}\right)^3 pn.$$

The proof of this lemma can be found in Appendix 2.A.3.

Now, to show that (2.7) holds with high probability for $n \geq C \log n/p$ for some constant C , we show that $\kappa \leq f_\kappa(r_0, a)$ and $\mu \leq f_\mu(r_0, a)$ with high probability, where f_κ and f_μ are independent of n . Recall that $\kappa(\mathbf{D}) = \sigma_1(\mathbf{D})/\sigma_4(\mathbf{D})$. We have

$$\begin{aligned} \mathbf{D}[i, j] &= \|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\mathbf{x}_i^T \mathbf{x}_j \\ &= (r_0 + \rho_i)^2 + (r_0 + \rho_j)^2 - 2\mathbf{x}_i^T \mathbf{x}_j \\ &= 2r_0^2 + (2r_0\rho_i + \rho_i^2) + (2r_0\rho_j + \rho_j^2) - 2\mathbf{x}_i^T \mathbf{x}_j, \end{aligned}$$

where ρ_i is distributed in such a way that we have uniform distribution over the circular band. We can show that

$$\mathbf{D} = \mathbf{A}\mathbf{S}\mathbf{A}^T,$$

where

$$\mathbf{A} = \begin{bmatrix} r_0 & x_{1,1} & x_{1,2} & 2r_0\rho_1 + \rho_1^2 \\ \vdots & \vdots & \vdots & \vdots \\ r_0 & x_{n,1} & x_{n,2} & 2r_0\rho_n + \rho_n^2 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 2 & 0 & 0 & \frac{1}{r_0} \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ \frac{1}{r_0} & 0 & 0 & 0 \end{bmatrix}.$$

We can further write \mathbf{S} as

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}, \quad \mathbf{\Lambda} = \text{diag} \left(-2, -2, \frac{r_0 + \sqrt{1+r_0^2}}{r_0}, \frac{r_0 - \sqrt{1+r_0^2}}{r_0} \right).$$

It follows that

$$\sigma_1(\mathbf{D}) \leq \frac{r_0 + \sqrt{1+r_0^2}}{r_0} \sigma_1(\mathbf{A}\mathbf{A}^T)$$

and

$$\sigma_4(\mathbf{D}) \geq \min \left(2, \frac{\sqrt{1+r_0^2} - r_0}{r_0} \right) \sigma_4(\mathbf{A}\mathbf{A}^T).$$

We can compute the expectation of this matrix over the distribution of node positions. Having uniform distribution of the sensors over the circular ring, we have for the probability distribution of ρ :

$$p_\rho(\rho) = \frac{r_0 + \rho}{r_0 a}, \quad \text{for } -\frac{a}{2} \leq \rho \leq \frac{a}{2}.$$

Thus, the expectation of the matrix $\mathbf{A}^T \mathbf{A}$ is easily computed as

$$\mathbb{E}[\mathbf{A}^T \mathbf{A}] = \begin{bmatrix} nr_0^2 & 0 & 0 & nr_0 \frac{a^2}{4} \\ 0 & \frac{n}{2}(r_0^2 + \frac{a^2}{4}) & 0 & 0 \\ 0 & 0 & \frac{n}{2}(r_0^2 + \frac{a^2}{4}) & 0 \\ nr_0 \frac{a^2}{4} & 0 & 0 & n(\frac{a^2}{16} + \frac{r_0^2 a^2}{3}) \end{bmatrix}.$$

Let the largest and smallest singular values of $\mathbb{E}[\mathbf{A}^T \mathbf{A}]$ be $n\sigma_{\max}(r_0, a)$ and $n\sigma_{\min}(r_0, a)$. Using the fact that $\sigma_i(\cdot)$ is a Lipschitz continuous function of its arguments, together with the Chernoff bound for large deviation of sums of i.i.d. random variables, we get

$$\begin{aligned} \mathbb{P}(\sigma_1(\mathbf{A}\mathbf{A}^T) > 2n\sigma_{\max}(r_0, a)) &\leq e^{-Cn}, \\ \mathbb{P}(\sigma_1(\mathbf{A}\mathbf{A}^T) < (1/2)n\sigma_{\max}(r_0, a)) &\leq e^{-Cn}, \end{aligned} \tag{2.9}$$

$$\mathbb{P}(\sigma_4(\mathbf{A}\mathbf{A}^T) < (1/2)n\sigma_{\min}(r_0, a)) \leq e^{-Cn}, \tag{2.10}$$

for some constant C . Hence, with high probability, $\kappa(\mathbf{D}) \leq \frac{4\sigma_{\max}(r_0, a)}{\sigma_{\min}(r_0, a)} = f_\kappa(r_0, a)$.

Now to bound μ , note that with probability 1 the columns of \mathbf{A} are linearly independent. Therefore, there exists a matrix $\mathbf{B} \in \mathbb{R}^{r \times r}$ such that $\mathbf{A} = \mathbf{V}\mathbf{B}^T$ with $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. The SVD of \mathbf{D} then reads $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$ with $\mathbf{\Sigma} = \mathbf{Q}^T\mathbf{B}^T\mathbf{S}\mathbf{B}\mathbf{Q}$ and $\mathbf{U} = \mathbf{V}\mathbf{Q}$ for some orthogonal matrix \mathbf{Q} . To show incoherence property **A0**, we need to show that, for all $i \in [n]$,

$$\|\mathbf{V}_i\|^2 \leq \frac{4\mu}{n}.$$

Since $\mathbf{V}_i = \mathbf{B}^{-1}\mathbf{A}_i$, we have $\|\mathbf{V}_i\|^2 \leq \sigma_4(\mathbf{B})^{-2}\|\mathbf{A}_i\|^2 \leq \sigma_4(\mathbf{A})^{-2}\|\mathbf{A}_i\|^2$. Combined with $\|\mathbf{A}_i\|^2 = r_0^2 + (r_0 + \rho_i)^2 + (2r_0\rho_i + \rho_i^2)^2 \leq r_0^2 + (r_0 + a)^2 + (2r_0a + a^2)^2$ and (2.10), we have

$$\|\mathbf{U}_i\|^2 \leq \frac{f_\mu(r_0, a)}{n}, \quad (2.11)$$

with high probability, where $f_\mu(r_0, a) = 2(r_0^2 + (r_0 + a)^2 + (2r_0a + a^2)^2)$.

To show incoherence property **A1**, we use $|\mathbf{D}[i, j]| \leq (2r_0 + a)^2$ and

$$\sigma_1(\mathbf{D}) \geq \frac{1}{4}n\sigma_{\min}(r_0, a) \min\left(2, \frac{\sqrt{1+r_0^2}-r_0}{r_0}\right)$$

from (2.9). Then,

$$\frac{|\mathbf{D}[i, j]|}{\sigma_1(\mathbf{D})} \leq \frac{g(r_0, a)}{n}, \quad (2.12)$$

with high probability, where $g(r_0, a) = \max\left(2, \frac{4r_0}{\sqrt{1+r_0^2}-r_0}\right)(2r_0 + a)^2/\sigma_{\min}(r_0, a)$. Combining (2.11) and (2.12), we see that the incoherence property is satisfied, with high probability.

Further, (2.8) holds, with high probability, if the right-hand side of (2.5) is less than $C_3 \frac{\sqrt{1+r_0^2}+r_0}{r_0}\sigma_{\max}(r_0, a)$, since $\sigma_4(\mathbf{D}) \leq \frac{1}{2}n \frac{\sqrt{1+r_0^2}+r_0}{r_0}\sigma_{\max}(r_0, a)$. This finishes the proof of Theorem 2.1. \square

2.A.2 Proof of Theorem 2.2

Recall from (1.6) that $(\mathbf{L}\mathbf{X}\mathbf{X}^T\mathbf{L} - \mathbf{L}\widehat{\mathbf{X}}\widehat{\mathbf{X}}^T\mathbf{L})$ has rank at most 2η where η is the dimension of the space in which sensors are placed (in our case $\eta = 2$). Therefore,

$$\left\|\mathbf{L}\mathbf{X}\mathbf{X}^T\mathbf{L} - \mathbf{L}\widehat{\mathbf{X}}\widehat{\mathbf{X}}^T\mathbf{L}\right\|_F \leq \sqrt{2\eta} \left\|\mathbf{L}\mathbf{X}\mathbf{X}^T\mathbf{L} - \mathbf{L}\widehat{\mathbf{X}}\widehat{\mathbf{X}}^T\mathbf{L}\right\|_2,$$

where we used the fact that for any matrix \mathbf{A} of rank r we have $\|\mathbf{A}\|_F \leq \sqrt{r}\|\mathbf{A}\|_2$. Furthermore, the spectral norm can be bounded in terms of \mathbf{D} and $\widehat{\mathbf{D}}$ as follows.

$$\begin{aligned} & \left\|\mathbf{L}\mathbf{X}\mathbf{X}^T\mathbf{L} - \mathbf{L}\widehat{\mathbf{X}}\widehat{\mathbf{X}}^T\mathbf{L}\right\|_2 \\ & \stackrel{(a)}{\leq} \left\|\mathbf{L}\mathbf{X}\mathbf{X}^T\mathbf{L} - \frac{1}{2}\mathbf{L}\widehat{\mathbf{D}}\mathbf{L}\right\|_2 + \left\|\frac{1}{2}\mathbf{L}\widehat{\mathbf{D}}\mathbf{L} - \mathbf{L}\widehat{\mathbf{X}}\widehat{\mathbf{X}}^T\mathbf{L}\right\|_2 \\ & \stackrel{(b)}{\leq} \frac{1}{2} \left\|\mathbf{L}(\mathbf{D} - \widehat{\mathbf{D}})\mathbf{L}\right\|_2 + \frac{1}{2} \left\|\mathbf{L}(-\mathbf{D} + \widehat{\mathbf{D}})\mathbf{L}\right\|_2, \end{aligned} \quad (2.13)$$

where in (a), we use the triangle inequality and $\mathbf{L}\widehat{\mathbf{X}} = \widehat{\mathbf{X}}$. In (b), we use

$$-\frac{1}{2}\mathbf{L}\mathbf{D}\mathbf{L} = \mathbf{L}\mathbf{X}\mathbf{X}^T\mathbf{L},$$

and the fact that for any matrix \mathbf{A} of rank η , $\left\|\frac{1}{2}\mathbf{L}\widehat{\mathbf{D}}\mathbf{L} - \widehat{\mathbf{X}}\widehat{\mathbf{X}}^T\right\|_2 \leq \left\|\frac{1}{2}\mathbf{L}\widehat{\mathbf{D}}\mathbf{L} - \mathbf{A}\right\|_2$. In particular, by setting $\mathbf{A} = \frac{1}{2}\mathbf{L}\mathbf{D}\mathbf{L}$ the second term in (2.13) follows. Since \mathbf{L} is a projection matrix we have $\|\mathbf{L}\|_2 = 1$. Hence, from (2.13) we can conclude that

$$\left\|\mathbf{L}\mathbf{X}\mathbf{X}^T\mathbf{L} - \mathbf{L}\widehat{\mathbf{X}}\widehat{\mathbf{X}}^T\mathbf{L}\right\|_2 \leq \left\|\widehat{\mathbf{D}} - \mathbf{D}\right\|_2.$$

This immediately leads to the result of Theorem 2.2. \square

2.A.3 Proof of Lemma 2.2

Note that by the definition of \mathbf{D}^s , we have $|\mathcal{P}_E(\mathbf{D}^s)[i, j]| \leq \delta_n^2$ for all i and j . Define \mathbf{A} as

$$\mathbf{A}[i, j] = \begin{cases} 1 & \text{if } (i, j) \in E \cap S, \\ 0 & \text{otherwise.} \end{cases}$$

We start from a simple relationship between an element-wise bounded matrix and its operator norm.

$$\|\mathcal{P}_E(\mathbf{D}^s)\|_2 \leq \delta_n^2 \max_{\|x\|=\|y\|=1} \sum_{i,j} |x_i| |y_j| \mathbf{A}[i, j] = \delta_n^2 \|\mathbf{A}\|_2. \quad (2.14)$$

The inequality in (2.14) follows from the fact that $\mathcal{P}_E(\mathbf{D}^s)$ is element-wise bounded by δ_n . We can further bound the operator norm $\|\mathbf{A}\|_2$, by applying the celebrated Gershgorin circle theorem to a symmetrized version of \mathbf{A} . Define a symmetric matrix $\overline{\mathbf{A}}$ as

$$\overline{\mathbf{A}}[i, j] = \begin{cases} 1 & \text{if } (i, j) \in E \cap S \text{ or } (j, i) \in E \cap S, \\ 0 & \text{otherwise.} \end{cases}$$

Since $0 \leq \mathbf{A}[i, j] \leq \overline{\mathbf{A}}[i, j]$ for all i and j , we have $\|\mathbf{A}\|_2 \leq \|\overline{\mathbf{A}}\|_2$. Applying the Gershgorin circle theorem we get

$$\|\overline{\mathbf{A}}\|_2 \leq \max_{i \in [n]} \sum_{j \in [n]} |\overline{\mathbf{A}}[i, j]|.$$

Define random variables $\{Y_1, \dots, Y_n\}$, where Y_i is the number of non-zero entries in the i th row of $\overline{\mathbf{A}}$. Then,

$$\|\overline{\mathbf{A}}\|_2 \leq \max_{i \in [n]} Y_i.$$

We need to show that Y_i concentrates around its mean. Since Y_i 's are binomial random variables, we can apply the Chernoff bound. Recall that $(i, j) \in S$ if $\|\mathbf{x}_i - \mathbf{x}_j\| \leq \delta_n$. By the definition of E , each sample is sampled with probability p . Then the probability that either (i, j) or (j, i) is in E is $2p - p^2$.

Each entry in the i th row of $\overline{\mathbf{A}}$ is an independent Bernoulli random variable with probability of being one equal to $q(2p - p^2)$, where q is the probability that a pair is in S . Thus, we have $\mathbb{E}[Y_i] = q(2p - p^2)n$. In order to find the bounds on $\mathbb{E}[Y_i]$, we need to bound q . Figure 2.12(a) shows the process for obtaining the bounds on q .

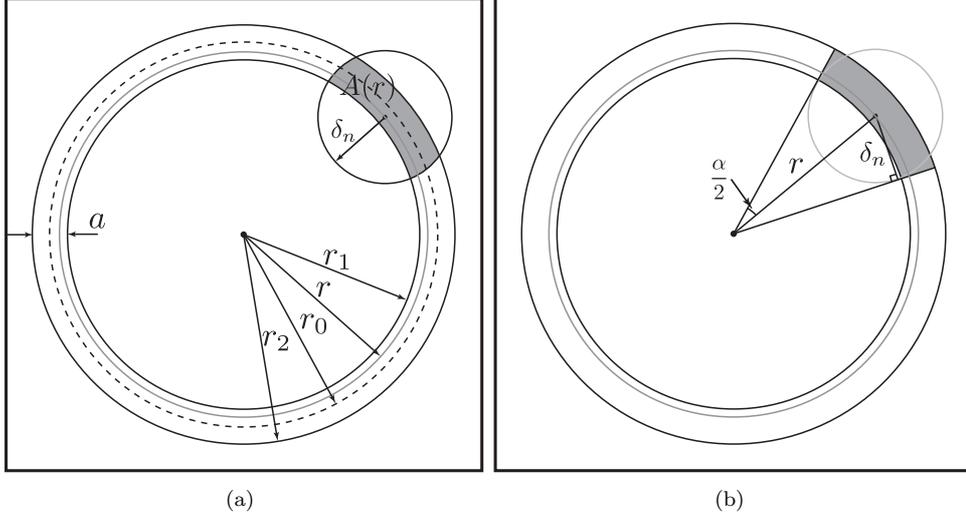


Figure 2.12: (a) The process for bounding the probability of a pair of sensors to fall in S . $r_1 = r_0 - a/2$ and $r_2 = r_0 + a/2$. (b) Upper bound on $A(r)$. The grey area made by the tangents to the δ_n circle is an upper bound for $A(r)$.

$$q = \mathbb{P}\{\|\mathbf{x}_i - \mathbf{x}_j\| \leq \delta_n\} = \int_{r_1}^{r_2} \frac{2\pi r}{\pi(r_2^2 - r_1^2)} p_2(r) dr,$$

where $p_2(r) = A(r)/(\pi(r_2^2 - r_1^2))$.

Upper Bound on $A(r)$: Obviously the area $A(r)$ can be bounded by what is shown in Figure 2.12(b). Thus, we will have $\sin(\alpha/2) = \delta_n/r$.

Note that for $0 < \alpha < \pi$, $\alpha/\pi \leq \sin \alpha/2 \leq \alpha/2$. Hence, $\alpha/\pi \leq \delta_n/r \leq \alpha/2$. So,

$$A(r) \leq \frac{\alpha}{2\pi} \pi(r_2^2 - r_1^2) \leq \frac{\delta_n \pi}{2r} (r_2^2 - r_1^2).$$

Thus

$$p_2(r) \leq \frac{\frac{\delta_n \pi}{2r} (r_2^2 - r_1^2)}{\pi(r_2^2 - r_1^2)} = \frac{\delta_n}{2r}.$$

$$q \leq \int_{r_1}^{r_2} \frac{2\pi r}{\pi(r_2^2 - r_1^2)} \cdot \frac{\delta_n}{2r} dr = \frac{\delta_n}{r_2 + r_1} = \frac{\delta_n}{2r_0}.$$

Lower Bound on $A(r)$: In order to find the lower bound, we consider two cases:

Case 1 ($\delta_n \leq a$):

In this case the minimum area of the intersection is achieved when the center of the circle is on the exterior boundary of the region as shown in Figure 2.13(a). In this case, one can show that,

$$A(r) \geq \frac{\pi \delta_n^2}{4}. \quad (2.15)$$

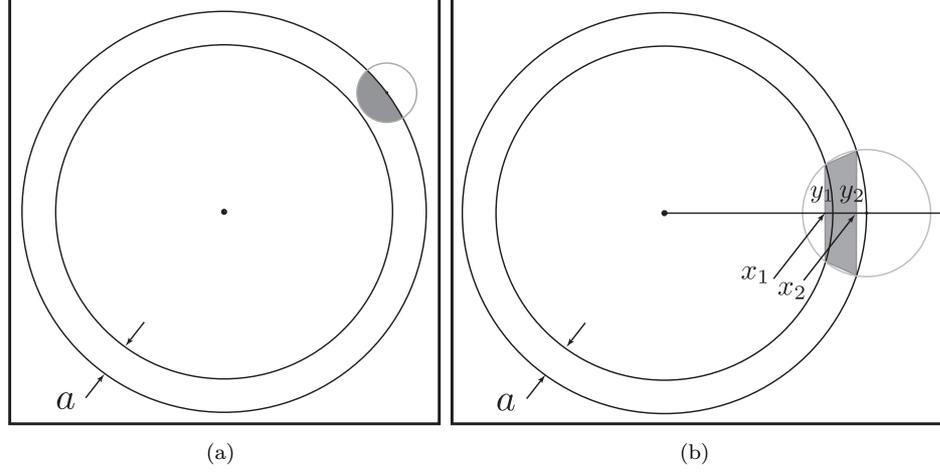


Figure 2.13: Evaluation of lower bound for $A(r)$. In (a) we assume that $\delta_n \leq a$ whereas in (b) we take $\delta_n > a$. In both cases the minimum intersection is achieved when the center of δ_n circle is on the exterior boundary of the region.

Case 2 ($\delta_n > a$):

In this case, the minimum area is achieved when the center of the circle is on the exterior boundary as in Figure 2.13(b), where

$$\begin{aligned} x_1 &= \frac{r_2^2 - \delta_n^2 + r_1^2}{2r_2}, & y_1 &= \frac{1}{r_2} \sqrt{(\delta_n^2 - a^2)(4r_0^2 - \delta_n^2)}, \\ x_2 &= \frac{r_2^2 - \delta_n^2 + r_2^2}{2r_2}, & y_2 &= \frac{1}{r_2} \sqrt{\delta_n^2(4r_2^2 - \delta_n^2)}. \end{aligned}$$

Thus, we will have

$$\begin{aligned} A(r) &\geq \frac{y_1 + y_2}{2} (x_2 - x_1) \\ &\geq \frac{\sqrt{\delta_n^2(4r_2^2 - \delta_n^2)}}{2r_2} \times \frac{r_2^2 - r_1^2}{2r_2} = \delta_n \sqrt{\left(r_2 - \frac{1}{4}\delta_n^2\right) \frac{r_2^2 - r_1^2}{2r_2^2}}. \end{aligned}$$

If we assume that $r_2 \geq \frac{1}{\sqrt{2}}\delta_n$, which is a reasonable assumption according to the problem statement, we will have

$$A(r) \geq \frac{1}{2}\delta_n^2 \frac{r_2^2 - r_1^2}{2r_2^2} \geq \frac{a r_0}{2(r_0 + a)^2} \delta_n^2. \quad (2.16)$$

Combining (2.15) and (2.16), we can find the lower bound for $A(r)$ as

$$A(r) \geq \min\left(\frac{\pi}{4}, \frac{a r_0}{2(r_0 + a)^2}\right) \delta_n^2 = \frac{a r_0}{2(r_0 + a)^2} \delta_n^2.$$

Thus,

$$\begin{aligned} q &= \int_{r_1}^{r_2} \frac{2\pi r}{\pi(r_2^2 - r_1^2)} p_2(r) dr \\ &= \int_{r_1}^{r_2} \frac{2\pi r}{\pi(r_2^2 - r_1^2)} \frac{A(r)}{\pi(r_2^2 - r_1^2)} dr \geq \frac{\delta_n^2}{4\pi(r_0 + a)^2}. \end{aligned}$$

From the above calculations, we have that $\frac{\delta_n^2}{4\pi(r_0+a)^2}pn \leq \mathbb{E}[Y_i] \leq \frac{1}{r_0}\delta_npn$. Applying the Chernoff bound to Y_i , we have

$$\mathbb{P}\left(Y_i > (1 + \alpha)\mathbb{E}[Y_i]\right) \leq 2^{-(1+\alpha)\mathbb{E}[Y_i]}.$$

In other words

$$\mathbb{P}\left(Y_i > (1 + \alpha)\frac{1}{r_0}\delta_npn\right) \leq 2^{-(1+\alpha)\frac{\delta_n^2}{4\pi(r_0+a)^2}pn}.$$

Applying the union bound, we get

$$\begin{aligned} \mathbb{P}\left(\max_{i \in [n]} Y_i > (1 + \alpha)\frac{1}{r_0}\delta_npn\right) &\leq n2^{-(1+\alpha)\frac{\delta_n^2}{4\pi(r_0+a)^2}pn} \\ &\leq 2^{-\left((1+\alpha)\frac{\delta_n^2}{4\pi(r_0+a)^2}pn - \log_2 n\right)}. \end{aligned}$$

By the assumption that $\delta_n p = \Omega(r_0 \sqrt{\log_2 n/n})$, there exists constants c and N , such that $\delta_n^2 p \geq cr_0^2 \log_2 n/n$, for $n \geq N$. Define a positive parameter β such that $1 + \beta = \frac{c(1+\alpha)r_0^2}{4\pi(r_0+a)^2}$. Then we will have

$$\mathbb{P}\left(\max_{i \in [n]} Y_i > \frac{4\pi(1+\beta)}{cr_0^3}(r_0+a)^2\delta_npn\right) \leq n^{-\beta}.$$

Finally with probability $1 - n^{-\beta}$,

$$\begin{aligned} \|\mathcal{P}_E(\mathbf{D}^s)\|_2 &\leq \frac{4\pi(1+\beta)}{c}\delta^3 \left(\sqrt{\frac{\log n}{n}}\right)^3 pn \\ &= C(r_0+a)^2\delta^3 \left(\sqrt{\frac{\log n}{n}}\right)^3 pn. \end{aligned}$$

This finishes the proof of Lemma 2.2. □

Chapter 3

Distributed Low-Complexity Sensor Localization

A great deal more is known than has been proved.

Richard Feynman

3.1 Introduction

In Chapter 2, we studied the calibration problem as a dual formulation for sensor localization, in which the local connectivity information was missing. Besides this characteristic, in the solution proposed in Chapter 2, the localization problem is divided into two sub-problems and each problem is treated separately; first using the properties of Euclidean distance matrices we estimate the missing distances; and then we estimate the positions from the completed matrix. One would ask if it is really necessary to solve the problem in two distinct steps? The answer is no. There are several methods in the literature that aim at solving the localization problem all at once [13, 14].

The localization process based on distance information can be performed in a *centralized* or *distributed* fashion. In centralized algorithms (e.g., MDSLICALIZE [34]) it is assumed that the sensors send the distance information to a central base station where the positions are estimated and forwarded back to the sensors. However, in order to decrease the communication costs (specially in dynamic networks) and also to split the computation power for estimating the positions, distributed sensor localization algorithms have gained more interest. In this chapter¹,

1. The results presented in this chapter are from joint works with S. Haghghatshoar, A. Hormati and M. Vetterli.

we propose a distributed algorithm in which each sensor estimates its own location based on the distance information it receives from its neighbors.

We can further divide the localization algorithms into two classes; anchor-based localization [111] and anchor-free localization [21]. In anchor based localization, the positions of few sensors are assumed fixed and known and the positions of the rest are estimated relative to the anchors. In an anchor-free localization, however, the relative positions are estimated up to rigid transformations (refer to Section 1.2.1). The analysis and algorithm presented in this chapter is for anchor-free localization, but they are easily extendable to the anchor-based case with simple modifications.

Distributed anchor-free sensor localization has been extensively studied in the past decade and in some cases simple algorithms have been proposed [21, 100, 136]. Since these methods rely on computation over individual sensors, they are normally not computationally demanding and use simple structures for estimating the positions. In most of the cases, however, reconstruction accuracy is sacrificed in order to achieve coarse but less expensive estimations. Moreover, because of the difficulty of the analysis for distributed algorithms in general, to the best of our knowledge there is no analysis available for these methods. Many of these methods are local minimization methods and commonly find false minima as the solution of the localization. As all the localization procedures are non-linear and non-convex, it is often impossible to guarantee a global minimum for the solution. However, we can try to design algorithms that avoid local minima in a more intelligent way. In this chapter, we introduce a very simple update procedure for each sensor to minimize the s-stress criterion, introduced in Section 1.3.3. We further provide numerical evidence that our distributed minimization method converges in most setups. We also study the open question of existence of non-global minimizers of the s-stress function. We propose an equivalent problem for finding the answer to this problem in the form of a hypothesis.

3.2 The S-stress Criterion For Sensor Localization

Similarly to the previous chapter, consider a set of n sensors lying in an η -dimensional space. In most applications η is equal to two or three. The distance matrix for such configuration is given by

$$\mathbf{D} = [d_{i,j}^2] = D(\mathbf{X}) = \|\mathbf{X}_i - \mathbf{X}_j\|^2, \quad i, j \in \{1, \dots, n\},$$

where $\mathbf{X}_i \in \mathbb{R}^\eta$, the i -th row of \mathbf{X} , is the coordinate of the sensor number i in the η -dimensional space.

Normally, the exact distances are not available. Only a noisy and/or incomplete version of the distance matrix is observed. We first assume that the distance matrix is corrupted with arbitrary symmetric noise. Moreover, we assume that after adding noise to the distance matrix, all its elements will remain non-negative. We call the noisy distance matrix $\tilde{\mathbf{D}}$:

$$\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{N},$$

where \mathbf{N} is a symmetric noise matrix and $\tilde{\mathbf{D}}$ has non-negative elements with a zero diagonal. We denote elements of $\tilde{\mathbf{D}}$ by $\tilde{d}_{i,j}^2$, for $i, j \in \{1, \dots, n\}$. We can further assume that the observed distance matrix is incomplete.

Assume that each sensor i , is connected only to its neighbors listed in \mathcal{N}_i (i.e., sensor i has only distance information from sensors in \mathcal{N}_i). For every configuration $\mathbf{X} \in \mathbb{R}^{n \times \eta}$, we define the

cost function $f(\mathbf{X})$ as follows

$$f(\mathbf{X}) = \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \left(\|\mathbf{X}_i - \mathbf{X}_j\|^2 - \tilde{d}_{i,j}^2 \right)^2. \quad (3.1)$$

By a configuration we mean an $n \times \eta$ matrix whose i -th row is \mathbf{X}_i^T , the position of the i -th sensor. This cost function in fact represents the Frobenius norm of the difference between the observed distance matrix and the distance matrix resulting from the estimated configuration. Also note that in the noiseless case, having a zero cost is equivalent to finding the exact configuration in \mathbb{R}^η . This cost function — also shown in Section 1.3.3 — is called the *s-stress criterion* in the multidimensional scaling literature [14].

The goal of this chapter is to solve the following problem: *Given the corrupted distance measurements $\tilde{\mathbf{D}}$, minimize the cost function $f(\mathbf{X})$.*

$$\begin{aligned} \text{s-stress:} \quad & \widehat{\mathbf{X}} = \arg \min_{\mathbf{X} \in \mathbb{R}^{n \times \eta}} \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \left(\|\mathbf{X}_i - \mathbf{X}_j\|^2 - \tilde{d}_{i,j}^2 \right)^2. \\ & \text{s.t.} \quad \tilde{\mathbf{D}} \in \mathbb{S}_h^n \end{aligned} \quad (3.2)$$

In the following, we first propose an algorithm with very simple updates to find a local minimum of the s-stress function in a fast, distributed and efficient way. Afterwards, we study the properties of the cost function in more details and try to shed some light on the behavior of this cost function and difficulties in its minimization. We also provide examples for which our update method finds the global minimum while classic minimization approaches fail.

3.3 Distributed Sensor Localization With the S-stress Criterion

We can decompose the s-stress function in (3.1) as

$$f(\mathbf{X}) = \sum_{i=1}^n f_i(\mathbf{X}),$$

with

$$f_i(\mathbf{X}) = \sum_{j \in \mathcal{N}_i} \left(\|\mathbf{X}_i - \mathbf{X}_j\|^2 - \tilde{d}_{i,j}^2 \right)^2.$$

This decomposition is a key step in finding a distributed algorithm for minimizing $f(\mathbf{X})$. For simplicity of representation, we assume that the sensors are in a two-dimensional space with $\mathbf{X}_i = [x_i, y_i]^T$. The results are immediately extendable to higher dimensions. A simple computation shows that

$$\frac{\partial f(\mathbf{X})}{\partial x_i} = 2 \frac{\partial f_i(\mathbf{X})}{\partial x_i}. \quad (3.3)$$

In order to minimize the cost function, we choose an alternate coordinate descent method. Assume at iteration k , the coordinate x_i is updated by $\Delta x_i^{(k)}$. Thus,

$$f_i(\mathbf{X}^{(k+1)}) = \sum_{j \in \mathcal{N}_i} \left((x_i^{(k)} + \Delta x_i^{(k+1)} - x_j^{(k)})^2 + (y_i^{(k)} - y_j^{(k)})^2 - \tilde{d}_{i,j}^2 \right)^2.$$

Algorithm 3.1 ALTERNATIVE COORDINATE DESCENT METHOD FOR MINIMIZING THE S-STRESS FUNCTION

Input: Distance matrix $\tilde{\mathbf{D}}$.

Output: Estimated positions: $\widehat{\mathbf{X}}$.

1. Assume an initial configuration for the sensors \mathbf{X}_0 (e.g. $\mathbf{X}_0 = \mathbf{0}$);
 2. **repeat**
 3. **for** sensor number $i = 1$ to n **do**
 4. Assume the configuration of the rest of the sensors fixed;
 5. Use the coordinate descent method to find the x coordinate of sensor i using distance information of its neighbors;
 6. Use the coordinate descent method to find the y coordinate of sensor i using distance information of its neighbors;
 7. Send the estimated position of sensor i to its neighbors;
 8. **end for**
 9. **until** convergence or maximum number of iterations is reached.
-

In order to minimize $f_i(\mathbf{X}^{(k+1)})$ we find its derivative with respect to $\Delta x_i^{(k)}$;

$$\begin{aligned} \frac{\partial f_i^{(k+1)}(\mathbf{X}^{(k+1)})}{\partial \Delta x_i^{(k+1)}} &= 4|\mathcal{N}_i|(\Delta x_i^{(k+1)})^3 \\ &+ \sum_{j \in \mathcal{N}_i} 3(x_i^{(k)} - x_j^{(k)})(\Delta x_i^{(k+1)})^2 \\ &+ \sum_{j \in \mathcal{N}_i} [3(x_i^{(k)} - x_j^{(k)})^2 + (y_i^{(k)} - y_j^{(k)})^2 - \tilde{d}_{i,j}^2] \Delta x_i^{(k+1)} \\ &+ [(x_i^{(k)} - x_j^{(k)})^3 + (x_i^{(k)} - x_j^{(k)})(y_i^{(k)} - y_j^{(k)})^2 - (x_i^{(k)} - x_j^{(k)})\tilde{d}_{i,j}^2]. \end{aligned}$$

Setting the derivative to zero will lead to at most three solutions for $\Delta x_i^{(k+1)}$. These roots are analytically defined and easy to compute given the distance information. We choose the root that decreases the value of $f_i(\mathbf{X})$ the most. After updating the x_i coordinate, we update y_i and then we continue to the next sensors. We call each round of updates from sensor 1 to n , a *sweep*.

The algorithm is summarized in Algorithm 3.1. Note that the decomposition in (3.3) allows us to implement this algorithm in a distributed manner. We assume that every sensor collects the distance information of itself from its neighbors. It then minimizes its cost function $f_i(\mathbf{X})$ and passes its estimated position to its neighbors.

Example 3.1

Consider 4 sensors in a two-dimensional space. Suppose that the distance matrix is noiseless and complete, i.e., $\tilde{\mathbf{D}} = \mathbf{D}$. The original configuration is shown in Figure 3.1(c) with red stars. We start from a random configuration and run Algorithm 3.1. Figures 3.1(a) and 3.1(b) show the value of the cost function and the reconstruction error $\text{dist}(\widehat{\mathbf{X}}, \mathbf{X})$ (defined in (2.3)), respectively. We observe that although Algorithm 3.1 guarantees a non-increasing update for the cost function at each iteration, the value of $\text{dist}(\widehat{\mathbf{X}}, \mathbf{X})$ might sometimes increase. The estimated positions are illustrated in Figure 3.1(c) by blue crosses. It is clear that the reconstruction is a translated and rotated version of the original configuration.

We will see in the next section that the s-stress function has a very complicated characteristic

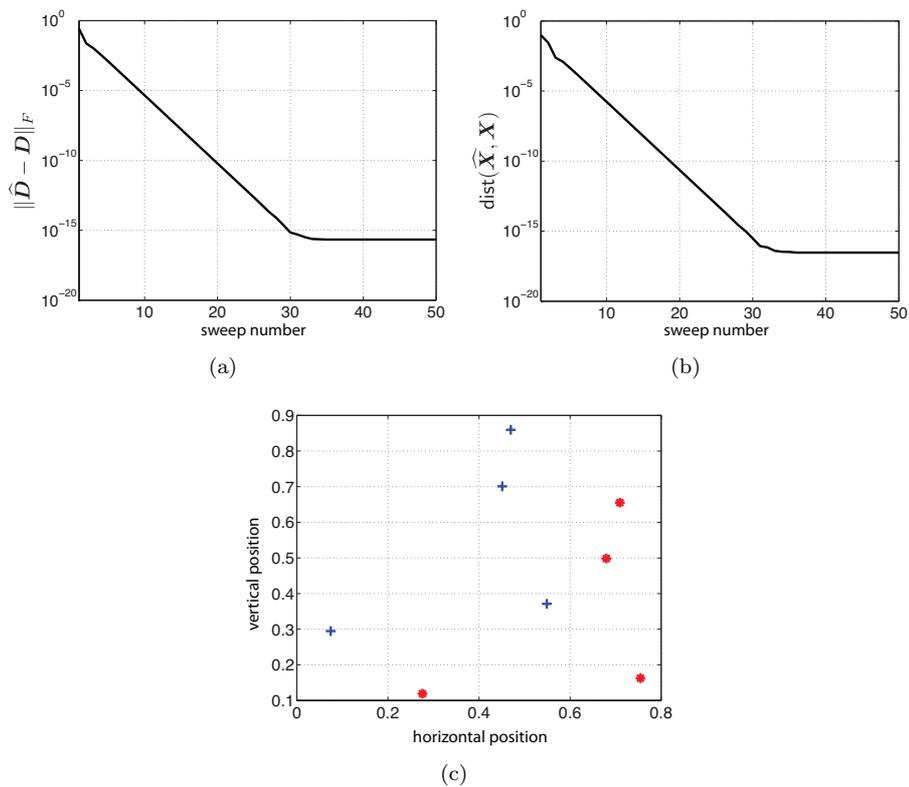


Figure 3.1: Figures for Example 3.1: (a) shows the decrease in the value of the s-stress function when Algorithm 3.1 is used for localization. (b) shows the reconstruction error for the positions of the sensors. (c) shows the original configuration of the sensors (red stars) and the reconstructed configuration (blue crosses).

with a behavior not yet fully understood by mathematicians. The unknown properties of this cost function, make the analysis of our algorithm also difficult. A-priori, there is no guarantee that our proposed algorithm will converge to a global minimizer of (3.2). Let us run a test to see what is the percentage of times that our algorithm converges to the global minimum.

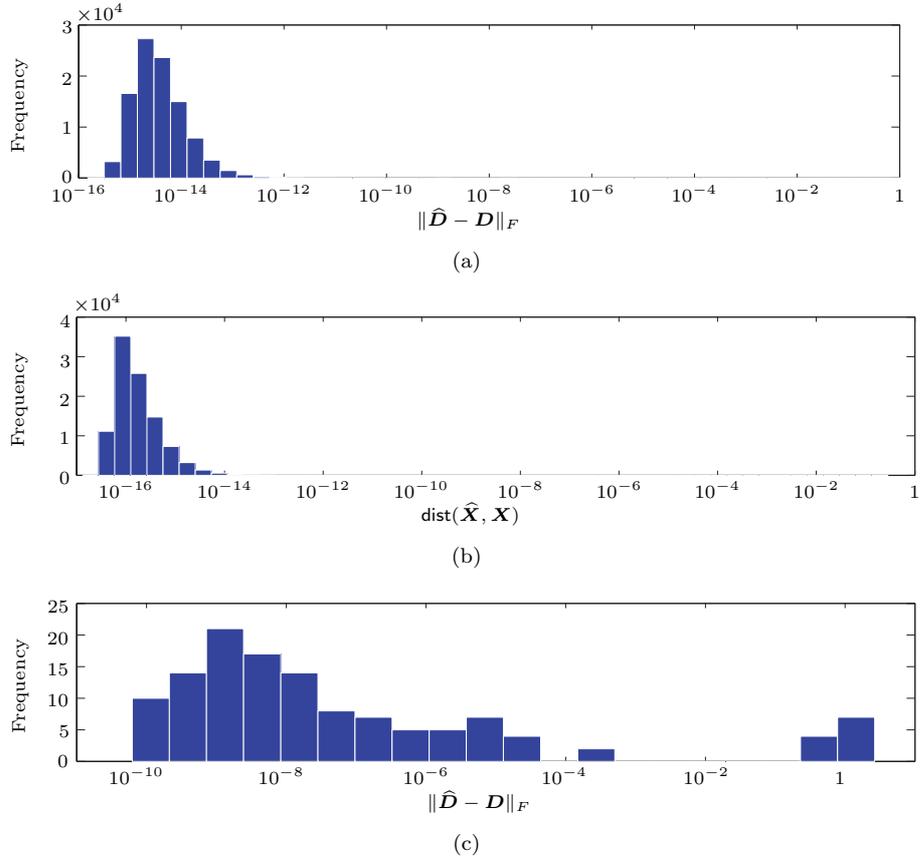


Figure 3.2: Figures of Example 3.2: (a) shows the histogram of the estimated values of the s -stress function for 100,000 experiments with 10 sensors randomly distributed in $[0, 1] \times [0, 1]$. (b) shows the histogram of the values of $\text{dist}(\widehat{\mathbf{X}}, \mathbf{X})$ for the same setup. (c) shows the zoomed version of the right part of (a) in the same setup.

Example 3.2

We run the following test 100,000 times: Consider 10 sensors spread uniformly at random on the two-dimensional square $[0, 1] \times [0, 1]$. This configuration produces our original distance matrix \mathbf{D} . We assume that the distance matrix is observed completely and without noise. Starting from an initial random configuration, we run the proposed algorithm to estimate the sensor positions. Figures 3.2(a) and 3.2(b) show the histogram of the values of $\|\widehat{\mathbf{D}} - \mathbf{D}\|_F$ and $\text{dist}(\widehat{\mathbf{X}}, \mathbf{X})$, respectively. Out of 100,000 experiments, 99.66% resulted in $\|\widehat{\mathbf{D}} - \mathbf{D}\|_F < 10^{-12}$ and 99.83% in $\text{dist}(\widehat{\mathbf{X}}, \mathbf{X}) < 10^{-12}$. Figure 3.2(c) shows the larger error distribution between 10^{-10} and 1. This shows that in almost any scenario, the algorithm finds the global minimum of the s -stress function. But it also shows that there exist cases for which the algorithm cannot find the global minimizer of the cost function.

Glunt et al. in [43] note:

“A number of algorithms have been proposed for the solution of the embedding problem [s-stress criterion] by researchers in multidimensional scaling. An algorithm due to Leeuw and Takane was modified by Browne (1987) by adding a Newton Raphson step and resulted in *the best algorithm known to us* for finding a local minimum solution of the embedding problem. Newton Raphson either finds the global minimum (*about 90% of the time in our examples*) or a local minimum with objective function near the global minimum.”

Although we do not have access to the setup that was used in the experiments of [43], it is mentioned in their paper that the test predistance matrices (a symmetric matrix with zero diagonal and non-negative entries) are produced randomly. This resembles the setup that we use in our example. The comparison in the performance of our algorithm versus the best algorithm known to the authors shows a large advantages of our proposed method.

In the following section, we introduce some key properties of the s-stress function and the space of local minimizers of it. This section can help in understanding the cost function in more depth, develop better algorithms and use better initial configurations for the existing algorithms.

3.4 Algebraic Properties of the S-stress Function

The cost function $f(\mathbf{X})$ defined in (3.1) is a non-convex function and normally finding the global minima of a non-convex function is not trivial. In this section, we aim at showing some key properties of the s-stress function that not only attest its non-convexity, but also show that simple optimization algorithms are unable to provide global convergence guarantees for it. These properties might as well serve as a tool for providing neighborhood convergence analysis for optimization algorithms applied on the s-stress function.

Let us define the connectivity graph associated with a given noisy and incomplete distance matrix $\tilde{\mathbf{D}}$.

Definition 3.1

The connectivity graph of a noisy and incomplete distance matrix $\tilde{\mathbf{D}}$ is a graph $G = (V, E)$ with $V = \{1, 2, \dots, n\}$ the set of nodes, where node i corresponds to sensor i , and E the set of edges where $\{i, j\} \in E$ if and only if $\tilde{\mathbf{D}}[i, j] > 0$ is revealed.

Now we introduce a condition on the observed matrix $\tilde{\mathbf{D}}$ which will be used to show the properties of the s-stress function.

Condition A

The connectivity graph associated with the measurement matrix $\tilde{\mathbf{D}}$ is connected.

We saw in Section 1.2.1 that the s-stress function is invariant to rigid transformation. Thus, the optimization method will be also invariant to this effect. In the following, without any loss of generality we will assume that sensors are centered around the origin, i.e.,

$$\mathbf{1}^T \mathbf{X} = \mathbf{0},$$

where $\mathbf{1}$ is the all one column vector of length n . In the rest of this chapter, we will consider the properties of the s-stress function on the plane $\mathcal{P} = \{\mathbf{X} \mid \mathbf{1}^T \mathbf{X} = \mathbf{0}\}$. The gradient of $f(\mathbf{X})$ can

be computed easily as

$$\frac{\partial f}{\partial \mathbf{X}[k,l]} = 2 \sum_{j|\{j,k\} \in E} (\mathbf{X}[k,l] - \mathbf{X}[j,l]) \left[\|\mathbf{X}_k - \mathbf{X}_j\|^2 - \tilde{d}_{k,j}^2 \right], \quad (3.4)$$

for $k \in \{1, \dots, n\}$ and $l \in \{1, \dots, \eta\}$. It is easy to see that the configuration $\mathbf{X} = \mathbf{0}$ is in fact a local maximum for this cost function and the value of the cost at this configuration would be $\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4$. Moreover, as $\mathbf{X}[k,l]$ grows, the cost will eventually increase. This can be also seen from the positive coefficients of $\mathbf{X}[k,l]^4$ in the s-stress function (3.1).

The above argument shows in fact that the cost function is not convex. In addition to the local minima and local maxima, the cost might also have saddle points.

Example 3.3

Take the original configuration as the one shown in Figure 3.3(a). Assume that the distance matrix is noiseless and complete, i.e.,

$$\tilde{\mathbf{D}} = \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix}.$$

For the configuration shown in Figure 3.3(b) all the gradients (defined in (3.4)) are zero. Further, Figure 3.3(c) shows how the value of the cost function changes as we move sensor number 2 in the horizontal (or vertical) direction. This clearly shows that the second configuration is a saddle point for the cost function.

Also note that since the cost function is a continuous and differentiable function of \mathbf{X} , a smooth global minimum of it always exists and all the gradients are zero at the global minimum. As the cost is invariant to reflection and rotation, the global minimum is not unique; rather, its rotations and reflections on the plane \mathcal{P} are also global minimizers of the cost function.

Lemma 3.1

Assuming Condition A is met, the equality:

$$\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 = 0 \quad (3.5)$$

holds on the plane \mathcal{P} if and only if \mathbf{X} is the zero configuration.

Proof.

If \mathbf{X} is the zero configuration, which is on the plane \mathcal{P} , the equality obviously holds. For the converse, assume two arbitrary $i, j \in \{1, 2, \dots, n\}$. The connectedness assumption of G implies that there is a path $P_{i,j} = i, n_1, n_2, \dots, n_r, j$ between i and j in the graph G .

As all the terms in the summation of (3.5) are non-negative this implies that:

$$\sum_{\{s,e\} \in P_{i,j}} \tilde{d}_{s,e}^2 \|\mathbf{X}_s - \mathbf{X}_e\|^2 = 0,$$

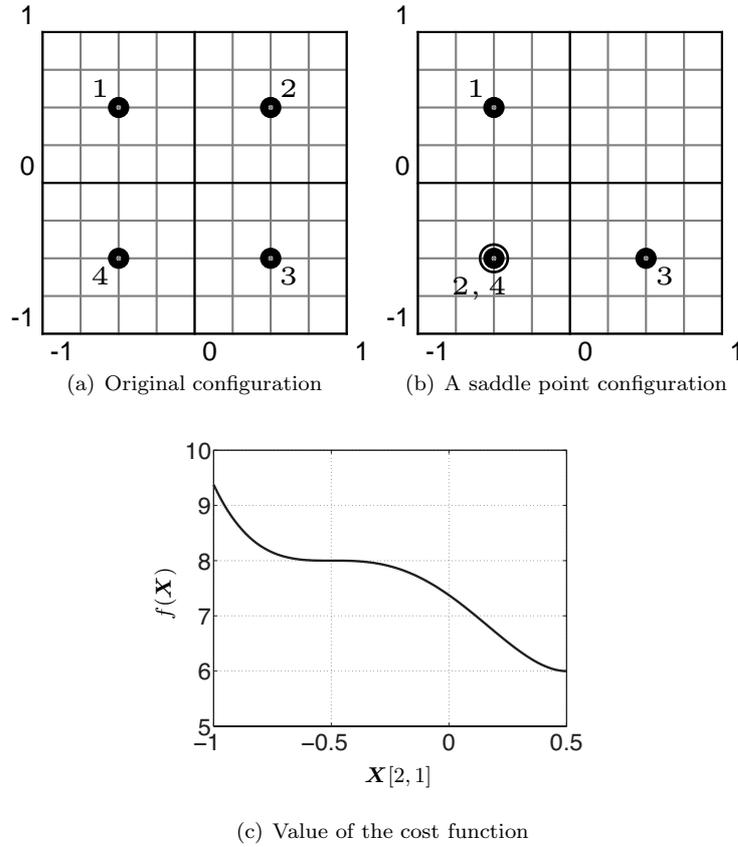


Figure 3.3: (a) shows the original configuration from which a noiseless distance matrix is measured. (b) shows the modified version of the original configuration for which the position of sensor 2 coincides with sensor 4. All the derivatives of the cost function are zero at this configuration. (c) shows the values of the cost function as the first coordinate of sensor 2 is changing (in this case its x component). For better visualization, the zero mean assumption of the sensor positions is neglected.

because $P_{i,j}$ is a sub-graph of G . By the definition of the connectivity graph G , $\tilde{d}_{s,e} > 0$ for all $\{s, e\} \in P_{i,j}$ and this implies that

$$\mathbf{X}_i = \mathbf{X}_{n_1} = \mathbf{X}_{n_2} = \dots = \mathbf{X}_{n_r} = \mathbf{X}_j,$$

and specifically $\mathbf{X}_i = \mathbf{X}_j$. As the choice of i and j was arbitrary this means that $\mathbf{X}_i = \mathbf{X}_j$ for every $i, j \in \{1, 2, \dots, n\}$. Considering the fact that on the plane \mathcal{P} , $\sum_{i=1}^n \mathbf{X}_i = \mathbf{0}$, this implies that $\mathbf{X}_i = \mathbf{0}$ for every $i \in \{1, 2, \dots, n\}$. Thus, \mathbf{X} is the zero configuration. \square

Lemma 3.2

Assuming that Condition A holds, for every fixed non-zero configuration \mathbf{X} on the plane \mathcal{P} , the scaling function $g(\mathbf{X}, \alpha) : \mathbb{R}^{n \times n} \times \mathbb{R} \rightarrow \mathbb{R}$ defined as $g(\mathbf{X}, \alpha) = f(\alpha \mathbf{X})$ has one local maximum and two local minima on the plane \mathcal{P} where the local maximum happens for $\alpha = 0$ and the two local minima are symmetric with respect to the origin.

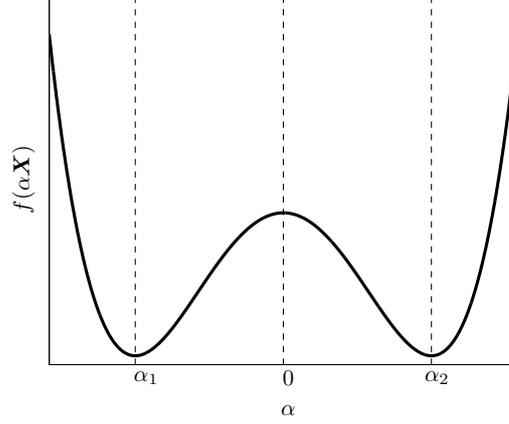


Figure 3.4: Shape of the cost function along any ray passing through origin and a configuration \mathbf{X} . The minima α_1 and α_2 are computed in (3.14).

Proof.

See Appendix 3.A.1. □

According to Lemma 3.2, a cross-section of the cost function on a ray passing through the origin will look like Figure 3.4. This has two immediate consequences:

- The cost function has a local maximum at the origin. This is because of the fact that it decreases along any ray passing it.
- The cost function has exactly one local maximum $\mathbf{X} = \mathbf{0}$. Because, if it had another local maximum, connecting it with a straight line to the origin, would result in a ray passing through the origin. We saw that along any ray passing the origin there is only one local maximum. This contradicts the possibility of having two local maxima.

Example 3.4

Consider the configuration $\mathbf{X} = [0, 1, 5]^T$ on a one-dimensional line. This point set produces the following EDM:

$$\tilde{\mathbf{D}} = \begin{bmatrix} 0 & 1 & 25 \\ 1 & 0 & 16 \\ 25 & 16 & 0 \end{bmatrix}.$$

As the s -stress function is invariant to translation, we can assume that $\widehat{\mathbf{X}}[1] = 0$. The level contours of the s -stress function are shown in Figure 3.5. It is evident from the figure that the cost function has a local maximum at the origin and two global minima at $\widehat{\mathbf{X}} = [0, 1, 5]^T$ and its reflection through the origin, $\widehat{\mathbf{X}} = [0, -1, -5]^T$.

Define for every $\mathbf{X} \neq \mathbf{0}$, the positive scaling $\alpha(\mathbf{X})$ which minimizes $g(\mathbf{X}, \alpha)$.

Lemma 3.3

The function $\alpha(\mathbf{X}) : \mathcal{P} \setminus \{\mathbf{0}\} \rightarrow \mathbb{R}$ is a continuous function of \mathbf{X} and $\alpha(\mathbf{X}) = \alpha(-\mathbf{X})$.

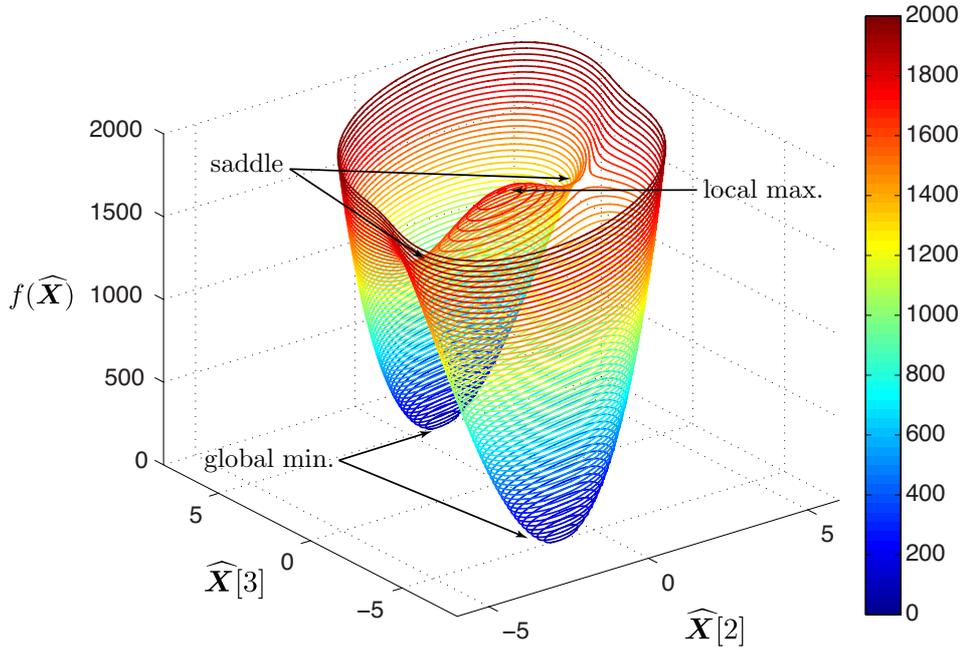


Figure 3.5: Values of the s-stress function for $\widehat{\mathbf{X}}[1] = 0$ and variable $\widehat{\mathbf{X}}[2]$ and $\widehat{\mathbf{X}}[3]$ in a one-dimensional space.

Proof.

From Lemma 3.2 we have the explicit form for $\alpha(\mathbf{X})$ as follows:

$$\alpha(\mathbf{X}) = \frac{\sqrt{\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2}}{\sqrt{\sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4}} \quad (3.6)$$

It is easy to see that $\alpha(-\mathbf{X}) = \alpha(\mathbf{X})$. For every non-zero configuration on the plane \mathcal{P} , the numerator and the denominator are continuous functions of \mathbf{X} and the denominator is non-zero. Thus, $\alpha(\mathbf{X})$ is a continuous function on $\mathcal{P} \setminus \{\mathbf{0}\}$. \square

Let us define the set \mathcal{S} as

$$\mathcal{S} = \{\mathbf{X} \in \mathcal{P} \setminus \{\mathbf{0}\} \mid \alpha(\mathbf{X}) = 1\}.$$

This is equivalent to saying that along each ray passing through the origin, we pick the minimizer and call the collection of all the minimizer configurations for different rays as \mathcal{S} .

Using (3.6), we may equivalently define the set \mathcal{S} as

$$\mathcal{S} = \left\{ \mathbf{X} \in \mathcal{P} \setminus \{\mathbf{0}\} \mid \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 = \sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4 \right\}. \quad (3.7)$$

Note that for a configuration \mathbf{X} on the set \mathcal{S} , the cost function $f(\mathbf{X})$ simplifies to

$$f(\mathbf{X}) = \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4 - \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2, \quad \text{for } \mathbf{X} \in \mathcal{S}. \quad (3.8)$$

Remark: What happens if the graph G associated with $\tilde{\mathbf{D}}$ is not connected? For simplicity assume that the graph G has two connected parts $G_i = (V_i, E_i)$, $i = 1, 2$ where $V = V_1 \cup V_2$ and $E = E_1 \cup E_2$. Let us consider a simple configuration in which $\mathbf{X}_i = C_1$ for $i \in V_1$ and $\mathbf{X}_j = C_2$ for all $j \in V_2$. We can always translate the configuration so that $\sum_i \mathbf{X}_i = \mathbf{0}$. Thus, without loss of generality we assume that our configuration is on the plane \mathcal{P} . Then the mutual distance between sensors in V_1 is zero and the same is true for sensors in V_2 . We can write the cost function as

$$f(\mathbf{X}) = \sum_{\{i,j\} \in E_1} (\|\mathbf{X}_i - \mathbf{X}_j\|^2 - \tilde{d}_{i,j}^2)^2 + \sum_{\{i,j\} \in E_2} (\|\mathbf{X}_i - \mathbf{X}_j\|^2 - \tilde{d}_{i,j}^2)^2.$$

which can be simplified to

$$f(\mathbf{X}) = \sum_{\{i,j\} \in E_1} \tilde{d}_{i,j}^4 + \sum_{\{i,j\} \in E_2} \tilde{d}_{i,j}^4.$$

Now if we look at the scaling cost function

$$g(\mathbf{X}, \alpha) = f(\alpha \mathbf{X}) = \sum_{\{i,j\} \in E_1} \tilde{d}_{i,j}^4 + \sum_{\{i,j\} \in E_2} \tilde{d}_{i,j}^4.$$

This shows that the cost function will be constant along some rays passing the origin. This can later cause some implausible cases. We avoid these degenerate cases by putting the connectivity assumption on G .

Lemma 3.4

If Condition A holds, then the set \mathcal{S} defines a bounded hyper-surface on the plane \mathcal{P} that is symmetric with respect to the origin.

Proof.

The symmetry with respect to the origin is trivial. For boundedness of \mathcal{S} , consider the cost function restricted to the set \mathcal{S} . Observe that for $\mathbf{X} \in \mathcal{S}$,

$$\begin{aligned} f(\mathbf{X}) &= \sum_{\{i,j\} \in E} (\|\mathbf{X}_i - \mathbf{X}_j\|^2 - \tilde{d}_{i,j}^2)^2 \\ &= \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4 - \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 \\ &= \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4 - \sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4 \geq 0, \end{aligned}$$

which implies that for every configuration \mathbf{X} in \mathcal{S} ,

$$\sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4 \leq \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4.$$

Using the Cauchy-Schwarz inequality we have

$$\left(\sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^2 \right)^2 \leq 2|E| \sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4 \leq 2|E| \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4.$$

Thus,

$$\sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^2 \leq \sqrt{2|E| \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4}. \quad (3.9)$$

Fix some node a in the graph. Since Condition A holds, the graph is connected, thus there exists at least one path from each node i to a . Call $P_{i,a}$ the shortest path (or any other path) between nodes i and a in the graph G . We can write

$$\mathbf{X}_i = \mathbf{X}_a + \sum_{\{e,s\} \in P_{i,a}} (\mathbf{X}_e - \mathbf{X}_s). \quad (3.10)$$

We know that on the set \mathcal{S} , $\sum_i \mathbf{X}_i = 0$. Thus,

$$-n\mathbf{X}_a = \sum_{i=1}^n \sum_{\{e,s\} \in P_{i,a}} (\mathbf{X}_e - \mathbf{X}_s).$$

Using Cauchy-Schwarz inequality,

$$\|\mathbf{X}_a\|^2 \leq \sum_{i=1}^n \sum_{\{e,s\} \in P_{i,a}} \|\mathbf{X}_e - \mathbf{X}_s\|^2. \quad (3.11)$$

Now, applying again the Cauchy-Schwarz inequality on (3.10) we have

$$\begin{aligned} \sum_{i=1}^n \|\mathbf{X}_i\|^2 &\leq n \sum_{i=1}^n \left(\|\mathbf{X}_a\|^2 + \sum_{\{e,s\} \in P_{i,a}} \|\mathbf{X}_e - \mathbf{X}_s\|^2 \right) \\ &\stackrel{(a)}{\leq} n \left(n \sum_{i=1}^n \sum_{\{e,s\} \in P_{i,a}} \|\mathbf{X}_e - \mathbf{X}_s\|^2 + \sum_{i=1}^n \sum_{\{e,s\} \in P_{i,a}} \|\mathbf{X}_e - \mathbf{X}_s\|^2 \right) \\ &= n(n+1) \sum_{i=1}^n \sum_{\{e,s\} \in P_{i,a}} \|\mathbf{X}_e - \mathbf{X}_s\|^2 \\ &\stackrel{(b)}{\leq} n(n+1) \sqrt{2|E| \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4}. \end{aligned}$$

In (a) we used (3.11) and in (b), we used (3.9). This shows that the set \mathcal{S} is bounded. \square

Lemma 3.5

Assuming that Condition A holds, there is a positive parameter D_m such that for all of the points on the set \mathcal{S}

$$\sum_{i=1}^n \|\mathbf{X}_i\|^2 \geq D_m^2.$$

Proof.

See Appendix 3.A.2. □

Lemma 3.6

The set \mathcal{S} is a closed set contained in the plane \mathcal{P} .

Proof.

From (3.7) a configuration $\mathbf{X} \in \mathcal{P} \setminus \{\mathbf{0}\}$ is in \mathcal{S} if and only if

$$\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 - \sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4 = 0. \quad (3.12)$$

Now consider a sequence of configurations $\{\mathbf{X}^{(m)}\}$ in \mathcal{S} converging to a configuration \mathbf{X} . Using the continuity of (3.12) and computing the limit as m goes to infinity we have

$$\lim_{m \rightarrow \infty} \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \left\| \mathbf{X}_i^{(m)} - \mathbf{X}_j^{(m)} \right\|^2 - \sum_{\{i,j\} \in E} \left\| \mathbf{X}_i^{(m)} - \mathbf{X}_j^{(m)} \right\|^4 = 0,$$

which implies that

$$\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 - \sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4 = 0. \quad (3.13)$$

Furthermore, the configuration $\mathbf{X}^{(m)}$ belongs to \mathcal{S} . Thus, using Lemma 3.5,

$$\lim_{m \rightarrow \infty} \left\| \mathbf{X}^{(m)} \right\|^2 = \|\mathbf{X}\|^2 \geq D_m^2 > 0.$$

Thus, the limit configuration $\mathbf{X} \neq \mathbf{0}$, satisfies (3.13) and it must belong to \mathcal{S} . This proves the closedness of the set \mathcal{S} . □

Lemma 3.7

If Condition A holds, then the set \mathcal{S} is path-wise connected. In other words for every two configurations in \mathcal{S} , there is a continuous curve lying in \mathcal{S} which connects these two configurations.

Proof.

See Appendix 3.A.3 □

Example 3.5

Let us consider a one-dimensional space and the distance matrix

$$\tilde{\mathbf{D}} = \begin{bmatrix} 0 & 0.01 & 0.16 \\ 0.01 & 0 & 0.25 \\ 0.16 & 0.25 & 0 \end{bmatrix}$$

The set \mathcal{S} for this distance matrix is shown in Figure 3.6. The color shows the value of the s -stress function at each particular point in space. Note that in this example the dimension of \mathcal{P} is 2 and thus, the set \mathcal{S} will be a one-dimensional curve lying on the plane \mathcal{P} .

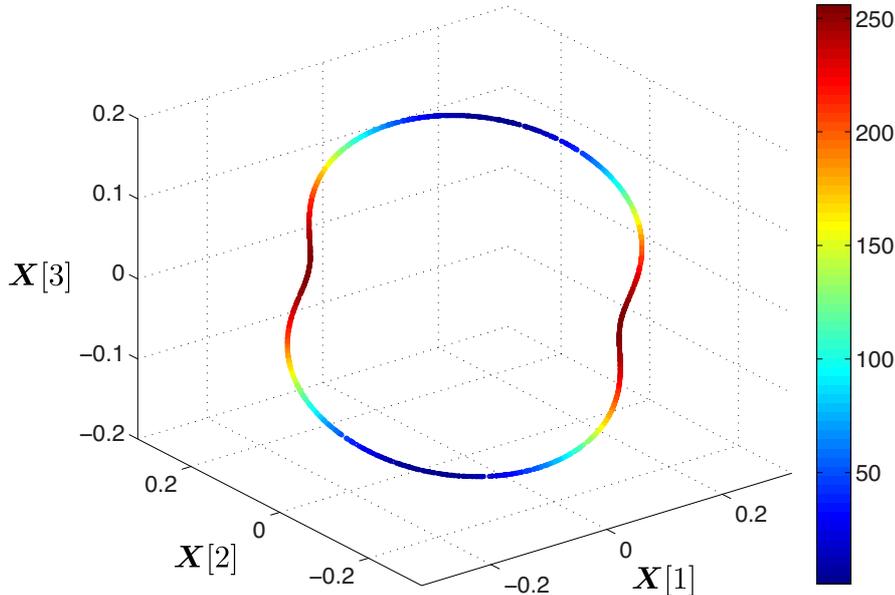


Figure 3.6: An example of the shape of the set \mathcal{S} . The color shows the value of the cost function at each particular point.

Lemma 3.8

Assuming that Condition A holds, all the local minima of the cost function $f(\mathbf{X})$ lie on \mathcal{S} .

Proof.

We know from Lemma 3.2 that the $\mathbf{0}$ configuration is a local maximum. If we consider any local minimum \mathbf{X} of the function $f(\cdot)$, obviously $\mathbf{X} \neq \mathbf{0}$ and if it is not in \mathcal{S} we can find a point in \mathcal{S} corresponding to \mathbf{X} by moving along the positive ray connecting \mathbf{X} to the origin. This implies that along this positive ray we have two local minima. This is a contradiction. \square

The result in Lemma 3.8 can help us to reduce the search space for finding the minimum of the s-stress cost function.

3.5 Does the S-stress Function Have Non-Global Minimizers?

The properties of the s-stress function that were studied in the previous section can help in better understanding of this cost function and introducing better starting points for the algorithms that minimize it. These analyses can also serve a more important purpose. There is a long standing question in the multidimensional scaling which asks for the existence of non-global minimizers of the s-stress function. While several people have come up with local minima of the stress criterion (1.8) [129], to the best of our knowledge no one has been able to design a scenario and show that a configuration is a local minimum (and not global) of the s-stress function. Malone and Trosset in [83] state that

“We are not aware that anyone has formally demonstrated the existence of non-global minimizers of s-stress. In this section [Section 2], we show that the example constructed by Trosset and Mathar (1997) for stress does not produce a non-global minimizer of s-stress. Indeed, we have been unable to construct a non-global minimizer of s-stress.”

The effort in finding local minima of the s-stress has resulted only in numerical evidence with a certain precision. For example, Žilinskas et al. in [133] consider the following measured distance matrix

$$\tilde{\mathbf{D}} = \begin{bmatrix} 0 & 1 & 1.44 & 1 \\ 1 & 0 & 1 & 1.44 \\ 1.44 & 1 & 0 & 1 \\ 1 & 1.44 & 1 & 0 \end{bmatrix},$$

And use Maple to numerically find the configurations for which the gradients are less than 10^{-10} . The proposed configuration in 2D is

$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 0.721 & 0 \\ 0 & 0.980 \\ 0.721 & 0.980 \end{bmatrix}.$$

Although the value of the gradient is very small at this configuration and the Hessian of the cost function is positive semi-definite, the authors do not provide any analytic guarantee that it is a local minimum.

We also have not been able to find an analytic design of a non-global minimizer of the s-stress criterion. In the following we provide a framework, under which we can construct an equivalent problem for the existence of non-global minimizers of the s-stress function.

Let us define another set \mathcal{E}_R as

$$\mathcal{E}_R = \left\{ \mathbf{X} \in \mathcal{P} \mid \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 = R, R \in \mathbb{R}^+ \right\}.$$

Lemma 3.9

If Condition A holds, then the set \mathcal{E}_R defines a bounded and closed hyper-surface in $\mathbb{R}^{n \times \eta}$.

Proof.

By Lemma 3.1 \mathcal{E}_0 consists of only zero configuration which is bounded. For $R > 0$,

$$\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 = R,$$

which implies that

$$\sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^2 \leq \frac{R}{d_{\min}^2}.$$

Using the same line of proof as we did in Lemma 3.6, we have

$$\sum_{i=1}^n \sum_{j=1}^n \|\mathbf{X}_i - \mathbf{X}_j\|^2 \leq n(n+1) \frac{R}{d_{\min}^2},$$

which implies the boundedness of \mathcal{E}_R for every $R > 0$. \square

Remark: Using Lemmas 3.5 and 3.9, we can see that when R is a small number $\mathcal{S} \cap \mathcal{E}_R = \emptyset$.

Note that for every configuration $\mathbf{X} \in \mathcal{S}$, we can find a radius R such that the hyper-surfaces \mathcal{S} and \mathcal{E}_R intersect at \mathbf{X} .

For every value of R , we see from (3.8) that

$$f(\mathbf{X}) = \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4 - R, \quad \text{for } \mathbf{X} \in \mathcal{E}_R \cap \mathcal{S}.$$

Also,

$$\sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4 = R, \quad \text{for } \mathbf{X} \in \mathcal{E}_R \cap \mathcal{S}.$$

Consider two functions $h_1(\mathbf{X}) = \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2$ and $h_2(\mathbf{X}) = \sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4$. Setting $h_1(\mathbf{X}) = R$ produces the set \mathcal{E}_R and $h_2(\mathbf{X}) = R$ gives $\mathcal{E}_R \cap \mathcal{S}$. Let R_{\min} be the smallest value of R such that \mathcal{E}_R and \mathcal{S} intersect. For a given distance matrix $\tilde{\mathbf{D}}$, the set \mathcal{S} is fixed. When the value of R changes the sets \mathcal{E}_R and $\mathcal{E}_R \cap \mathcal{S}$ change. We observe that the rate of growth for $h_1(\mathbf{X})$ and $h_2(\mathbf{X})$ is different (order of \mathbf{X} is 2 in h_1 and 4 in h_2). For this reason, the sets \mathcal{E}_R and \mathcal{S} will have an intersection until a certain R_{\max} .

We know from Lemma 3.8 that all the local minima of the s-stress function belong to the set \mathcal{S} . Thus, we have

$$\text{global-min}(f(\mathbf{X})) = \sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4 - R_{\max}.$$

By $\text{global-min}(f(\mathbf{X}))$ we mean its global minimum. The maximum value of $f(\mathbf{X})$ restricted to the set \mathcal{S} is $\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^4 - R_{\min}$.

Example 3.6

Consider the same setup as in Example 3.5. In this case, we have $R_{\min} = 0.019$ and $R_{\max} = 0.176$. The sets \mathcal{S} , $\mathcal{E}_{R_{\min}}$ and $\mathcal{E}_{R_{\max}}$ are shown in Figure 3.7. It is clear from the figure that $\mathcal{E}_{R_{\min}}$ intersects the set \mathcal{S} on the maximum of $f(\mathbf{X})$ and $\mathcal{E}_{R_{\max}}$ on the minimum of $f(\mathbf{X})$.

When R changes smoothly from R_{\min} towards R_{\max} , the intersection set $\mathcal{E}_R \cap \mathcal{S}$ changes. A non-global minimum happens if we observe a non-smooth change (a jump) in the intersection set $\mathcal{E}_R \cap \mathcal{S}$. This is equivalent to saying that if the number of connected components of $\mathcal{E}_R \cap \mathcal{S}$ (A connected component is a set in which there exists a connected path between each pair of configurations belonging to it) changes, we have a local minimum. Let us show the change in the number of connected components in an example. The example and the sets given in it are not related to this problem and are provided for visualization purposes only.

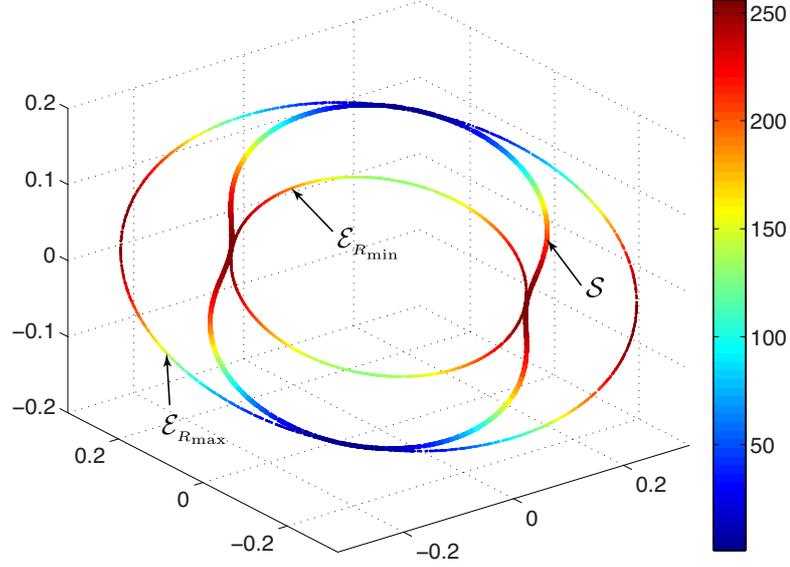


Figure 3.7: The sets \mathcal{S} , $\mathcal{E}_{R_{\min}}$ and $\mathcal{E}_{R_{\max}}$ for the setup in Example 3.6. For this setup $R_{\min} = 0.019$ and $R_{\max} = 0.177$. The colors represent the value of the cost function at each point.

Example 3.7

Take two sets \mathcal{K}_1 and $\mathcal{K}_2(R)$. \mathcal{K}_1 is fixed and shown in Figure 3.8, but $\mathcal{K}_2(R)$ expands with R . At some R_{\min} the two sets start intersecting. Each connected component of the intersection in this case is a single point. The number of connected components of $\mathcal{K}_1 \cap \mathcal{K}_2(R_{\min} + \varepsilon)$ is equal to 8. When R reaches some critical value R_c , the number of connected components drops to 6 and for $R_c + \varepsilon$ becomes 4.

In our setting a local minimum occurs if something similar to Example 3.7 happens for the intersection of the sets \mathcal{S} and \mathcal{E}_R . Thus, in order to prove (or disprove) that the s-stress function has only global minimizers we need to verify (or reject) the following hypothesis:

Hypothesis 3.1

When R changes in $[R_{\min} + \varepsilon, R_{\max} - \varepsilon]$, the number of connected components of $\mathcal{E}_R \cap \mathcal{S}$ does not change.

The question that Hypothesis 3.1 is true or false remains an open problem to us.

3.6 An Optimal English Keyboard Design

In this section we present a fun example of the applications of EDMs and the method that was presented in this chapter.

The most important goal in designing a keyboard layout is (or should be) to make it easier to type the common key combinations. Some argue that the current English keyboard layout is

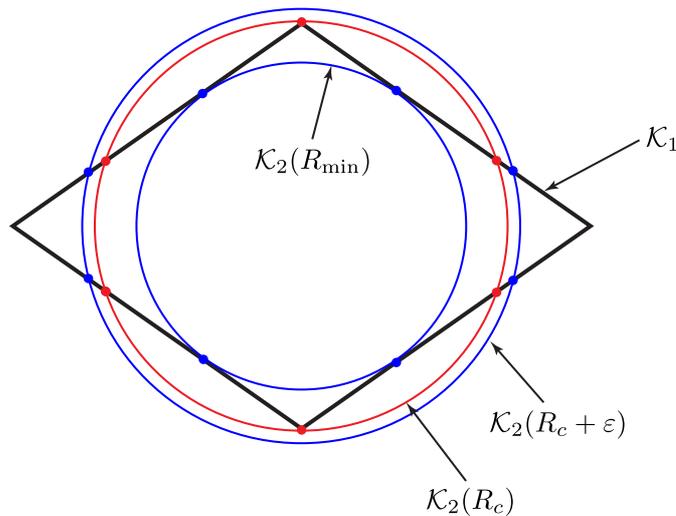


Figure 3.8: An example in which the number of connected components of the intersection of two sets changes as one of them grows.

designed to serve the opposite. The QWERTY keyboard, patented in 1878 by Latham Sholes is believed to have been designed to slow down the typist and avoid jamming the typing machine.

Assume that you can type with only one hand. The optimal layout for a keyboard in such case would be: If two alphabetic characters occur close to each other in English words, they should be located also close to each other on the keyboard, so that the fingers do not move a long distance to type the consecutive characters. We computed the average distance of each pairs of English alphabet characters in a collection of around 70,000 English words. For example in the word "love" we have $D[l, o] = 1$, $D[l, v] = 2$, $D[l, e] = 3$, and so on. The following matrix presents the average distances between each pairs of characters in the studied data-base.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0.0	2.9	3.6	3.5	3.9	3.2	3.3	3.7	3.6	3.0	2.8	3.1	3.4	3.6	4.1	3.7	3.5	3.2	4.1	3.2	3.9	3.1	2.6	3.7	4.6	3.1
b	2.9	0.0	4.2	4.4	3.7	3.9	4.2	4.1	3.5	2.9	3.8	2.8	3.4	4.6	3.1	4.7	4.2	3.3	4.3	4.0	3.0	3.9	3.5	4.0	4.4	3.5
c	3.6	4.2	0.0	4.5	4.0	3.6	5.1	3.2	3.5	3.2	2.3	4.0	4.3	4.1	3.5	4.3	4.2	3.7	4.5	4.0	3.5	4.2	4.0	4.0	4.8	5.2
d	3.5	4.4	4.5	0.0	3.2	4.3	4.3	4.4	3.3	3.2	4.0	4.1	4.2	4.1	3.7	4.4	4.4	3.7	4.6	4.6	4.2	4.0	3.7	4.0	4.6	3.6
e	3.9	3.7	4.0	3.2	0.0	3.6	3.7	3.8	4.1	3.6	2.9	3.5	3.7	3.8	4.3	3.9	3.8	3.2	3.8	3.5	4.3	2.9	3.2	2.9	5.0	3.1
f	3.2	3.9	3.6	4.3	3.6	0.0	4.3	4.0	3.0	3.6	4.2	3.3	3.9	4.2	3.1	4.0	4.4	3.1	4.7	4.0	2.9	4.4	3.6	2.8	5.2	4.3
g	3.3	4.2	5.1	4.3	3.7	4.3	0.0	3.9	3.1	3.6	4.2	3.5	4.2	2.9	3.3	4.7	6.4	3.4	4.7	4.0	3.5	4.1	4.4	4.6	3.9	3.7
h	3.7	4.1	3.2	4.4	3.8	4.0	3.9	0.0	3.6	3.6	3.8	4.2	4.1	4.1	3.3	3.1	4.8	3.6	4.0	3.3	3.7	4.1	3.2	4.2	3.9	4.2
i	3.6	3.5	3.5	3.3	4.1	3.0	3.1	3.6	0.0	3.8	2.6	3.4	3.5	3.3	3.7	4.0	3.5	3.6	3.7	3.2	3.9	2.8	2.9	3.6	4.9	2.4
j	3.0	2.9	3.2	3.2	3.6	3.6	3.6	3.6	3.8	0.0	3.3	4.8	3.4	3.9	2.8	3.2	3.2	3.4	4.7	4.2	2.1	3.8	3.1	3.4	5.7	3.8
k	2.8	3.8	2.3	4.0	2.9	4.2	4.2	3.8	2.6	3.3	0.0	3.0	3.6	3.0	2.9	3.7	4.3	3.2	3.6	3.8	3.3	4.0	3.4	3.8	3.1	4.2
l	3.1	2.8	4.0	4.1	3.5	3.3	3.5	4.2	3.4	4.8	3.0	0.0	4.2	4.5	3.5	4.1	4.6	4.4	4.3	3.9	3.5	3.5	3.2	4.0	3.0	3.2
m	3.4	3.4	4.3	4.2	3.7	3.9	4.2	4.1	3.5	3.4	3.6	4.2	0.0	4.0	3.2	3.8	4.9	3.7	4.2	4.1	3.0	5.0	3.8	3.8	4.7	4.2
n	3.6	4.6	4.1	4.1	3.8	4.2	2.9	4.1	3.3	3.9	3.0	4.5	4.0	0.0	3.3	4.7	4.5	4.2	4.3	3.7	3.6	4.3	3.6	4.5	4.8	4.0
o	4.1	3.1	3.5	3.7	4.3	3.1	3.3	3.3	3.7	2.8	2.9	3.5	3.2	3.3	0.0	3.2	4.1	3.1	4.0	3.4	3.5	3.4	2.2	3.4	4.4	3.3
p	3.7	4.7	4.3	4.4	3.9	4.0	4.7	3.1	4.0	3.2	3.7	4.1	3.8	4.7	3.2	0.0	4.3	3.1	4.2	4.3	3.6	4.9	3.7	2.9	4.9	4.6
q	3.5	4.2	4.2	4.4	3.8	4.4	6.4	4.8	3.5	3.2	4.3	4.6	4.9	4.5	4.1	4.3	0.0	4.2	3.9	4.7	1.6	4.8	4.3	3.1	6.2	4.5
r	3.2	3.3	3.7	3.7	3.2	3.1	3.4	3.6	3.6	3.4	3.2	4.4	3.7	4.2	3.1	3.1	4.2	0.0	4.2	3.4	3.2	3.3	3.2	3.7	4.2	3.6
s	4.1	4.3	4.5	4.6	3.8	4.7	4.7	4.0	3.7	4.7	3.6	4.3	4.2	4.3	4.0	4.2	3.9	4.2	0.0	3.9	3.4	4.3	3.9	4.8	4.9	5.4
t	3.2	4.0	4.0	4.6	3.5	4.0	4.0	3.3	3.2	4.2	3.8	3.9	4.1	3.7	3.4	4.3	4.7	3.4	3.9	0.0	3.6	3.9	3.4	4.0	4.0	3.7
u	3.9	3.0	3.5	4.2	4.3	2.9	3.5	3.7	3.9	2.1	3.3	3.5	3.0	3.6	3.5	3.6	1.6	3.2	3.4	3.6	0.0	4.2	3.9	3.3	5.4	4.1
v	3.1	3.9	4.2	4.0	2.9	4.4	4.1	4.1	2.8	3.8	4.0	3.5	5.0	4.3	3.4	4.9	4.8	3.3	4.3	3.9	4.2	0.0	3.6	4.4	5.0	5.2
w	2.6	3.5	4.0	3.7	3.2	3.6	4.4	3.2	2.9	3.1	3.4	3.2	3.8	3.6	2.2	3.7	4.3	3.2	3.9	3.4	3.9	3.6	0.0	2.9	3.9	3.1
x	3.7	4.0	4.0	4.0	2.9	2.8	4.6	4.2	3.6	3.4	3.8	4.0	3.8	4.5	3.4	2.9	3.1	3.7	4.8	4.0	3.3	4.4	2.9	0.0	4.4	4.9
y	4.6	4.4	4.8	4.6	5.0	5.2	3.9	3.9	4.9	5.7	3.1	3.0	4.7	4.8	4.4	4.9	6.2	4.2	4.9	4.0	5.4	5.0	3.9	4.4	0.0	3.9
z	3.1	3.5	5.2	3.6	3.1	4.3	3.7	4.2	2.4	3.8	4.2	3.2	4.2	4.0	3.3	4.6	4.5	3.6	5.4	3.7	4.1	5.2	3.1	4.9	3.9	0.0

Using this matrix, we can try to find the best configuration in a two-dimensional space (two-dimensional embedding), for the characters of English alphabet. We use Algorithm 3.1 to find

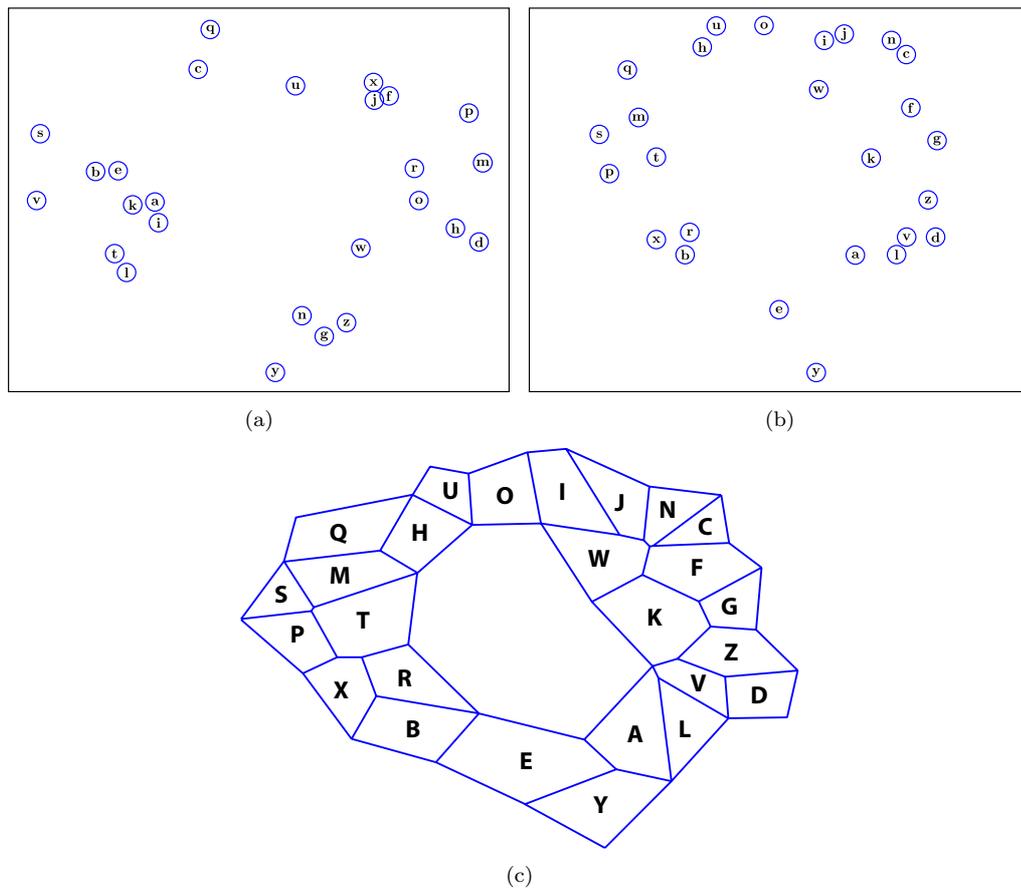


Figure 3.9: (a) shows a constructed keyboard layout for English based on the average distances of alphabet characters in English words. (b) shows the same when the frequencies of the occurrences of the pairs are used as weights to find the keyboard layout. (c) shows the same as (b) divided into Voronoi cells.

the configuration. This is shown in Figure 3.9(a). This can be a possible keyboard layout. However, there is a slight problem; for example we do not encounter the combination “xj” often in English words, but they are very close to each other in the layout.

In order to have an optimal keyboard, we need to take into account the *frequency* of each pair of characters. We collect these in the matrix \mathbf{W} .

$$\mathbf{w} = \begin{matrix} & \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} & \mathbf{e} & \mathbf{f} & \mathbf{g} & \mathbf{h} & \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{l} & \mathbf{m} & \mathbf{n} & \mathbf{o} & \mathbf{p} & \mathbf{q} & \mathbf{r} & \mathbf{s} & \mathbf{t} & \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{x} & \mathbf{y} & \mathbf{z} \\ \mathbf{a} & 6.6 & 2.1 & 5.2 & 3.3 & 9.1 & 1.0 & 2.3 & 2.6 & 8.5 & 0.2 & 0.7 & 6.2 & 3.4 & 7.1 & 5.5 & 3.1 & 0.2 & 7.1 & 5.4 & 7.2 & 3.0 & 0.9 & 0.5 & 0.3 & 1.9 & 0.3 \\ \mathbf{b} & 2.1 & 0.4 & 0.7 & 0.6 & 2.2 & 0.2 & 0.4 & 0.4 & 1.8 & 0.0 & 0.2 & 1.4 & 0.5 & 1.2 & 1.4 & 0.3 & 0.0 & 1.5 & 1.1 & 1.1 & 0.9 & 0.1 & 0.1 & 0.0 & 0.4 & 0.1 \\ \mathbf{c} & 5.2 & 0.7 & 1.9 & 1.4 & 5.5 & 0.5 & 0.6 & 1.5 & 5.0 & 0.1 & 0.4 & 2.8 & 1.4 & 3.6 & 4.2 & 1.5 & 0.1 & 3.5 & 3.0 & 3.5 & 2.0 & 0.4 & 0.1 & 0.1 & 1.1 & 0.1 \\ \mathbf{d} & 3.3 & 0.6 & 1.4 & 1.0 & 4.9 & 0.4 & 0.7 & 0.8 & 3.5 & 0.1 & 0.2 & 1.7 & 1.0 & 2.7 & 2.5 & 0.9 & 0.1 & 2.4 & 1.9 & 1.9 & 1.4 & 0.3 & 0.3 & 0.1 & 0.6 & 0.1 \\ \mathbf{e} & 9.1 & 2.2 & 5.5 & 4.9 & 10.0 & 1.5 & 2.4 & 2.8 & 9.0 & 0.2 & 0.7 & 6.5 & 3.5 & 8.8 & 6.5 & 3.8 & 0.3 & 9.4 & 8.1 & 8.6 & 3.8 & 1.6 & 0.8 & 0.5 & 1.8 & 0.3 \\ \mathbf{f} & 1.0 & 0.2 & 0.5 & 0.4 & 1.5 & 0.4 & 0.3 & 0.2 & 1.3 & 0.0 & 0.1 & 0.9 & 0.3 & 0.9 & 0.9 & 0.2 & 0.0 & 1.0 & 0.7 & 0.8 & 0.6 & 0.1 & 0.1 & 0.0 & 0.2 & 0.0 \\ \mathbf{g} & 2.3 & 0.4 & 0.6 & 0.7 & 2.4 & 0.3 & 0.5 & 0.7 & 2.5 & 0.0 & 0.1 & 1.4 & 0.6 & 2.3 & 1.7 & 0.5 & 0.0 & 1.7 & 1.2 & 1.3 & 0.9 & 0.2 & 0.2 & 0.0 & 0.5 & 0.1 \\ \mathbf{h} & 2.6 & 0.4 & 1.5 & 0.8 & 2.8 & 0.2 & 0.7 & 0.6 & 2.2 & 0.0 & 0.2 & 1.3 & 0.8 & 1.5 & 2.2 & 1.2 & 0.0 & 2.0 & 1.9 & 1.9 & 0.8 & 0.1 & 0.3 & 0.1 & 0.7 & 0.1 \\ \mathbf{i} & 8.5 & 1.8 & 5.0 & 3.5 & 9.0 & 1.3 & 2.5 & 2.2 & 6.5 & 0.1 & 0.5 & 5.8 & 3.2 & 8.4 & 6.2 & 3.1 & 0.2 & 6.3 & 6.7 & 7.8 & 3.1 & 1.3 & 0.4 & 0.3 & 1.8 & 0.4 \\ \mathbf{j} & 0.2 & 0.0 & 0.1 & 0.1 & 0.2 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 0.1 & 0.0 & 0.1 & 0.1 & 0.0 & 0.0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ \mathbf{k} & 0.7 & 0.2 & 0.4 & 0.2 & 0.7 & 0.1 & 0.1 & 0.2 & 0.5 & 0.0 & 0.1 & 0.4 & 0.1 & 0.5 & 0.4 & 0.1 & 0.0 & 0.5 & 0.4 & 0.3 & 0.2 & 0.0 & 0.1 & 0.0 & 0.1 & 0.0 \\ \mathbf{l} & 6.2 & 1.4 & 2.8 & 1.7 & 6.5 & 0.9 & 1.4 & 1.3 & 5.8 & 0.1 & 0.4 & 2.8 & 1.7 & 3.5 & 4.1 & 1.8 & 0.1 & 3.1 & 3.5 & 3.6 & 2.5 & 0.6 & 0.4 & 0.2 & 1.8 & 0.2 \\ \mathbf{m} & 3.4 & 0.5 & 1.4 & 1.0 & 3.5 & 0.3 & 0.6 & 0.8 & 3.2 & 0.0 & 0.1 & 1.7 & 0.9 & 2.3 & 2.7 & 0.9 & 0.0 & 2.2 & 2.0 & 2.2 & 1.3 & 0.2 & 0.1 & 0.1 & 0.7 & 0.1 \\ \mathbf{n} & 7.1 & 1.2 & 3.6 & 2.7 & 8.8 & 0.9 & 2.3 & 1.5 & 8.4 & 0.1 & 0.5 & 3.5 & 2.3 & 4.1 & 5.8 & 1.9 & 0.1 & 4.5 & 4.8 & 5.7 & 3.2 & 0.8 & 0.4 & 0.2 & 1.3 & 0.2 \\ \mathbf{o} & 5.5 & 1.4 & 4.2 & 2.5 & 6.5 & 0.9 & 1.7 & 2.2 & 6.2 & 0.1 & 0.4 & 4.1 & 2.7 & 5.8 & 4.3 & 2.9 & 0.1 & 5.7 & 4.9 & 5.5 & 2.6 & 0.7 & 0.5 & 0.2 & 1.6 & 0.2 \\ \mathbf{p} & 3.1 & 0.3 & 1.5 & 0.9 & 3.8 & 0.2 & 0.5 & 1.2 & 3.1 & 0.0 & 0.1 & 1.8 & 0.9 & 1.9 & 2.9 & 1.0 & 0.0 & 2.7 & 2.4 & 2.2 & 1.2 & 0.2 & 0.1 & 0.1 & 0.9 & 0.1 \\ \mathbf{q} & 0.2 & 0.0 & 0.1 & 0.1 & 0.3 & 0.0 & 0.0 & 0.0 & 0.2 & 0.0 & 0.0 & 0.1 & 0.0 & 0.1 & 0.1 & 0.0 & 0.0 & 0.1 & 0.1 & 0.1 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ \mathbf{r} & 7.1 & 1.5 & 3.5 & 2.4 & 9.4 & 1.0 & 1.7 & 2.0 & 6.3 & 0.1 & 0.5 & 3.1 & 2.2 & 4.5 & 5.7 & 2.7 & 0.1 & 3.0 & 4.4 & 5.5 & 3.0 & 0.8 & 0.5 & 0.2 & 1.5 & 0.2 \\ \mathbf{s} & 5.4 & 1.1 & 3.0 & 1.9 & 8.1 & 0.7 & 1.2 & 1.9 & 6.7 & 0.1 & 0.4 & 3.5 & 2.0 & 4.8 & 4.9 & 2.4 & 0.1 & 4.4 & 4.6 & 4.7 & 3.2 & 0.6 & 0.4 & 0.1 & 1.2 & 0.1 \\ \mathbf{t} & 7.2 & 1.1 & 3.5 & 1.9 & 8.6 & 0.8 & 1.3 & 1.9 & 7.8 & 0.1 & 0.3 & 3.6 & 2.2 & 5.7 & 5.5 & 2.2 & 0.1 & 5.5 & 4.7 & 3.5 & 2.9 & 0.7 & 0.3 & 0.2 & 1.5 & 0.2 \\ \mathbf{u} & 3.0 & 0.9 & 2.0 & 1.4 & 3.8 & 0.6 & 0.9 & 0.8 & 3.1 & 0.1 & 0.2 & 2.5 & 1.3 & 3.2 & 2.6 & 1.2 & 0.3 & 3.0 & 3.2 & 2.9 & 1.2 & 0.4 & 0.1 & 0.1 & 0.7 & 0.1 \\ \mathbf{v} & 0.9 & 0.1 & 0.4 & 0.3 & 1.6 & 0.1 & 0.2 & 0.1 & 1.3 & 0.0 & 0.0 & 0.6 & 0.2 & 0.8 & 0.7 & 0.2 & 0.0 & 0.8 & 0.6 & 0.7 & 0.4 & 0.1 & 0.0 & 0.0 & 0.2 & 0.0 \\ \mathbf{w} & 0.5 & 0.1 & 0.1 & 0.3 & 0.8 & 0.1 & 0.2 & 0.3 & 0.4 & 0.0 & 0.1 & 0.4 & 0.1 & 0.4 & 0.5 & 0.1 & 0.0 & 0.5 & 0.4 & 0.3 & 0.1 & 0.0 & 0.1 & 0.0 & 0.1 & 0.0 \\ \mathbf{x} & 0.3 & 0.0 & 0.1 & 0.1 & 0.5 & 0.0 & 0.0 & 0.1 & 0.3 & 0.0 & 0.0 & 0.2 & 0.1 & 0.2 & 0.2 & 0.1 & 0.0 & 0.2 & 0.1 & 0.2 & 0.1 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 \\ \mathbf{y} & 1.9 & 0.4 & 1.1 & 0.6 & 1.8 & 0.2 & 0.5 & 0.7 & 1.8 & 0.0 & 0.1 & 1.8 & 0.7 & 1.3 & 1.6 & 0.9 & 0.0 & 1.5 & 1.2 & 1.5 & 0.7 & 0.2 & 0.1 & 0.1 & 0.2 & 0.0 \\ \mathbf{z} & 0.3 & 0.1 & 0.1 & 0.1 & 0.3 & 0.0 & 0.1 & 0.1 & 0.4 & 0.0 & 0.0 & 0.2 & 0.1 & 0.2 & 0.2 & 0.1 & 0.0 & 0.2 & 0.1 & 0.2 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{matrix} \times \frac{1}{10}.$$

The weights are normalized, such that the maximum weight is equal to 1. Now, instead of solving the classic s-stress criterion, we solve the weighted s-stress

$$f(\mathbf{X}) = \sum_{i=1}^n \sum_{j=1}^n \mathbf{W}[i, j] \left(\|\mathbf{X}_i - \mathbf{X}_j\|^2 - \tilde{\mathbf{D}}[i, j] \right)^2.$$

Minimizing the weighted s-stress using Algorithm 3.1 results in the alphabet positions shown in Figure 3.9(b). If we divide the space into Voronoi cells (with modifications on the boundaries) we get a design shown in Figure 3.9(c). Note that the proposed layout could be useful only for typing with one hand, as in two-hand keyboards, the design is such that consecutive characters are typed with different hands to increase the typing speed. Although the provided example is far from a real design of an English keyboard, it serves as a showcase for the practical benefits of EDMs and the proposed algorithm for minimizing the s-stress function, even in the presence of weights.

3.7 Summary

In this chapter we introduced a novel simple algorithm for sensor localization and multidimensional scaling. The proposed algorithm has the following advantages:

- i*) It is distributed.
- ii*) The updates performed by individual sensors are as simple as computing the roots of a cubic polynomial.
- iii*) It converges fast and the number of sweeps until convergence is observed to be independent of the number of sensors.
- iv*) It converges to the global minimum of the s-stress function in 99.66% of the cases (in a randomized experiment) whereas the best competing methods reach only 90%.

We will exploit the proposed algorithm further in Chapters 4 and 5.

Later in the chapter, we aimed at showing some key properties of the s-stress criterion. We showed that the set of all the critical points of this cost function produce a bounded, closed

and path-wise connected hyper-surface in $\mathbb{R}^{n \times \eta}$. We finished the chapter by a hypothesis that if proves true means that the s-stress function has only global minimizers and if false, it also has non-global minimizers.

3.A Analysis

3.A.1 Proof of Lemma 3.2

Consider a fixed non-zero configuration \mathbf{X} on the plane \mathcal{P} . There must be at least two i and j for which $\mathbf{X}_i - \mathbf{X}_j \neq \mathbf{0}$, because otherwise, $\mathbf{X}_i = \mathbf{X}_j$ for every $i, j \in \{1, 2, \dots, n\}$ and considering the condition $\sum_{i=1}^n \mathbf{X}_i = \mathbf{0}$ on plane \mathcal{P} , this implies that $\mathbf{X}_i = \mathbf{0}$ for every $i \in \{1, 2, \dots, n\}$. Hence, for every non-zero configuration on \mathcal{P} , $\sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^2 > 0$. The scaling cost function

$$\begin{aligned} g(\mathbf{X}, \alpha) &= f(\alpha \mathbf{X}) \\ &= \sum_{\{i,j\} \in E} \left[\alpha^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 - \tilde{d}_{i,j}^2 \right]^2 \end{aligned}$$

is a fourth degree even polynomial of α which approaches ∞ as $|\alpha|$ grows and it has at most three local extrema. By taking the derivative it is easy to see that $g(\mathbf{X}, \alpha)$ has a maximum at $\alpha = 0$ and two local minima at

$$\alpha_1 = \frac{\sqrt{\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2}}{\sqrt{\sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4}}, \quad \alpha_2 = -\frac{\sqrt{\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2}}{\sqrt{\sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4}}. \quad (3.14)$$

Notice that α_1 is well-defined and non-zero, because $\sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4 > 0$ and $\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 > 0$ by Lemma 3.1. □

3.A.2 Proof of Lemma 3.5

Consider a specific configuration on the set \mathcal{S} and define two parameters

$$\begin{aligned} d_{\min}^2 &\stackrel{\text{def}}{=} \min_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \\ x_{\max}^2 &\stackrel{\text{def}}{=} \max_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^2. \end{aligned}$$

Both of these parameters are positive. Hence we can write

$$\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 \geq \frac{d_{\min}^2}{x_{\max}^2} \sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4. \quad (3.15)$$

We know that on the set \mathcal{S} ,

$$\sum_{\{i,j\} \in E} \tilde{d}_{i,j}^2 \|\mathbf{X}_i - \mathbf{X}_j\|^2 = \sum_{\{i,j\} \in E} \|\mathbf{X}_i - \mathbf{X}_j\|^4.$$

Using this in (3.15), we have

$$x_{\max}^2 \geq d_{\min}^2.$$

On the other hand,

$$\sum_{i=1}^n \sum_{j=1}^n \|\mathbf{X}_i - \mathbf{X}_j\|^2 \geq x_{\max}^2.$$

Since on the plane \mathcal{P} , $\sum_i \mathbf{X}_i = \mathbf{0}$,

$$\sum_{i=1}^n \sum_{j=1}^n \|\mathbf{X}_i - \mathbf{X}_j\|^2 = 2n \sum_{i=1}^n \|\mathbf{X}_i\|^2.$$

Hence,

$$\sum_{i=1}^n \|\mathbf{X}_i\|^2 \leq \frac{1}{2n} d_{\min}^2.$$

Setting $D_m = \frac{1}{\sqrt{2n}} d_{\min}$ proves the lemma. \square

3.A.3 Proof of Lemma 3.7

Consider two configurations \mathbf{X} and \mathbf{Y} in \mathcal{S} . Choose an arbitrary configuration $\tilde{\mathbf{Z}}$ in \mathcal{P} which does not lie on the ray connecting \mathbf{X} and \mathbf{Y} to the origin. In other words

$$\tilde{\mathbf{Z}} \notin \{\beta \mathbf{X} | \beta \in \mathbb{R}\} \cup \{\gamma \mathbf{Y} | \gamma \in \mathbb{R}\}$$

As $\tilde{\mathbf{Z}} \neq \mathbf{0}$, there is a point $\mathbf{Z} = \alpha(\tilde{\mathbf{Z}})\tilde{\mathbf{Z}}$ which is the scaled version of $\tilde{\mathbf{Z}}$ and lies on the set \mathcal{S} . Now consider two continuous curves

$$\begin{aligned} \tilde{\delta} : [0, 1] &\rightarrow \mathcal{P}, & \tilde{\delta}(t) &= (1-t)\mathbf{X} + t\mathbf{Z} \\ \tilde{\lambda} : [0, 1] &\rightarrow \mathcal{P}, & \tilde{\lambda}(t) &= (1-t)\mathbf{Y} + t\mathbf{Z}. \end{aligned}$$

It is easy to check that $\tilde{\delta}(t)$ and $\tilde{\lambda}(t)$ lie in \mathcal{P} for every $t \in [0, 1]$ because for example for $\tilde{\delta}$

$$\mathbf{1}^T \tilde{\delta}(t) = (1-t)\mathbf{1}^T \mathbf{X} + t\mathbf{1}^T \mathbf{Z} = 0.$$

Observe that neither $\tilde{\delta}$ nor $\tilde{\lambda}$ contains the zero configuration, because otherwise it would imply that there is a $t_0 \in [0, 1]$ for which $\tilde{\delta}(t_0) = \mathbf{0}$; t_0 cannot be 0 or 1, because it means that one of the configurations \mathbf{X} or \mathbf{Z} is the zero configuration which is a contradiction. Therefore, $t_0 \in (0, 1)$. Then $\tilde{\delta}(t_0) = (1-t_0)\mathbf{X} + t_0\mathbf{Z} = \mathbf{0}$ and it implies that $\mathbf{Z} = -\frac{1-t_0}{t_0}\mathbf{X}$. This shows that \mathbf{Z} is on the ray connecting \mathbf{X} to the origin which is again a contradiction.

As the curves $\tilde{\delta}$ and $\tilde{\lambda}$ are on $\mathcal{P} \setminus \{\mathbf{0}\}$, and by Lemma 3.4, α is continuous on $\mathcal{P} \setminus \{\mathbf{0}\}$ then the curve

$$r : [0, 2] \rightarrow \mathcal{P} \setminus \{\mathbf{0}\}, \quad \begin{cases} \alpha(\tilde{\delta}(t))\tilde{\delta}(t) & t \in [0, 1] \\ \alpha(\tilde{\lambda}(2-t))\tilde{\lambda}(2-t) & t \in (1, 2] \end{cases}$$

is a continuous curve connecting \mathbf{X} to \mathbf{Y} which completely lies in the set \mathcal{S} . This shows the path-wise connectivity of the set \mathcal{S} . \square

Chapter 4

Room Geometry Estimation and In-Room Localization Using EDMs

Truth is ever to be found in the simplicity,
and not in the multiplicity and confusion of
things.

Isaac Newton

4.1 Introduction

In Chapter 2 we used a fundamental property of Euclidean distance matrices for completing the measured EDMs in the calibration process. Then, in Chapter 3 we presented a new algorithm with fast convergence properties for sensor localization in an optimization framework. In this chapter¹ we use a combination of both techniques presented in Chapters 2 and 3 to solve an interesting problem in acoustics.

Imagine that you are blindfolded inside an unknown room. You snap your fingers and try to estimate the geometry of the room and your location by listening to the early reflections of the sound you made. In other words; can you hear the shape of a room [30]?!

A similar question was posed by Kac in 1966 [66]: Can one hear the shape of a drum? Although the question is asked in a general manner, it is specifically looking for drums having the same resonant frequencies. Gordon et al. show that the answer to this question and a similar question in astrophysics is negative [22, 44–46]. Note that although we are asking a similar

1. This chapter includes results from a collaboration with I. Dokmanić, A. Walther, Y. M. Lu and M. Vetterli [31].

question for rooms, the nature of the two approaches are fundamentally different. We are not restricting ourselves to estimating the room geometry using only the resonant frequencies. We aim at solving the problem using the room impulse response (RIR) and reflection times from the walls. This does not seem an unrealistic attempt as many animals (including bats and dolphins) and humans are able to estimate the room geometry using early reflections off the walls. Knowing the geometry of the room can help in many scenarios. Indoor sound-source localization [8, 32, 106, 108], teleconferencing, auralization and virtual reality [80] can benefit largely from known room geometry.

In this chapter we use the arrival times of early reflections (extracted from RIRs) together with the image source model to estimate the room geometry. We use properties of EDMs to find the correct combination of echoes that correspond to the same wall.

Several prior works have addressed the problem of room geometry estimation using RIRs. In [7], the authors try to solve the problem in 2D using multiple source locations. In [107] the authors address the problem by ℓ_1 -regularized template matching with a pre-measured dictionary of impulse responses. Note that such approach would require measuring a very large matrix of impulse responses. In contrast, our method works with arbitrary measurement geometries. Furthermore, we prove that the first-order echoes provide a unique description of the room for almost all setups. A subspace-based formulation allows us to use the minimal number of microphones (4 microphones in 3D). It is impossible to further reduce the number of microphones, unless we consider higher-order echoes, as attempted in [30]. However, the arrival times of higher order echoes are often challenging to obtain and delicate to use, both for theoretical and practical reasons. Therefore, in the proposed method, we use more than one microphone, avoiding the need for higher order echoes.

Later in this chapter we use the model and formulation of the above problem to solve another important problem in the field of acoustics; microphone localization. We show that if the geometry of a convex polyhedral room is known, only one source with known location is enough to estimate and track the location of a microphone inside that room.

4.2 Room Geometry Estimation Using EDMs

4.2.1 Modeling the Room

Consider a convex polyhedral room with K planar walls. The acoustic channel between a sound source and a sound receiver in a room can be modeled by the room impulse response (RIR). In an ideal case, an RIR is a train of impulses with the first impulse corresponding to the direct path between the source and the receiver and the rest of the impulses corresponding to the reflections off the walls. An RIR is often modeled as

$$h(t) = \sum_i a_i \delta(t - t_i),$$

where t_i 's are the echo arrival times. We can relate the room geometry to the echo times using the image source model [6, 15]. In this model, every echo from a wall can be modeled as a direct sound coming from a virtual source which is the mirror of the original source with respect to the same wall. An example of the image source model is shown in Figure 4.1. The time of arrival (ToA) of the echo from the i -th wall is $t_i = \|\tilde{\mathbf{s}}_i - \mathbf{r}\|/c$, where c is the speed of sound.

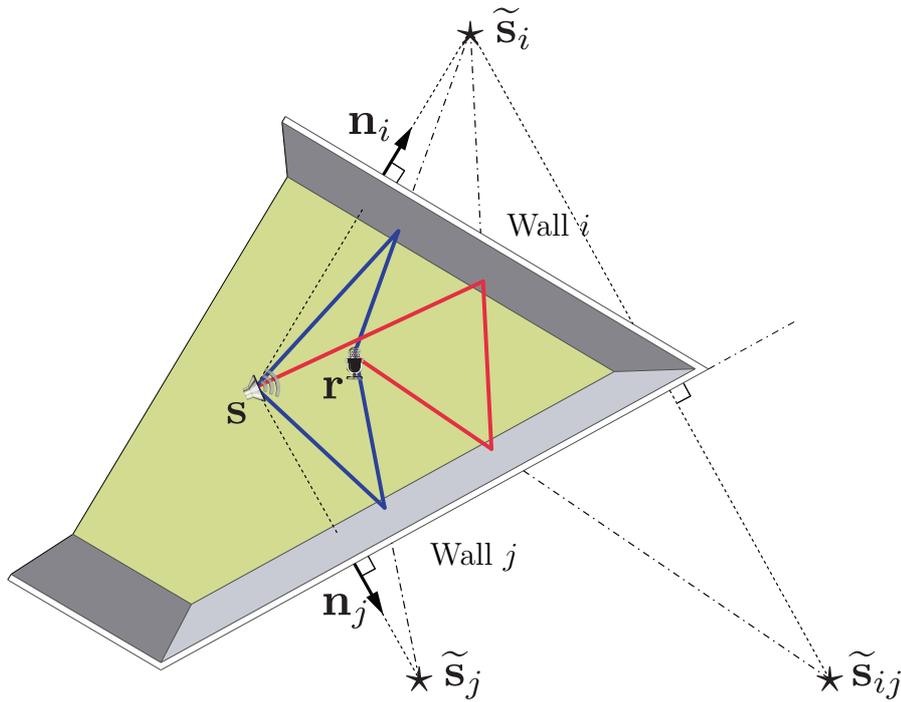


Figure 4.1: Illustration of the image source model for first and second-order echoes. Vector \mathbf{n}_i is the outward pointing unit normal associated with the i th wall. Stars denote the image sources, and $\tilde{\mathbf{s}}_{ij}$ is the image source corresponding to the second-order echo. Sound rays corresponding to first reflections are shown in blue, and the ray corresponding to the second-order reflection is shown in red.

We consider a room with a loudspeaker and an array of M microphones positioned so that they hear the first-order echoes (we typically use $M = 5$). Denote the receiver positions by $\mathbf{r}_1, \dots, \mathbf{r}_M, \mathbf{r}_m \in \mathbb{R}^3$ and the source position by $\mathbf{s} \in \mathbb{R}^3$.

If we have more than three microphones in the room, we can estimate the location of the sound source using the direct sound [10, 76, 119]. Afterwards, knowing the positions of the image sources is equivalent to knowing the position of the walls (a wall is the plane halving the line segment between the source and the receiver and perpendicular to it). Thus, we can simplify the problem by locating the image sources instead of the walls.

In order to locate an image source we need to extract out the echoes corresponding to it from the RIRs of the microphones. The challenge is that these echoes are unlabeled in each RIR. For example the echo from a wall might not arrive to a microphone in the same order as it arrives to other microphones. This is shown visually in Figure 4.2. The echoes corresponding to $\tilde{\mathbf{s}}_1$ do not reach the two microphones in the same order. Thus, the task is to identify the right combination of echoes that correspond to the same image source.

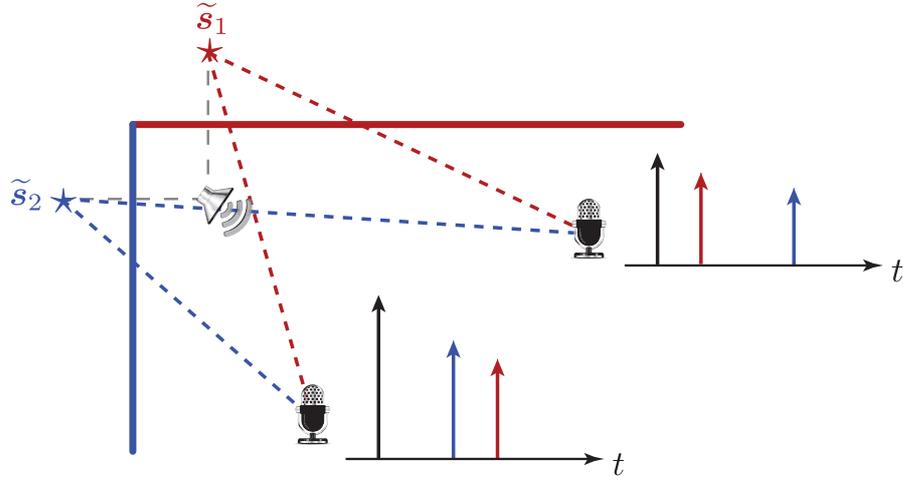


Figure 4.2: In order to estimate the position of the image sources we need to identify their corresponding echoes in each RIR. The challenge is that the echoes might get swapped. This figure shows an example that the echoes from \tilde{s}_1 and \tilde{s}_2 do not arrive to the microphones in the same order.

4.2.2 Echo Labeling Using EDMs

Let $\mathbf{D} \in \mathbb{R}^{M \times M}$ be a matrix whose entries are squared distances between microphones, $\mathbf{D}[i, j] = \|\mathbf{r}_i - \mathbf{r}_j\|_2^2$, $1 \leq i, j \leq M$. Matrix \mathbf{D} is an EDM corresponding to the microphone setup.

When the loudspeaker emits a sound, each microphone receives the direct sound and K first-order echoes from the K walls. The arrival times of the received echoes are proportional to the distances between image sources and microphones.

Our solution for the echo labeling problem is based on the rank property of EDMs. Recall from Theorem 1.2 that in a 3-dimensional space, the rank of an EDM is at most 5. We assume that we know the distances between the microphones; this is not an unrealistic assumption as we setup the measurement system for hearing the shape of the room. Thus, the matrix \mathbf{D} is a known EDM with rank at most 5. When we extract peaks from the the RIRs of the microphones, some of them might not correspond to an image source (they are falsely chosen) and for those that do, we do not know from which image source they are coming. We select one echo from each microphone's RIR. The aim is to find the collection of echoes that come from the same wall. Let us explain the echo labeling procedure with the help of Figure 4.3. With each selection of echoes we augment the EDM \mathbf{D} . For example in the figure two possible augmentations are shown. $\mathbf{D}_{\text{aug},1}$ is a plausible augmentation of \mathbf{D} because all the distances correspond to a single image source, and they appear in the correct order. This matrix is an EDM and has a rank at most 5. The second matrix, $\mathbf{D}_{\text{aug},2}$, is the result of an incorrect echo assignment, as it contains entries coming from different walls. A priori, we cannot tell whether the *red* echo comes from Wall 1 or from Wall 2. It is simply an unlabeled peak in the RIR recorded by microphone 1. However, the augmented matrix $\mathbf{D}_{\text{aug},2}$ is not an EDM and has a rank larger than 5, so we conclude that the corresponding combination of echoes is not correct. In other words, wrong assignments lead to augmentations of \mathbf{D} that are not EDMs. In particular, these augmentations do not have the

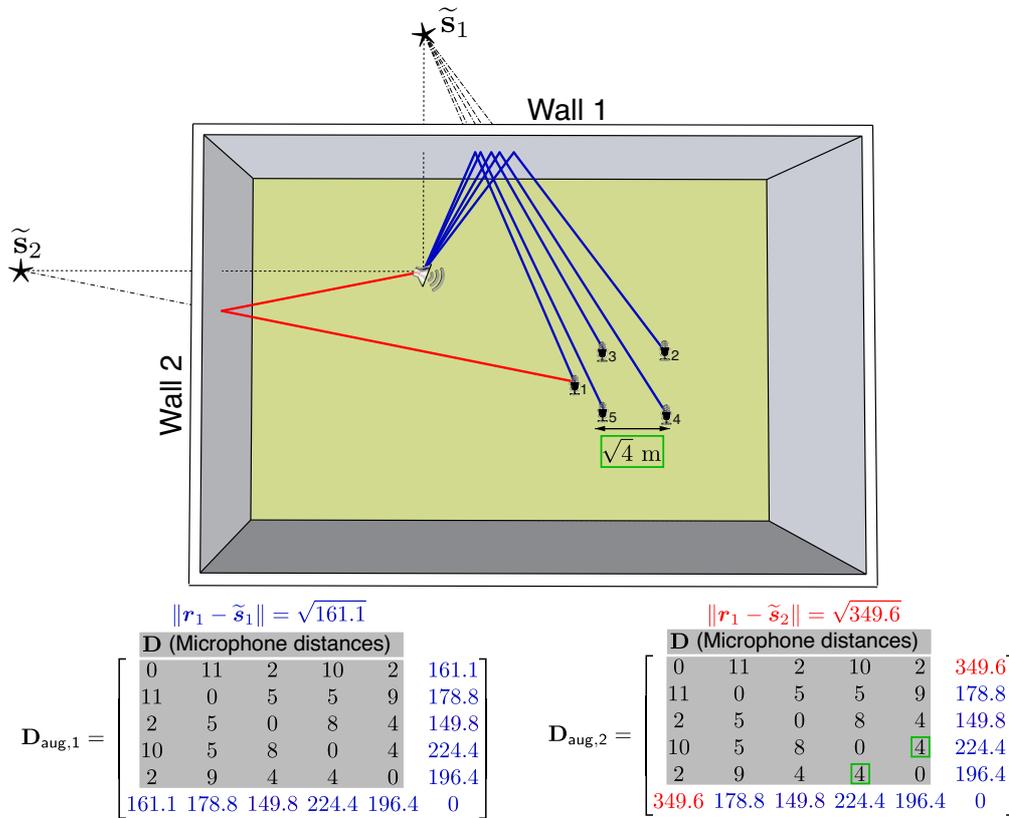


Figure 4.3: Microphones receive the echoes from all the walls, and we aim to identify echoes coming from a single wall. We select one echo from each microphone and use these echoes to augment the EDM of the microphone setup, \mathbf{D} . If all the selected echoes come from the same wall, the augmented matrix is an EDM as well. In the figure, $\mathbf{D}_{\text{aug},1}$ is an EDM since it contains the distances to a single point $\tilde{\mathbf{s}}_1$. $\mathbf{D}_{\text{aug},2}$ contains a wrong distance (shown in red) for microphone 1, so it is not an EDM.

correct rank.

More formally, let \mathbf{e}_m list the candidate distances computed from the RIR recorded by the m -th microphone. We proceed by augmenting the matrix \mathbf{D} with a combination of M unlabeled squared distances $\mathbf{d}_{(i_1, \dots, i_M)}$ to get \mathbf{D}_{aug} ,

$$\mathbf{D}_{\text{aug}}(\mathbf{d}_{(i_1, \dots, i_M)}) = \begin{bmatrix} \mathbf{D} & \mathbf{d}_{(i_1, \dots, i_M)} \\ \mathbf{d}_{(i_1, \dots, i_M)}^T & 0 \end{bmatrix}.$$

The column vector $\mathbf{d}_{(i_1, \dots, i_M)}$ is constructed as

$$\mathbf{d}_{(i_1, \dots, i_M)}[m] = \mathbf{e}_m^2[i_m],$$

with $i_m \in \{1, \dots, \text{length}(\mathbf{e}_m)\}$. In words, we construct a candidate combination of echoes \mathbf{d} by selecting one echo from each microphone. Note that $\text{length}(\mathbf{e}_m) \neq \text{length}(\mathbf{e}_n)$ for $m \neq n$ in general. That is, we can pick different number of echoes from each microphone.

Algorithm 4.1 Hearing the shape of a room using first-order echoes.

Input: Microphone EDM \mathbf{D} , and e_m , $m \in \{1, \dots, M\}$.

Output: Shape of the room.

1. For every $\mathbf{d}_{(i_1, \dots, i_M)}$, $\text{score}[\mathbf{d}_{(i_1, \dots, i_M)}] \leftarrow \text{s-stress}(\mathbf{D}_{\text{aug}})$.
 2. Sort the scores collected in score .
 3. Compute the image source locations.
 4. Remove image sources that do not correspond to walls (higher-order by their geometric dependencies to first order echoes and the “ghost” sources by heuristics).
 5. Reconstruct the room.
-

If $\text{rank}(\mathbf{D}_{\text{aug}}) \leq 5$ or more specifically \mathbf{D}_{aug} verifies the EDM property, then the selected combination of echoes corresponds to an image source, or equivalently to a wall. Even if this approach requires testing all the echo combinations, in practical cases the number of combinations is small enough that this does not pose a problem.

4.2.3 Practical Room Geometry Estimation Algorithm

Due to several uncertainties in the measurement process the rank test that was explained in the previous section is not applicable in practical scenarios. This is because of the measurement noise in the echo arrival times, sampling resolution, precision in measuring the distances of the microphone array, etc. Because of the above limitations, we need to find an alternative to the rank test of \mathbf{D}_{aug} . Note that the rank property is not the only characteristic of EDMs (refer to Section 1.2). A natural approach is to check how close \mathbf{D}_{aug} is to an EDM. To this end, we use the s-stress criterion that we studied in great detail in Chapter 3. Let us write the cost function again; Given an observed noisy matrix $\tilde{\mathbf{D}}_{\text{aug}}$, $\text{s-stress}(\tilde{\mathbf{D}}_{\text{aug}})$ is the value of the following optimization program,

$$\text{s-stress}(\tilde{\mathbf{D}}_{\text{aug}}) = \underset{\mathbf{D}_{\text{aug}} \in \text{EDM}_{(3)}}{\text{minimize}} \sum_{i,j} \left(\mathbf{D}_{\text{aug}}[i, j] - \tilde{\mathbf{D}}_{\text{aug}}[i, j] \right)^2. \quad (4.1)$$

By $\text{EDM}_{(3)}$ we denote the set of EDMs with embedding dimension equal to 3. We say that $\text{s-stress}(\tilde{\mathbf{D}}_{\text{aug}})$ is the score of the matrix $\tilde{\mathbf{D}}_{\text{aug}}$, and use it to assess the likelihood that a combination of echoes corresponds to a wall. In order to solve (4.1) we use the alternate coordinate descent method introduced in Section 3.3.

Combining the described ingredients, we design an algorithm for estimating the shape of a room. The algorithm takes as input the arrival times of echoes at different microphones (computed from RIRs). For every combination of echoes, it computes the score using the s-stress criterion. For the highest ranked combinations of echoes, it computes the image source locations. We employ an additional step to eliminate invalid echoes, second-order image sources and other impossible solutions. This is summarized in Algorithm 4.1.

It is not necessary to test all echo combinations. An echo from a fixed wall will arrive at all the microphones within the time given by the largest inter-microphone distance. Therefore, it suffices to combine echoes within a temporal window corresponding to the array diameter. This substantially reduces the running time of the algorithm. Consequently, we can be less conservative in the peak-picking stage.

We can also find the right echo combinations based on a simple linear condition. The details of the method are out of the scope of this thesis. Interested readers are referred to [31].

4.2.4 Uniqueness

In the previous section we introduced an algorithm to reconstruct the geometry of a convex polyhedral room from first order echoes collected by five microphones. Can we guarantee that only one room corresponds to the set of selected first order echoes? In order to answer this question, we need to define a *good* room-microphone array combination. Since our algorithm needs the knowledge of first order echoes, we need to restrict ourselves to the cases in which we can hear them.

Definition 4.1 (*Feasibility*)

Given a room \mathcal{R} and a loudspeaker position \mathbf{s} , we say that the point $\mathbf{x} \in \mathcal{R}$ is feasible if a microphone placed at \mathbf{x} can receive all the first-order echoes of a pulse emitted from \mathbf{s} .

Using the above definition, we can state the unicity theorem as follows.

Theorem 4.1

Consider a room with a loudspeaker and $M \geq 4$ microphones placed uniformly at random inside the feasible region. Then the unlabeled set of first-order echoes uniquely specifies the room with probability 1. In other words, almost surely exactly one assignment of first-order echoes to walls describes a room.

Proof.

See Appendix 4.A.1. □

This means that we can reconstruct any convex polyhedral room if the microphones are in the feasible region.

4.2.5 Experimental Evaluations

We ran the experiments in two distinctly different environments. One set was conducted in a lecture room at EPFL, where our modeling assumptions are approximately satisfied. Another experiment was conducted in a portal of the Lausanne cathedral. The portal is non-convex, with numerous non-planar reflecting objects. It essentially violates the modeling assumptions, and the objective was to see whether the algorithm still gives useful information. In all experiments, microphones were arranged in an arbitrary geometry, and we measured the distances between the microphones approximately with a tape measure. We did not use any specialized equipment or microphone arrays. Nevertheless, the obtained results are remarkably accurate and robust.

The lecture room is depicted in Figure 4.4(a). Two walls are glass windows, and two are gypsum-board partitions. The room is equipped with a perforated metal plate ceiling suspended below a concrete ceiling. To make the geometry of the room more interesting, we replaced one wall by a wall made of tables. Results are shown for two positions of the table wall and two different source types. We used an off-the-shelf directional loudspeaker (Genelec 8030A), an omni-directional loudspeaker (Lange D12A dodecahedron) and five non-matched omni-directional microphones (Behringer ECM 8000). The RIRs were estimated by the sine sweep technique [38]. In the first experiment, we used an omni-directional loudspeaker to excite the room, and the

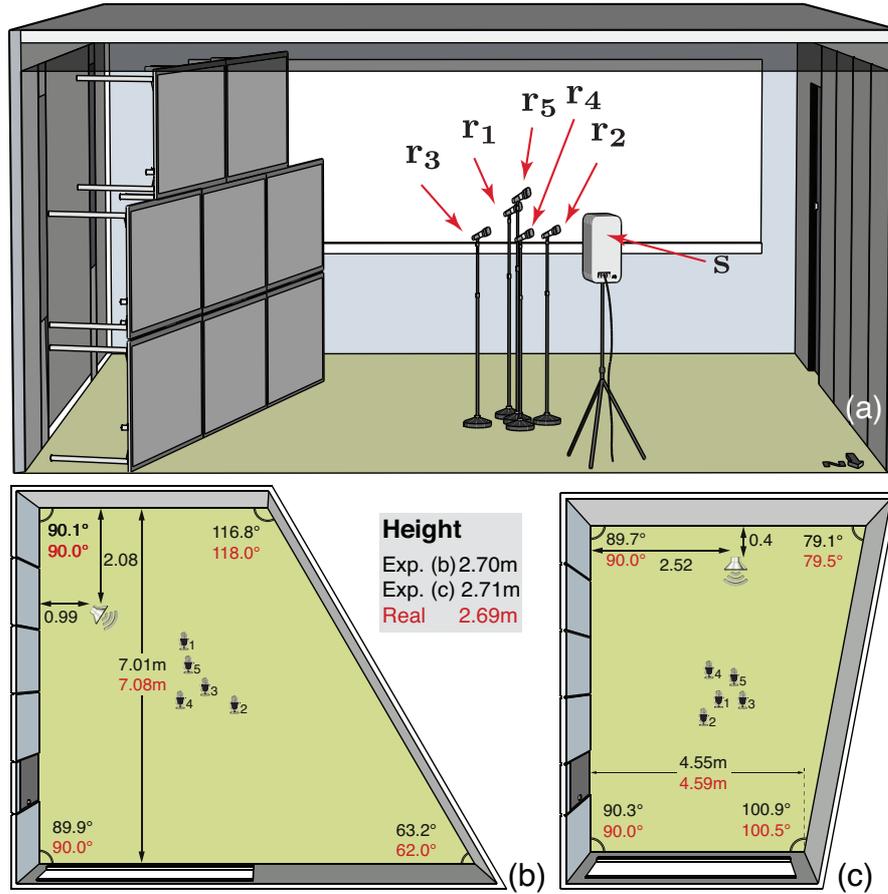


Figure 4.4: (a) Illustration of the room used in the experiment with a movable wall. (b, c) Two reconstruction results. Real values are indicated in red, and the estimated values are indicated in black.

algorithm reconstructed all six walls correctly, as shown in Figure 4.4(b). Note that the floor and the ceiling are estimated near-perfectly. In the second experiment, we used a directional loudspeaker. As the power radiated to the rear by this loudspeaker is small, we placed it against the north wall, thus avoiding the need to reconstruct it. Surprisingly, even though the loudspeaker is directional, the proposed algorithm reconstructs all the remaining walls accurately, including the floor and the ceiling.

Figures 4.5(a,b) show a panoramic view and the floor plan of the portal of the Lausanne cathedral. The central part is a pit reached by two stairs. The side and back walls are closed by glass windows, with their lower parts in concrete. In front of each side wall, there are two columns, and the walls are joined by column rows indicated in the figure. The ceiling is a dome approximately 9 meters high. We used a directional loudspeaker placed at point 'L' in Figure 4.5(c). Microphones were placed around the center of the portal. In spite of the complex room structure with obstacles in front of walls, the proposed algorithm correctly retrieves the

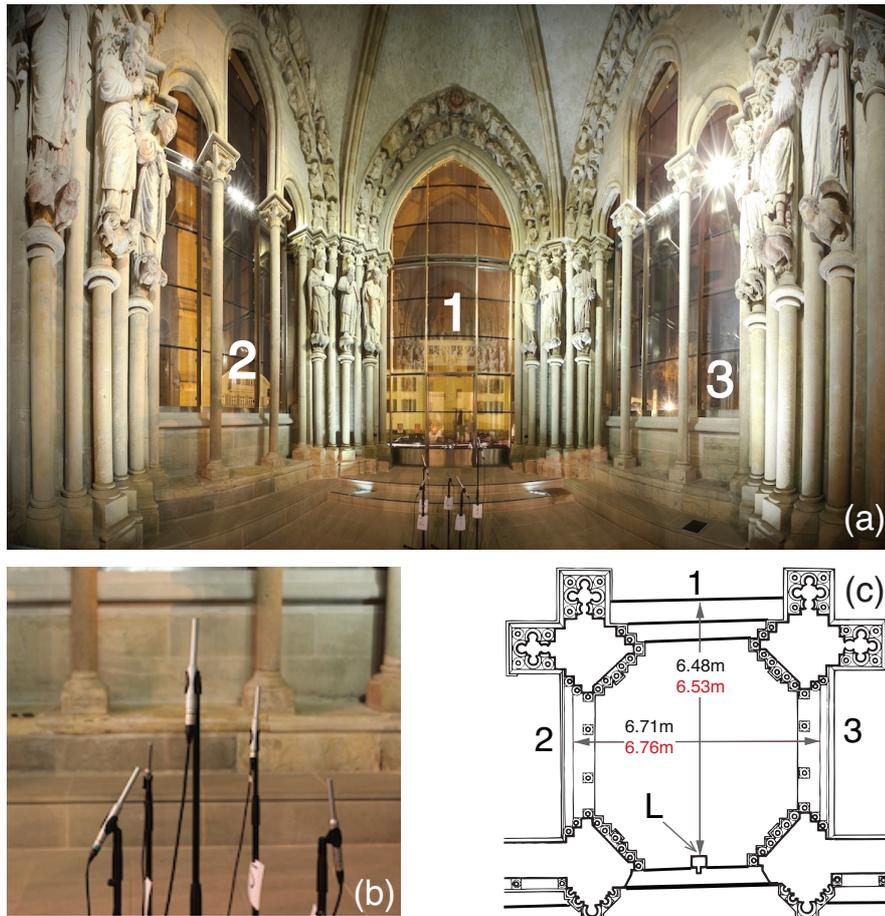


Figure 4.5: (a) Panoramic photo of the portal of the Lausanne cathedral. (b) A close up of the microphone array used in cathedral experiments. (c) Floor plan of the portal. Real values are indicated in red, and the estimated values are indicated in black.

three glass walls and the floor. Alas, in this case we do not have a way to remove unwanted image sources. Even though the RIRs contain many echoes from numerous objects in the portal, our algorithm successfully groups the echoes corresponding to the walls and the floor, certifying the robustness of the method. More details about the experiments are given in the SI Appendix of [31].

4.3 In-Room Microphone Localization Using EDMs

In the previous section we saw how to find the geometry of a convex polyhedral room given five microphones and a loudspeaker. In this section we address a dual problem; we assume that we know the geometry of the room and also the position of the loudspeaker and would like to localize the microphone.

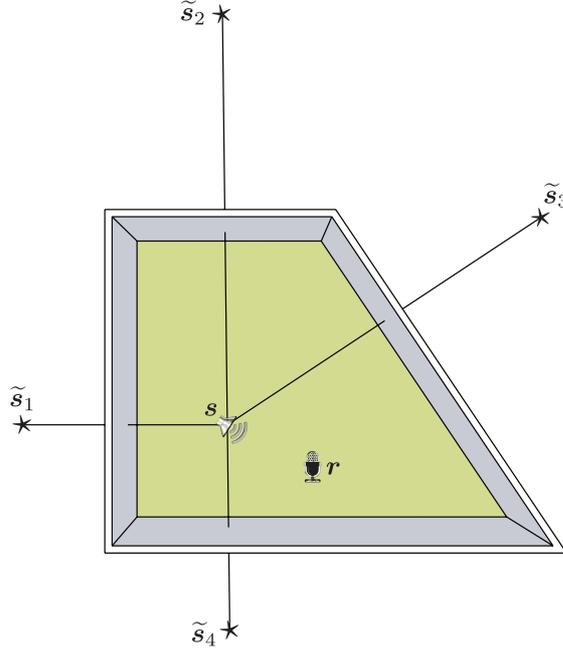


Figure 4.6: The image source model for a single loudspeaker and a microphone inside a known room. If the position of the loudspeaker is known, the image sources are known as well. The image sources for the floor and the ceiling are skipped for better visualization.

There are several approaches in the literature to calibrate the position of a microphone (or a microphone array) inside a room [11, 97, 102]. This problem is also similar to source localization in acoustics [10, 117]. In this section, we propose an alternative approach for microphone localization using the foundations established so far.

Consider a known room \mathcal{R} with K planar walls and a loudspeaker with a known position inside the room. An example setup is shown in Figure 4.6. As the room and the loudspeaker position are known, we know the position of the image sources. Call \mathbf{D} , the distance squared matrix with elements

$$\mathbf{D}[i, j] = \begin{cases} \|\tilde{\mathbf{s}}_i - \tilde{\mathbf{s}}_j\|^2 & i, j \geq 2 \\ \|\mathbf{s} - \tilde{\mathbf{s}}_j\|^2 & i = 1 \\ \|\tilde{\mathbf{s}}_i - \mathbf{s}\|^2 & j = 1 \end{cases}, \quad (4.2)$$

where $\tilde{\mathbf{s}}_i$ are the image source locations. The matrix \mathbf{D} is known a-priori. When we extract the echoes from microphone \mathbf{r} , if we know which echo corresponds to which image source the localization will be trivial. However, the echoes that are extracted from the RIR of the microphone are unlabeled. Again we are facing a labeling problem.

4.3.1 Echo Labeling for Microphone Localization

Similar to Section 4.2.2, let \mathbf{e} list the candidate distances computed from the RIR recorded by the microphone. We proceed by augmenting the matrix \mathbf{D} with a combination of K unlabeled

squared distances $\mathbf{d}_{(i_1, \dots, i_K)}$ to get \mathbf{D}_{aug} ,

$$\mathbf{D}_{\text{aug}}(\mathbf{d}_{(i_1, \dots, i_K)}) = \begin{bmatrix} \mathbf{D} & \mathbf{d}_{(i_1, \dots, i_K)} \\ \mathbf{d}_{(i_1, \dots, i_K)}^T & 0 \end{bmatrix}.$$

The column vector $\mathbf{d}_{(i_1, \dots, i_K)}$ is constructed as

$$\mathbf{d}_{(i_1, \dots, i_K)}[k] = \mathbf{e}^2[i_k],$$

with $i_k \in \{1, \dots, \text{length}(\mathbf{e})\}$. In words, we construct a candidate combination of echoes \mathbf{d} by selecting K echoes out of all extracted echoes from the microphone. Note that $\text{length}(\mathbf{e}) \neq K$ in general, meaning that we might pick more than K echoes from the RIR of the microphone.

If $\text{rank}(\mathbf{D}_{\text{aug}}) \leq 5$ or more specifically \mathbf{D}_{aug} verifies the EDM properties, then the selected combination of echoes is the correct permutation. Again note that even if this approach requires testing all the echo combinations, in practical cases the number of combinations is small enough that this does not pose a problem.

4.3.2 Practical Echo Labeling for Microphone Calibration

Both the measurements for \mathbf{D} and \mathbf{e} are often noisy. Thus, the rank test is not suitable for this application. We try to find the \mathbf{D}_{aug} that is the closest to an EDM. Similar to Section 4.2.3, for each selection of echoes that results in \mathbf{D}_{aug} , we find the value $\text{s-stress}(\mathbf{D}_{\text{aug}})$ with the algorithm presented in Section 3.3. The combination of echoes which results in the minimum value for the s-stress score is selected as the correct permutation. The algorithm is summarized as

- i) For every $\mathbf{d}_{(i_1, \dots, i_K)}$, $\text{score}[\mathbf{d}_{(i_1, \dots, i_K)}] \leftarrow \text{s-stress}(\mathbf{D}_{\text{aug}})$
- ii) find the minimum score collected in score ,
- iii) Compute the microphone location.

Also note that while finding the s-stress score, we actually estimate the position of the microphone at the same time. Thus, no more processing is needed to estimate this position after finding \mathbf{D}_{aug} with the minimum score.

4.3.3 Experimental Evaluations

In order to evaluate the theory for in-room localization of microphones, we use the data from experiment shown in Figure 4.4(c). The room dimensions are known a-priori and the loudspeaker location was measured during the experiment. The experimental setup with the image sources of the loudspeaker are shown in Figure 4.7. As the loudspeaker is placed against the north wall, we do not consider the image source for this wall. We extract the echo times from the RIR of the microphone as shown in Figure 4.8. The matrix \mathbf{D} — defined in (4.2) — is

$$\mathbf{D} = \begin{matrix} & \mathbf{s} & \tilde{\mathbf{s}}_1 & \tilde{\mathbf{s}}_2 & \tilde{\mathbf{s}}_3 & \tilde{\mathbf{s}}_4 & \tilde{\mathbf{s}}_5 \\ \mathbf{s} & \begin{bmatrix} 0 & 25.40 & 178.48 & 5.91 & 4.66 & 10.38 \\ 25.40 & 0 & 203.90 & 55.40 & 30.07 & 35.77 \\ 178.48 & 203.90 & 0 & 172.38 & 183.15 & 188.86 \\ 5.91 & 55.40 & 172.38 & 0 & 10.58 & 16.28 \\ 4.66 & 30.07 & 183.15 & 10.58 & 0 & 28.94 \\ 10.38 & 35.77 & 188.86 & 16.28 & 28.94 & 0 \end{bmatrix} \end{matrix}.$$

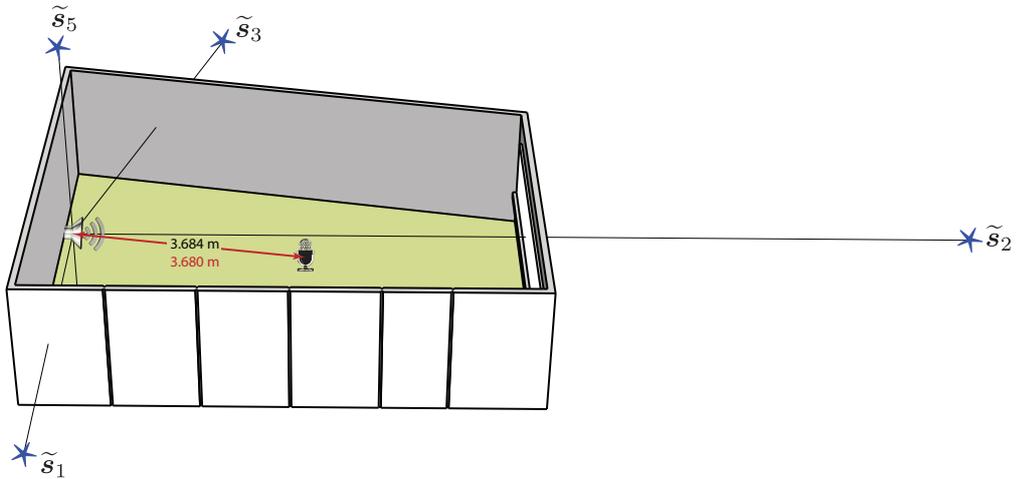


Figure 4.7: Sketch of a room in the EPFL campus where the in-room localization experiment is performed. The image sources of the loudspeaker are shown with stars. The image source of the floor (\tilde{s}_4) is not shown for better visualization. The distance of the loudspeaker and the microphone is shown in red while the estimated distance is in black.

We augment this matrix with 6-tuples of echoes selected from the microphone’s RIR. For each combination we find the value of $s\text{-stress}(\mathbf{D}_{\text{aug}})$. The combination that results in the minimum score is selected as the correct combination and the microphone position is found using the estimated permutation of the echoes. As it is shown in Figure 4.7 the distance of the microphone from the loudspeaker is estimated with less than 1 *cm* error.

4.4 Summary

Using the rank property of Euclidean distance matrices (Theorem 1.2) we proposed a novel method for estimating the shape of an unknown convex polyhedral room. The main advantages

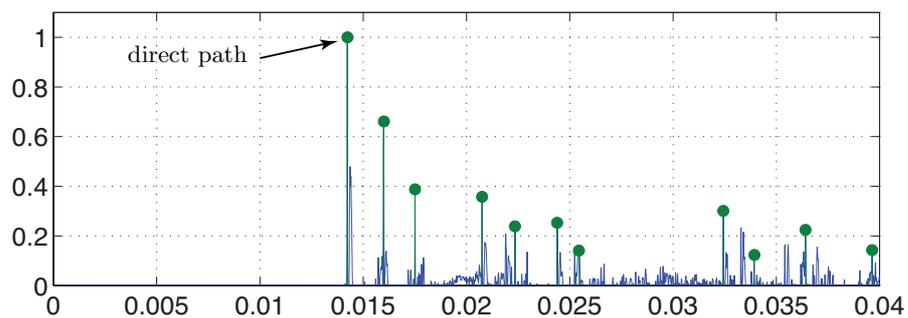


Figure 4.8: The room impulse response and the echoes chosen from it for indoor localization of the microphone. The first peak corresponds to the direct path from the loudspeaker to the microphone.

of the method are its simplicity, its ability to find the room shape uniquely and robustness against model mismatch. In the noisy case, we used the proposed algorithm for minimization of the s-stress function from Section 3.3 to evaluate each combination of the echoes. The s-stress function enabled us to give a score for each combination and pick those with the best score.

Using the same geometrical arguments, we could also design a method for microphone localization inside a known room with only one known loudspeaker. Through several experiments, we showed that the proposed methods for hearing the shape of a room and in-room localization are accurate and remarkably stable even when the room does not necessarily satisfy the model conditions.

4.A Analysis

4.A.1 Proof of Theorem 4.1

It is sufficient to prove the claim for $M = 4$. Cases when $M > 4$ follow by considering any subset of four microphones. Draw independently and uniformly at random microphone locations $\mathbf{r}_1, \dots, \mathbf{r}_4$ in the feasible region. To this particular choice of microphone locations we correspond vectors \mathbf{y}_k and $\tilde{\mathbf{y}}_k$ as follows,

$$y_{k,m} \stackrel{\text{def}}{=} \|\tilde{\mathbf{s}}_k - \mathbf{r}_m\|^2 = \|\tilde{\mathbf{s}}_k\|^2 - 2\tilde{\mathbf{s}}_k^T \mathbf{r}_m + \|\mathbf{r}_m\|^2,$$

and

$$\tilde{y}_{k,m} \stackrel{\text{def}}{=} -\frac{1}{2}(y_{k,m} - \|\mathbf{r}_m\|^2),$$

where $\tilde{\mathbf{s}}_k$ is the location of the image source with respect to Wall k . We have in vector form

$$\begin{bmatrix} \tilde{y}_{k,1} \\ \tilde{y}_{k,2} \\ \tilde{y}_{k,3} \\ \tilde{y}_{k,4} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \\ \mathbf{r}_4^T \end{bmatrix} \begin{bmatrix} -\frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{s}}_k \\ \|\tilde{\mathbf{s}}_k\|^2 \end{bmatrix}, \quad \text{or} \quad \tilde{\mathbf{y}}_k = \mathbf{R}\tilde{\mathbf{u}}_k.$$

Thanks to the condition $\sum_{m=1}^4 \mathbf{r}_m = \mathbf{0}$, we have that

$$\mathbf{1}^T \tilde{\mathbf{y}}_k = -\frac{M}{2} \|\tilde{\mathbf{s}}_k\|^2 \quad \text{or} \quad \|\tilde{\mathbf{s}}_k\|^2 = -\frac{2}{M} \sum_{m=1}^M \tilde{y}_{k,m}.$$

The image source is found as

$$\tilde{\mathbf{s}}_k = \mathbf{S}\tilde{\mathbf{y}}_k,$$

where \mathbf{S} is a matrix satisfying

$$\mathbf{S}\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

It follows that

$$\frac{2}{4} \mathbf{1}^T \tilde{\mathbf{y}}_k + \|\mathbf{S}\tilde{\mathbf{y}}_k\|^2 = 0. \quad (4.3)$$

Vector $\tilde{\mathbf{y}}_k$ corresponds to the k th wall, or k th image source (it is the correct permutation). We now show that wrong permutations cannot satisfy (4.3). We do it by replacing one, two, or three entries in $\tilde{\mathbf{y}}_k$ by wrong values and arguing that these are not good combinations. We choose

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{R}^\dagger,$$

where \mathbf{R}^\dagger is the Moore-Penrose pseudo-inverse of \mathbf{R} . With this choice, any column submatrix of \mathbf{S} with $n \leq 3$ columns is rank n with probability one.

1. (*1 replacement*). Without loss of generality, let us replace the 4-th entry of $\tilde{\mathbf{y}}_k$ ($\tilde{y}_{k,4}$), by $\tilde{y}_{k',4}$, $k' \neq k$, and plug it into (4.3). We can rewrite the equation as

$$\alpha + \beta \tilde{y}_{k',4} + \gamma \tilde{y}_{k',4}^2 = 0, \quad (4.4)$$

where α , β and γ do not functionally depend on $\tilde{y}_{k',4}$, and $\gamma \neq 0$ with probability 1. For any realization of $\tilde{y}_{k,1}, \dots, \tilde{y}_{k,3}$, the distribution of $\tilde{y}_{k',4}$ is continuous, thus the probability that it assumes any given value is zero (note that this is not true for $\tilde{y}_{k,4}$ —for echoes coming from the same wall, knowing three of them constraints the fourth to two possible values). Therefore the probability that $\tilde{y}_{k',4}$ is one of at most two real roots of (4.4) is zero.

2. (*2 replacements*). Now we replace $\tilde{y}_{k,3}$ and $\tilde{y}_{k,4}$ by $\tilde{y}_{k',3}$ and $\tilde{y}_{k'',4}$. We can have either a) $k' = k''$ or b) $k' \neq k''$. We rewrite (4.3) as

$$[\tilde{y}_{k',3} \quad \tilde{y}_{k'',4}] \mathbf{A} \begin{bmatrix} \tilde{y}_{k',3} \\ \tilde{y}_{k'',4} \end{bmatrix} + \mathbf{a}^T \begin{bmatrix} \tilde{y}_{k',3} \\ \tilde{y}_{k'',4} \end{bmatrix} + a = 0, \quad (4.5)$$

where $\mathbf{A} = \mathbf{S}[:, 3:4]^T \mathbf{S}[:, 3:4]$ (with Matlab notation) is full rank with probability 1 and is positive semi-definite. Also, \mathbf{A} , \mathbf{a} and a do not functionally depend on $[\tilde{y}_{k',3} \quad \tilde{y}_{k'',4}]^T$. Locus of the roots of (4.5) is an ellipse. But for any realization of $\tilde{y}_{k,1}$ and $\tilde{y}_{k,2}$ the distribution of $[\tilde{y}_{k',3} \quad \tilde{y}_{k'',4}]^T$ is continuous over some two-dimensional subset of \mathbb{R}^2 both in cases a) and b). Therefore the probability that it takes a value on the root ellipse of (4.5) is zero.

3. (*3 replacements*). Here we replace $\tilde{y}_{k,2}, \tilde{y}_{k,3}, \tilde{y}_{k,4}$ by $\tilde{y}_{k',2}, \tilde{y}_{k'',3}, \tilde{y}_{k''',4}$. If $k' = k'' = k'''$, then the argument is the same as in the case of one replacement. If $k' = k''$ or $k' = k'''$ or $k'' = k'''$, but not all three are equal, then we can just repeat the argument for the case of 2 replacements (b). Finally if they are all different, we write

$$[\tilde{y}_{k',2} \quad \tilde{y}_{k'',3} \quad \tilde{y}_{k''',4}] \mathbf{B} \begin{bmatrix} \tilde{y}_{k',2} \\ \tilde{y}_{k'',3} \\ \tilde{y}_{k''',4} \end{bmatrix} + \mathbf{b}^T \begin{bmatrix} \tilde{y}_{k',2} \\ \tilde{y}_{k'',3} \\ \tilde{y}_{k''',4} \end{bmatrix} + b = 0. \quad (4.6)$$

Again $\mathbf{B} = \mathbf{S}[:, 2:4]^T \mathbf{S}[:, 2:4]$ is full rank with probability 1, so the locus of the roots of (4.6) is an ellipsoid. The set of values that $[\tilde{y}_{k',2} \quad \tilde{y}_{k'',3} \quad \tilde{y}_{k''',4}]$ takes is again some three-dimensional region in \mathbb{R}^3 and the probability that the triplet takes value on an ellipsoid is zero.

In conclusion, almost surely only one (correct) combination of echoes satisfies (4.3), so almost surely only one room corresponds to collected first-order echoes.

Chapter 5

Microphone Array Calibration with Diffuse Noise

One person's data is another person's noise.

K. C. Cole

5.1 Introduction

Growing use of smart phones in people's daily interactions makes researchers think of ideas to incorporate them in different applications. One of the main fields to use smart phones is for speech acquisition as a microphone array (e.g., in conference rooms). A great challenge to reach this goal, however, is to estimate the geometry of the ad-hoc network of the microphones in the measurement space. Knowing the relative location of the microphones enables one to remove the echoes and other effects of the room from speech and deliver a higher quality sound¹.

We can classify the common microphone calibration techniques in three categories:

With Loudspeaker(s) and Known Source Signals An a-priori known source is played from one or multiple loudspeakers to perform the calibration. Sachar et al. [110] present an experimental setup using a pulsed acoustic excitation generated by five domed tweeters. The ToFs between the loudspeakers and microphones are used to find the relative geometry. Raykar et al. [103] use a maximum length sequence or chirp signal in a distributed computing platform. The time difference of arrival of microphone signals are then computed by cross-correlation and used for estimating the microphone locations. Since the original signal is known, these techniques are

1. This chapter is the result of a collaboration with M. J. Taghizadeh, P. N. Garner and H. Bourlard [121].

robust to noise and reverberation. These techniques are the most common ones and are the basis for finding the time-of-flights. We saw an example of such methods in Chapter 2. The microphone localization technique introduced in Chapter 4, also falls into this category.

With Loudspeaker(s) and Unknown Source Signals These methods do not assume a-priori knowledge of the sent signals. Microphone calibration is usually integrated with source localization. Flanagan and Bell [39] propose a method using the Weiss-Friedlander technique where the sensor location and direction of arrival of the sources are estimated alternately until the algorithm has converged. Another approach is proposed in [24] by introducing an energy based method for joint microphone calibration and speaker localization. The energy of the signal is computed and a nonlinear optimization problem is formulated to perform maximum likelihood estimation of the source sensor positions. This method requires several active sources for accurate localization and calibration. These methods are useful for microphone calibration with an active speaker where the speech signal is unknown.

Without Loudspeaker(s) and Unknown Source Signals Recently, a third approach is proposed by McCowan and Lincoln [88] using the characteristics of a diffuse noise field. A diffuse noise field is characterized by noise signals that propagate with equal probability in all directions. An interesting property of diffuse noise field is used to estimate the mutual distances of the sensors. As the present noise in the environment is approximately (and not exactly) diffuse, this method needs careful post-processing for accurate localization.

The state-of-the-art techniques are usually applicable for conventional small microphone arrays. Estimation of the pairwise distances becomes unreliable when the distances between the microphones are large. Some methods have been already proposed for calibration of large ad-hoc microphone arrays (one example is MDS-MAP explained in Chapter 2). In this chapter we use the methods provided in Chapters 2 and 3 for obtaining more accurate localization. Using an improved version of the method proposed in [88], we estimate the pairwise distances of the microphones in close proximity based on the coherence of their recorded signals in a diffuse noise field. We show that a simple averaging among the noise frames yields more accurate estimates and speeds up the algorithm. Then, by adding constraints to the matrix completion method we try to further improve the completion results for microphone array calibration. We show that imposing EDM characteristics on matrix completion using the Cadzow algorithm improves the robustness and accuracy of extracting the ad-hoc microphones geometry. Although adding more constraints to matrix completion, makes it difficult to provide theoretical performance bounds on the method as in Chapter 2, but it helps to improve the accuracy of reconstruction.

5.2 Distance Estimation in a Diffuse Noise Field

5.2.1 Coherence in Diffuse Noise Fields

Let us start by the definition of a diffuse noise field:

Definition 5.1 (*Diffuse noise field [88]*)

A diffuse noise field is characterized by noise signals that propagate with equal probability in all directions and have equal power at all locations.

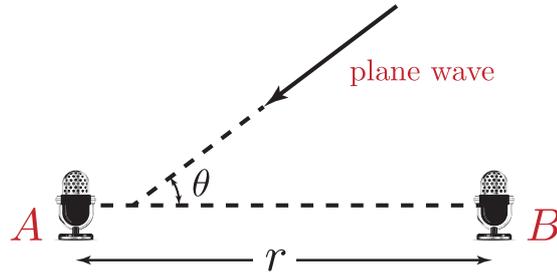


Figure 5.1: A plane wave hitting two microphone at positions A and B .

Cook in [25] presents the following analysis: consider two microphones in positions A and B , with distance r from each other. The correlation coefficient between the sound pressures at the two points is defined as

$$\gamma_{A,B} = \frac{\frac{1}{T} \int_0^T x_A(t)x_B(t)dt}{\left(\int_0^T x_A(t)^2 dt \int_0^T x_B(t)^2 dt\right)^{1/2}},$$

where $x_A(t)$ and $x_B(t)$ are the instantaneous sound pressures in points A and B and T is some large value. Suppose that a plane wave with wave number k passes from points A and B with angle θ (see Figure 5.1). It is easy to verify that

$$\gamma_{A,B} = \cos(kr \cos \theta).$$

In a diffuse sound field, we assign the same weights for plane waves coming from all directions. Thus, if the two microphones are in a three-dimensional space, then

$$\bar{\gamma}_{A,B} = \frac{1}{4\pi} \int_0^\pi \int_0^{2\pi} \cos(kr \cos \theta) \sin \theta d\phi d\theta = \frac{\sin(kr)}{kr}.$$

This is the cross-correlation coefficient for the sound pressure at two points with distance r . When there are more than one frequencies, we can perform the same analysis with the Fourier transform of the signals: The coherence $\Gamma_{A,B}(\omega)$ between two microphone signals is defined as

$$\Gamma_{A,B}(\omega) = \frac{\Phi_{A,B}(\omega)}{\sqrt{\Phi_A(\omega)\Phi_B(\omega)}}, \quad (5.1)$$

where $\Phi_{A,B}(\omega)$ is the cross-spectral density between microphone signals $X_A(\omega)$ and $X_B(\omega)$ (Fourier transforms of $x_A(t)$ and $x_B(t)$) defined as [124]:

$$\Phi_{A,B}(\omega) = \mathbb{E}[X_A(\omega)X_B(\omega)^*]. \quad (5.2)$$

In a diffuse noise field, we have

$$\Gamma_{A,B}(\omega) = \text{sinc}\left(\frac{\omega r}{c}\right), \quad (5.3)$$

where c is the sound speed in the medium.

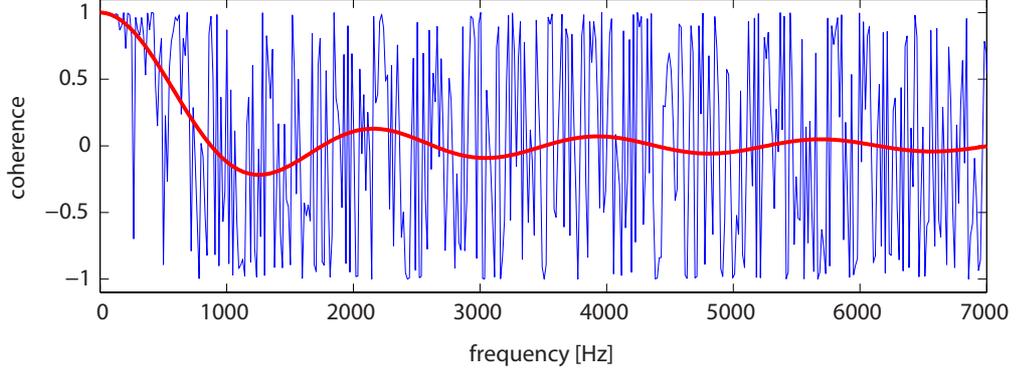


Figure 5.2: Fitting a sinc function on one frame of the diffuse noise coherence; the correct distance is 20 cm and the estimated distance is 19.3 cm

5.2.2 Distance Estimation from Coherence Information

Equation (5.3) is of great importance as it can help in estimating the distance of two microphones in a diffuse noise field. A simple way to extract the distance information is to divide the microphone signals into several frames in time and apply the above computation to each frame to estimate the mutual distances. This is the approach taken in [88].

Assume we are interested in finding the mutual distance of microphones l and k . In order to fit a sinc function to the coherence of the recorded signals in frame number j , we use the simple quadratic objective function:

$$\epsilon_{l,k}^j(d) = \sum_{\omega_{\min}^j}^{\omega_{\max}^j} \left| \Re\{\Gamma_{l,k}(\omega)\} - \text{sinc}\left(\frac{\omega d}{c}\right) \right|^2$$

$$\hat{d}_{l,k}^j = \arg \min_d \epsilon_{l,k}^j(d).$$

where $[\omega_{\min}^j, \omega_{\max}^j]$ defines the frame support in which the fitting is performed and j is the frame number.

The pairwise distances can be estimated from each frame of the noise signal. After the distance is estimated in each frame, the authors in [88] use a k-means clustering to classify the distances versus their corresponding cost value into two classes. The cluster with the lower cost mean is chosen as the winner and the distance is estimated as the center of the cluster. An example of distance extraction from one frame in real measurements is shown in Figure 5.2. Although the estimated distance is close to the actual one, it is clear from the figure that the estimation cannot be accurate using one frame estimations. We further improve the method for removing the outliers proposed in [88] by adding another stage of clustering to the system. We add a Gaussian mixture model (GMM) classifier to the system to enable separating overlapping clusters. After clustering, the cluster with the minimum variance (as opposed to minimum mean objective function) is chosen as the winner class. An example of such clustering with real measurements is shown in Figure 5.3.

Another simple approach for estimating the distance from several frames is to use a 2-

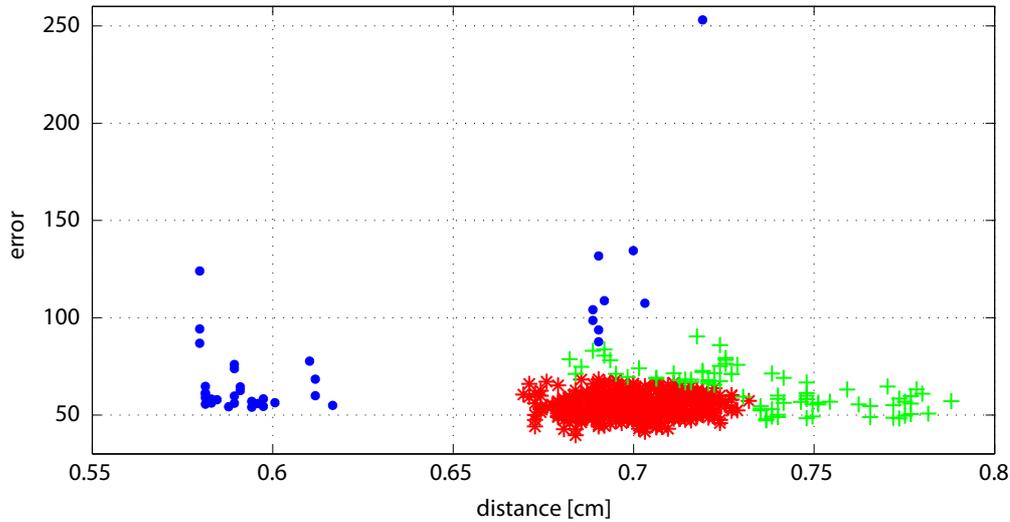


Figure 5.3: Example of k-means followed by GMM clustering for outlier detection for distance estimation between two microphones in a diffuse noise field. Each data point represents an estimation of the distance from one single frame. Blue dots have large errors, while green crosses have more variance and red stars are the winners. The actual distance is 70 cm and the estimated pairwise distance is 69.9 cm.

dimensional histogram of the distance points. This approach also shows a robust performance in our experiments. An example of a simple histogram is shown in Figure 5.4.

The clustering methods that we saw above indeed improve the distance estimation results presented in [88] for large distance values. However, there is an important fact that is neglected in the analysis of [88]; the expectation in (5.2). This expectation can be performed over several frames. To this end, we simply find the coherence function from the average coherence over several frames. Figure 5.5 demonstrates an example from real measurement data. The coherence coefficients are averaged over 100 temporal frames of the noise signals. The fitting is significantly improved in this case with a simple averaging operation.

5.3 Microphone Calibration in Diffuse Noise Field

In real scenarios (as we will see in the next section) the distance estimates in diffuse noise fields become noisy and unreliable as the distances grow, while the estimations for small distances are good. In such scenarios we can simply neglect the measurements for large distances, and estimate them using properties of EDMs. We saw in Chapter 2 how to use the rank property of EDMs to estimate missing distance data. As we have seen in Chapter 2, not all low-rank matrices are necessarily EDMs. In this section we incorporate more properties of distance matrices in estimating the missing data.

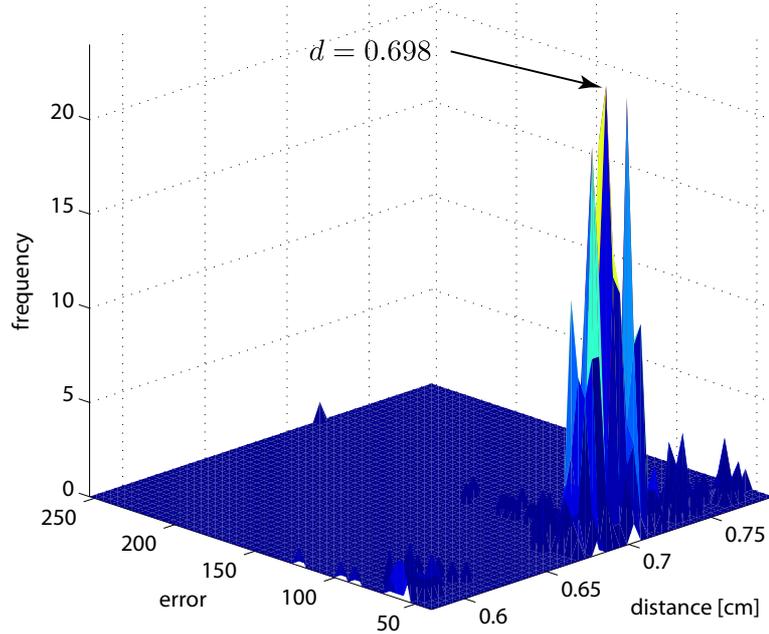


Figure 5.4: Example of two-dimensional histogram clustering for distance estimation between two microphones. The actual distance is 70 cm and the estimated distance is 69.8 cm.

5.3.1 Cadzow Projection to EDM Properties

The classic matrix completion algorithm as described in Section 2.4.1 recovers a low-rank matrix which does not necessarily correspond to an EDM. We have seen that an EDM is symmetric with diagonal elements equal to zero. These properties are not incorporated in a basic matrix completion algorithm. In order to have the outputs of the completion process closer to an EDM we modify the matrix completion algorithm used in Section 2.4.1; we apply a Cadzow-like method. The Cadzow (also known as Papoulis-Gershberg) algorithm [19] is a method for finding a signal which satisfies a composite of properties by alternately projecting the signal into the property sets. We modify the OPTSPACE algorithm by inserting an extra step in its iterations. Recall that in the classic version of this algorithm a simple rank- q approximation ($q = 4$ for the distance matrix of a two-dimensional microphone array) is used as the starting point for the iterations using gradient descent on (2.2). After each step of the gradient descent, we apply the transformation $\phi : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ on the new matrix to make sure that the output satisfies the following properties *sequentially*:

- ◇ It is symmetric: We replace each element by the average of it and its symmetric counterpart.
- ◇ It has a zero diagonal: We set the diagonal values equal to zero.
- ◇ It has only non-negative elements (we set all the negative elements to zero).
- ◇ It is rank- q .

Thus, the modified iteration can be summarized in two steps (refer to (2.2)):

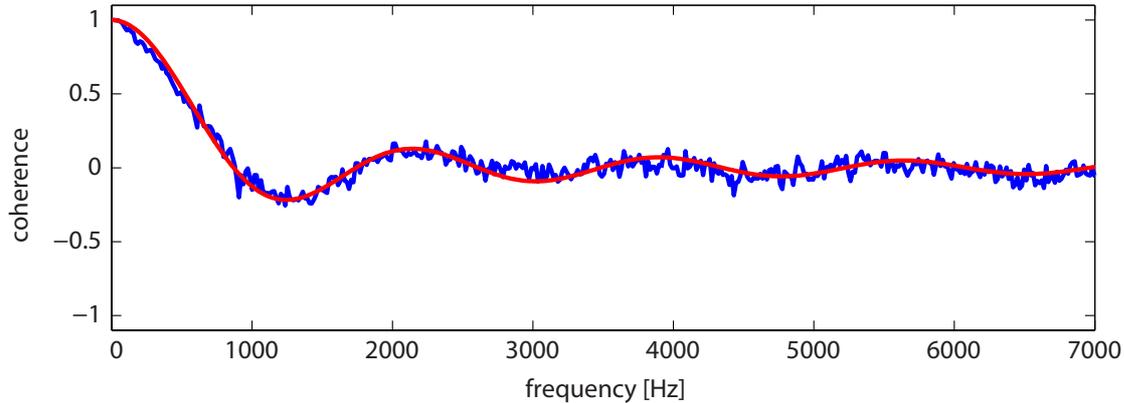


Figure 5.5: Fitting a sinc function on average of 100 frames of diffuse noise coherence; The correct distance is 20 cm and the estimated distance is 19.8 cm

– iteration $k + 1/2$:

$$\begin{aligned}\mathbf{X}^{k+1/2} &= \mathbf{X}^k + \mu \frac{\partial F(\mathbf{X}^k, \mathbf{Y}^k)}{\partial \mathbf{X}} \\ \mathbf{Y}^{k+1/2} &= \mathbf{Y}^k + \mu \frac{\partial F(\mathbf{X}^k, \mathbf{Y}^k)}{\partial \mathbf{Y}} \\ \mathbf{S}^{k+1/2} &= \arg \min_{\mathbf{S}} \mathcal{F}(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{S})\end{aligned}$$

– iteration $k + 1$:

$$(\mathbf{X}^{k+1}, \mathbf{Y}^{k+1}, \mathbf{S}^{k+1}) = \text{svd} \left(\phi(\mathbf{X}^{k+1/2} \mathbf{S}^{k+1/2} \mathbf{Y}^{k+1/2 T}), \mathbf{q} \right),$$

where μ is the step size found using line search.

In the next section, we compare the performance of this simple modification to the classic OPTSPACE and also to the s-stress criterion studied in Chapter 3.

5.4 Experimental Evaluations

5.4.1 Diffuse Noise Recording Set-up

We use the geometrical setup of the MONC corpus [89]. Twelve microphones are located in a planar area (i.e., two-dimensional space): eight of them are located on a circle with diameter 20 cm while one microphone is located at the center of the circle. There are three additional microphones at 70 cm from the central microphone. The microphones are Sennheiser MKE-2-5-C omnidirectional miniature lapel microphones. The floor is covered with carpet and surrounded with plaster walls having two big windows. The enclosure is a $8 \times 5.5 \times 3.5 \text{ m}^3$ rectangular room and is moderately reverberant. It contains a centrally located $4.8 \times 1.2 \text{ m}^2$ rectangular table. The sampling rate is 48 kHz.

In addition to the real recordings, we simulated the scenario described above to enable some evaluations in a controlled set-up. The results on the simulated data indicate the performance

bounds of the methods. In the simulations, we consider 32 white Gaussian noise sources distributed in the room. The room impulse responses are generated with the image source model [5] using intra-sample interpolation up to 15th order reflections. The corresponding reflection ratio, β used by the image source model was calculated via Eyring's formula:

$$\beta = \exp\left(\frac{-13.82}{c \times (L_x^{-1} + L_y^{-1} + L_z^{-1}) \times T}\right),$$

where L_x, L_y and L_z are the room dimensions, c is the speed of sound in the air and T is the room reverberation time. In our experiments, $T = 300 \text{ ms}$ and the direct-path propagation is discarded from the impulse response for generating a diffuse noise field [51].

5.4.2 Microphone Calibration with Simulated Data

In order to estimate the pairwise distances, we take two microphone signals of length 2.14 s and split them into short windows of length 1024 samples using a Tukey function (parameter = 0.25) and apply Fourier transform. For each frame, we compute the coherence function through (5.1). The average of the coherence functions are computed and used for estimation of the pairwise distances by fitting a sinc function. This procedure is repeated for all pairs of microphones to construct an estimated distance matrix. The geometry of the array is extracted using the s-stress method introduced in Chapter 3. Figure 5.6(a) illustrates the microphone calibration results. The reconstruction error is $\text{dist}(\mathbf{X}, \widehat{\mathbf{X}}) = 1.43$ as defined in (2.3), whereas the error using the method proposed in [88] is 4 (all the errors reported in this chapter are the value of $\text{dist}(\mathbf{X}, \widehat{\mathbf{X}})$). In addition, our averaging method speeds up the calibration by a factor of 60 compared to [88] since the k-means clustering is not required to identify the accurate frames.

To perform evaluations for a scenario of having microphones at long distances, we consider the three additional microphones. The proposed method enables a reasonable estimate up to 73 cm (this value is obtained empirically). We confirm empirically that the distances beyond that are not reliably estimated so we regard them as *missing*. Thereby, the following entries of the Euclidean distance matrix are missing, $d_{10,11}, d_{10,12}, d_{1,10}, d_{7,10}, d_{8,10}, d_{12,11}, d_{5,11}, d_{6,11}, d_{7,11}, d_{3,12}, d_{4,12}, d_{5,12}$ (see Figure 5.6(b)). Microphone calibration is achieved in two steps. In the first step, the s-stress method is used to find 9 close microphones. These microphones are then used as anchor points. In the second step, either the s-stress or the matrix completion method is used to calibrate the full network. Figure 5.6(b) demonstrates the results. The estimated error for the s-stress and the matrix completion methods are 65 and 33 respectively.

5.4.3 Microphone Calibration with Real Data

In the first step, the geometry of the 9 microphones are estimated. To further improve the performance, we use a two-dimensional (errors vs. pairwise distances) histogram to remove the outliers. The resolution of the bins is chosen to be 2.5 mm. The *green* curve in Figure 5.7(a) depicts the results based on our averaging method on 100 frames. The estimated error is 8.04. The *blue* curve shows the improved estimates by a hybrid of averaging and outlier removal using 1000 frames where the averaging method is applied on five frames to estimate the pairwise distances and construct the two-dimensional histogram. The estimated error in this case is 5.

To estimate the full network geometry where the pairwise distances of far away microphones are missing, we apply the proposed EDM completion technique and compare the results with the

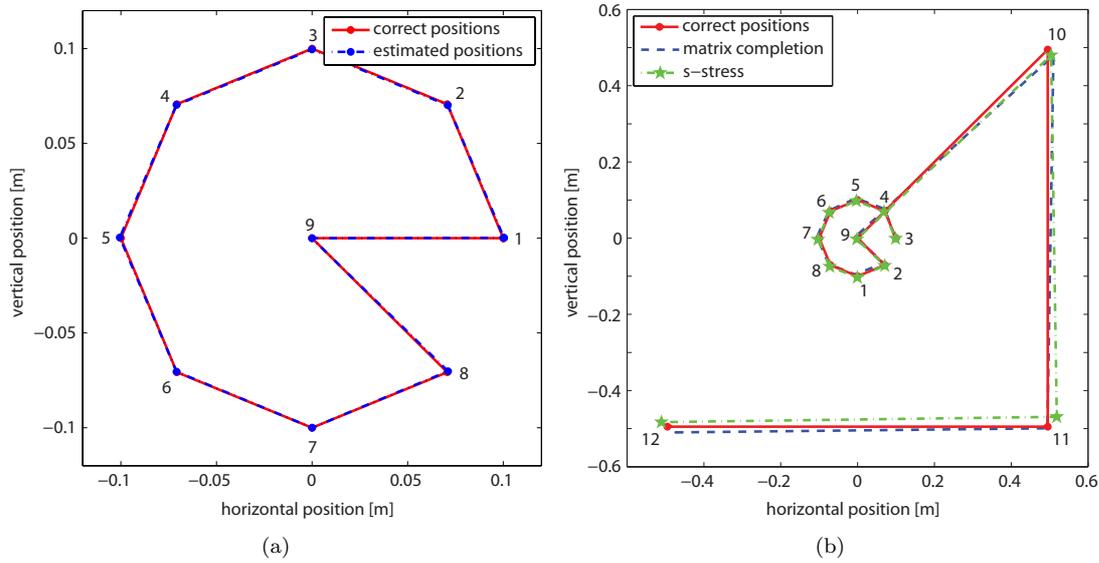


Figure 5.6: (a) Simulation of calibration of a 9-element microphone array with synthetic diffuse noise. The distances are estimated using averaging method and localization is obtained by the s-stress algorithm. (b) Calibration of a 12-element microphone array on synthetic diffuse noise. First the central microphones are localized using s-stress and then either the s-stress or matrix completion is used to calibrate the full network.

alternative s-stress method. Figure 5.7(b) demonstrates the results. Similar to the previous test, $d_{10,11}$, $d_{10,12}$, $d_{1,10}$, $d_{7,10}$, $d_{8,10}$, $d_{12,11}$, $d_{5,11}$, $d_{6,11}$, $d_{7,11}$, $d_{3,12}$, $d_{4,12}$, $d_{5,12}$ are missing from the distance matrix. We observe that the matrix completion method combined with Cadzow algorithm yields the best results with error 54. The s-stress and the classic matrix completion methods have errors equal to 99 and 159 respectively.

5.5 Summary

In this chapter we showed an application of the theory introduced in Chapters 2 and 3 for a real scenario in microphone array calibration. We proposed modifications to the method in [88] for better estimation of the mutual distances between microphones in a diffuse noise field and also for better microphone localization with matrix completion. We saw that imposing further constraints on the matrix completion algorithm helps in finding complete matrices which are closer to an EDM. Using these modifications we saw major improvements in the accuracy of microphone calibration in diffuse noise fields compared to state-of-the-art methods.

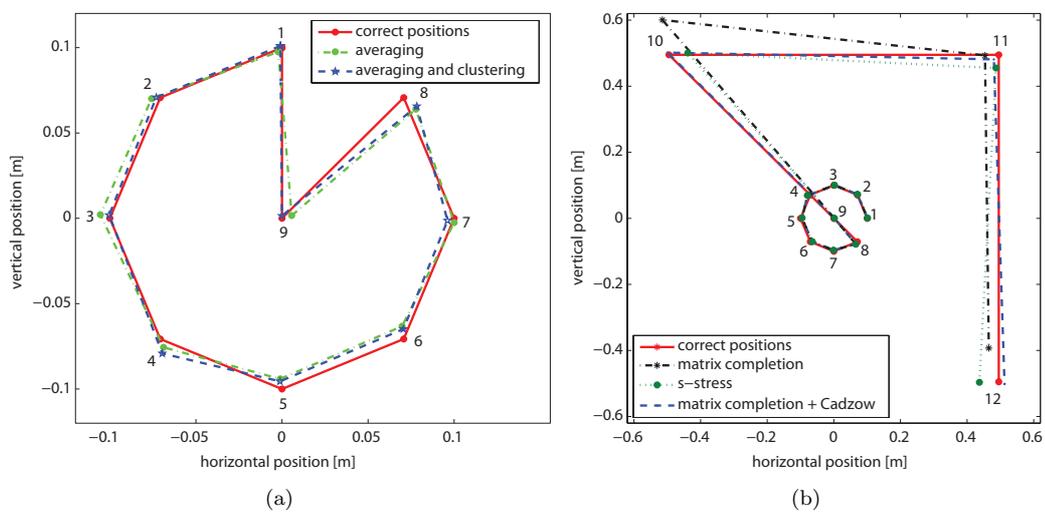


Figure 5.7: (a) Calibration of a 9-element microphone array on real diffuse noise recordings using averaging and a hybrid of averaging and histogram-based distance estimation. Localization is obtained by the *s*-stress method. (b) Calibration of a 12-element microphones array in real diffuse noise. The central array is calibrated using the *s*-stress method. Different approaches are used to calibrate the full network while several pairwise distances are missing.

Chapter 6

Sequences with Minimal Time-Frequency Spreads

The quest for absolute certainty is an immature, if not infantile, trait of thinking.

Herbert Feigl

6.1 Introduction

Recall from Chapter 4 the procedure for measuring the room impulse response. We used a sine sweep to calculate the impulse response. Also in Chapter 5 we used a short-time Fourier transform to frame the measured noise in each microphone. In these applications and many other, the signal of interest needs to be multiplied with a proper window with specific desired characteristics. Windowing in the time domain is equivalent to filtering in the frequency domain. Thus, we can mention the same argument for filtering. Suppose you are asked to design filters that are sharp in the frequency domain and at the same time compact in the time domain. The same problem is posed in designing sharp probing basis functions with compact frequency characteristics. In order to formulate these problems mathematically, we need to have a correct and universal definition of compactness and clarify what we mean by saying a signal is spread in time or frequency¹.

These notions are well defined and established for continuous-time signals [56, 131] and their properties are studied thoroughly in the literature. For a continuous-time signal, we can define the time and frequency characteristics as in Table 6.1. Note the connection of these definitions

1. The results presented in this chapter are part of a collaboration with Y. Barbotin and M. Vetterli [94]

domain	center	spread
time	$\mu_t = \frac{1}{\ x\ ^2} \int_{t \in \mathbb{R}} t x(t) ^2 dt$	$\Delta_t^2 = \frac{1}{\ x\ ^2} \int_{t \in \mathbb{R}} (t - \mu_t)^2 x(t) ^2 dt$
frequency	$\mu_{\omega_c} = \frac{1}{2\pi\ x\ ^2} \int_{\omega \in \mathbb{R}} \omega X(\omega) ^2 d\omega$	$\Delta_{\omega_c}^2 = \frac{1}{2\pi\ x\ ^2} \int_{\omega \in \mathbb{R}} (\omega - \mu_{\omega_c})^2 X(\omega) ^2 d\omega$

Table 6.1: Time and frequency centers and spreads for a continuous-time signal $x(t)$.

domain	center	spread
time	$\mu_n = \frac{1}{\ x\ ^2} \sum_{n \in \mathbb{Z}} n x_n ^2$	$\Delta_n^2 = \frac{1}{\ x\ ^2} \sum_{n \in \mathbb{Z}} (n - \mu_n)^2 x_n ^2$
frequency	$\mu_{\omega_\ell} = \frac{1}{2\pi\ x\ ^2} \int_{-\pi}^{\pi} \omega X(e^{j\omega}) ^2 d\omega$	$\Delta_{\omega_\ell}^2 = \frac{1}{2\pi\ x\ ^2} \int_{-\pi}^{\pi} (\omega - \mu_{\omega_\ell})^2 X(e^{j\omega}) ^2 d\omega$

Table 6.2: Time and frequency centers and spreads for a discrete-time signal x_n as extensions of Table 6.1 [131].

with the mean and variance of a probability distribution function $|x(t)|^2 / \|x\|^2$. The value of Δ_t^2 is considered as the spread of the signal in the time domain while $\Delta_{\omega_c}^2$ represents its spread in the frequency domain. We say that a signal is compact in time (or frequency) if it has a small time (or frequency) spread.

The Heisenberg uncertainty principle [56] states that continuous-time signals cannot be arbitrarily compact in both domains. Specifically, for any $x(t) \in L^2(\mathbb{R})$,

$$\eta_c = \Delta_t^2 \Delta_{\omega_c}^2 \geq \frac{1}{4},$$

where the lower bound is achieved for Gaussian signals of the form $x(t) = \gamma e^{-\alpha t^2}$, $\alpha > 0$ [131]. The subscript c stands for continuous-time definitions. We call η_c the *time-frequency spread* of x .

Although the continuous Heisenberg uncertainty principle is widely used in theory, in practice we often work with discrete-time signals (e.g. filters and wavelets). Thus, equivalent definitions for discrete-time sequences are needed in signal processing. In the next section we study two common definitions of center and spread available in the literature.

6.1.1 Uncertainty principles for sequences

An obvious and intuitive extension of the definitions in Table 6.1 for discrete-time signals is presented in Table 6.2, where

$$X(e^{j\omega}) = \sum_{n \in \mathbb{Z}} x_n e^{-j\omega n} \quad \omega \in \mathbb{R}, \quad (6.1)$$

is the discrete-time Fourier transform (DTFT) of x_n .

Using the definitions in Table 6.2, we can also state the Heisenberg uncertainty principle for discrete-time signals as [131]:

$$\eta_\ell = \Delta_n^2 \Delta_{\omega_\ell}^2 > \frac{1}{4}, \quad x_n \in \ell^2(\mathbb{Z}) \text{ with } X(e^{j\pi}) = 0, \quad (6.2)$$

where the subscript ℓ stands for *linear* in reference to the definition of the frequency spread. Note the extra assumption on the Fourier transform of the signal in (6.2). This assumption is necessary for the result to hold.

Example 6.1

Take $x_n = \delta_n + 7\delta_{n-1} + 2\delta_{n-2}$. It is easy to verify that $|X(e^{j\pi})| = 0.22 \neq 0$, which violates the condition $X(e^{j\pi}) = 0$. The linear time-frequency spread of this signal according to Table 6.2 is $\eta_\ell = 0.159 < 1/4$.

Let us give another example regarding the limit of the time-frequency spread of discrete sequences.

Example 6.2

Consider the second order binomial filter $H(z) = (1 + z^{-1})^2$. We are interested in studying the behavior of this filter in the limit of its length. Thus, we iterate the filter. The iteration is in the sense of filter-banks and is equivalent to upsampling the filter and then convolving it with itself. If a filter is iterated J times, its z -transform is

$$H^{(J)}(z) = H(z)H(z^2) \cdots H(z^{2^{J-1}}).$$

The filter $H(z)$ and some of its iterations are shown in Figure 6.1. If we use the definitions from Table 6.2, we can see empirically that

$$\lim_{J \rightarrow \infty} \eta_\ell(h_n^{(J)}) = \frac{3}{10}.$$

The surprise comes when we find out that for the continuous hat function

$$h(t) = \begin{cases} 1 - |t| & |t| < 1 \\ 0 & \text{otherwise,} \end{cases}$$

the continuous time-frequency spread is

$$\eta_c(h(t)) = \frac{3}{10}.$$

In this particular case, through several pages of computation², we can prove the limit. But the question is what happens in the limit for other general filters?

In addition to the restriction on the Heisenberg uncertainty principle, shown in Example 6.1, the definitions in Table 6.2 do not capture the periodic nature of $X(e^{j\omega})$ for the frequency center and spread. In the search for more natural properties, we can adopt definitions for circular moments widely used in quantum mechanics [18] and directional statistics [84].

For a sequence x_n , $n \in \mathbb{Z}$, with a 2π periodic DTFT, $X(e^{j\omega})$ as in (6.1), the *first trigonometric*

2. In the interest of time and space, we skip the details.

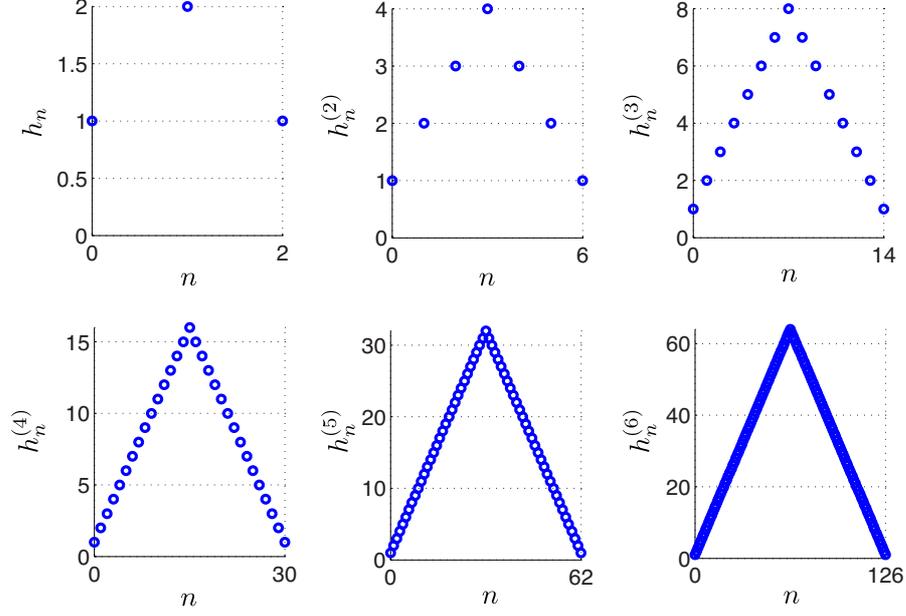


Figure 6.1: The second order binomial filter and its iterations used in Example 6.2.

domain	center	spread
time	$\mu_n = \frac{1}{\ x\ ^2} \sum_{n \in \mathbb{Z}} n x_n ^2$	$\Delta_n^2 = \frac{1}{\ x\ ^2} \sum_{n \in \mathbb{Z}} (n - \mu_n)^2 x_n ^2$
frequency	$\mu_{\omega_p} = 1 - \tau(x)$	$\Delta_{\omega_p}^2 = \frac{1 - \tau(x) ^2}{ \tau(x) ^2} = \left \frac{\ x\ ^2}{\sum_{n \in \mathbb{Z}} x_n x_{n+1}^*} \right ^2 - 1$

Table 6.3: Time and frequency centers and spreads for a discrete-time signal x_n using circular moments, where $\tau(x)$ is defined in (6.3).

moment is defined as [98, 99]

$$\tau(x) = \frac{1}{2\pi \|x\|^2} \int_{-\pi}^{\pi} e^{j\omega} |X(e^{j\omega})|^2 d\omega. \quad (6.3)$$

This moment was originally defined for probability distributions on a circle. However, we can also apply it to periodic functions with proper normalization.

Using (6.3), the *periodic frequency spread* is defined as [18]:

$$\Delta_{\omega_p}^2 = \frac{1 - |\tau(x)|^2}{|\tau(x)|^2} = \left| \frac{\|x\|^2}{\sum_{n \in \mathbb{Z}} x_n x_{n+1}^*} \right|^2 - 1.$$

The definition of Δ_n^2 remains unchanged as in Table 6.2. These are collected in Table 6.3.

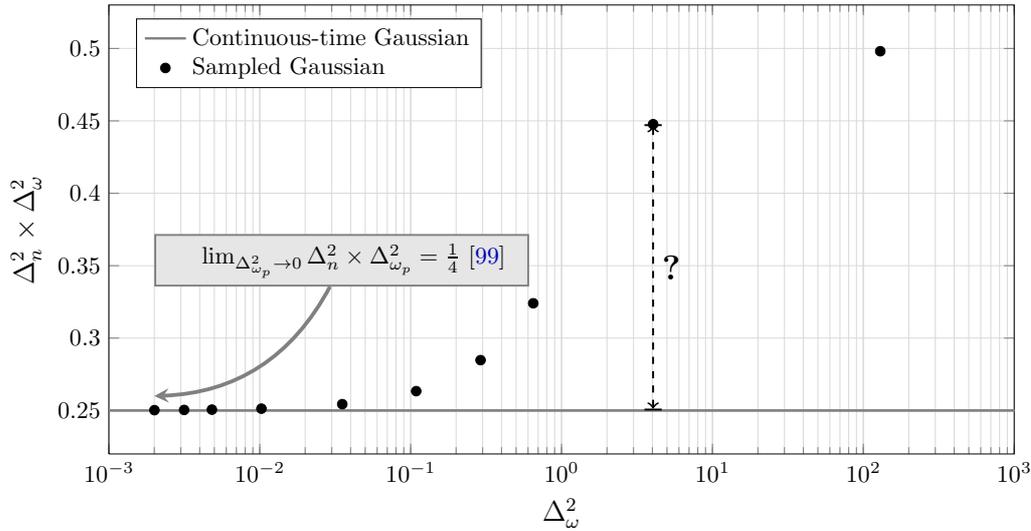


Figure 6.2: Time-frequency spread of continuous vs. sampled Gaussians. The solid line shows the $1/4$ Heisenberg bound which is achieved by continuous Gaussian signals (with definitions in Table 6.1), while the markers show the time-frequency spread (according to Table 6.3) of sampled Gaussian sequences. The question is if the gap between the two curves is inherited from the properties of sequences or sampled Gaussians are not optimal in discrete domain.

6.1.2 Contribution

In this chapter, using the definitions in Table 6.3, we revisit the Heisenberg uncertainty principle for discrete-time signals.

We address the fundamental yet unanswered question: If someone asks us to design a discrete-time filter with a certain frequency spread ($\Delta_{\omega_p}^2$ fixed), can we return the sequence with minimal time spread Δ_n^2 ? In other words, the problem is to find the solution to

$$\begin{aligned} \Delta_{n,\text{opt}}^2 &= \underset{x_n}{\text{minimize}} && \Delta_n^2 \\ &\text{subject to} && \Delta_{\omega_p}^2 = \text{fixed}. \end{aligned} \quad (6.4)$$

One could also ask the dual question: Given a fixed Δ_n^2 , minimize the value of $\Delta_{\omega_p}^2$. By the end of this chapter, we will see that both of these questions lead to the same answer. In order to provide an insight on the uncertainty principle in the discrete time domain, we do a simple test. In Figure 6.2, the solid line of ordinate $1/4$ shows the time-frequency spread of continuous Gaussian signals. This is the Heisenberg's uncertainty bound which is achievable by continuous-time Gaussians. Further, using the definitions in Table 6.3, we compute the time-frequency spread of sampled Gaussian sequences. Prestin et al. in [99] prove that the time-frequency spread tends to $1/4$ as the frequency spread of Gaussians decreases. But this experiment shows that when the frequency spread is large, the values are far from the uncertainty bound. Two questions arise here:

- Is the $1/4$ uncertainty bound also tight for discrete sequences?
- Are sampled Gaussians the minimizers of the uncertainty in the discrete domain?

Answers to these questions will be apparent if we can solve (6.4). We call the solution of (6.4) a *maximally compact* sequence.

By formulating the design of maximally compact sequences as an optimization problem, we show that in contrast to the continuous-time case, it is not possible to reach a constant time-frequency lower bound for discrete-time sequences. We further develop a simple optimization framework to find maximally compact sequences in the time domain for a given frequency spread. In other words, we provide in a constructive and numerical way, a *sharp uncertainty principle for sequences*, later shown in Figure 6.4. We also show that the Fourier spectra of maximally compact sequences are in fact a very special class of Mathieu functions. Using the asymptotical expansion of these functions, we develop closed-form bounds on the time-frequency spread of maximally compact sequences.

6.1.3 Related Work

The classic uncertainty principle [56] assumes continuous-time non-periodic signals. Several works in the signal processing community also address the discrete-time/discrete-frequency case [33, 85, 101]. Our work bridges these two cases by considering the discrete-time/continuous-frequency regime.

Note that not all studies about the uncertainty principle concern the notion of spread. For example, the authors in [101] propose the uncertainty bound on the information content of signals (entropy) and [33] provides a bound on the non-zero coefficients of discrete-time sequences and their discrete Fourier transforms.

The discrete-time/continuous-frequency scenario has been recently encountered in many practical applications in signal processing. Examples include uncertainty principle on graphs [2] and on spheres [70]. Studies on the periodic frequency spread can be found in [18] and [132]. A comprehensive work on the uncertainty relations for discrete sequences is found in [99]. The authors show that $1/4$ is a lower-bound on the time-frequency spread, which can only be achieved asymptotically as the sequence spreads in time. We provide sharp achievable bounds in the non-extreme case which match the results in [99] in the asymptotic regime.

This problem is similar to — although different than — the design of Slepian’s discrete prolate spheroidal sequences (DPSS’s). First introduced by Slepian in 1978 [118], DPSS’s are sequences designed to be both limited in the time and band-limited in the frequency domains. For a finite length N in time and a cut-off frequency W , the DPSS’s are a collection of N discrete-time sequences that are strictly band-limited to the digital frequency range $|f| < W$, yet highly concentrated in time to the index range $n = 0, 1, \dots, N - 1$. Such sequences can be found using an algorithm similar to the Papoulis-Gerchberg method. Note the difference of such sequences to those that we intend to design here; we do not impose any constraints on the bandwidth of the sequences in the frequency domain. Also, the original ideas presented in this chapter are applicable both to finite and infinite length sequences. Moreover, we use the notion of variance for the concentration of signals in time and frequency domains.

6.2 Maximally Compact Sequences

In the search for maximally compact sequences, we use the definitions of spreads presented in Table 6.3. Although the choice for the definition of periodic variance might not seem intuitive at first sight, we show in [94] that $\Delta_{\omega_p}^2$ corresponds to the finite difference operator — which

is the simplest among discrete differentials — applied on the momentum operator in time as a consequence of discretization in the temporal domain.

Using the definitions in Table 6.3, we can state the Heisenberg uncertainty principle for discrete sequences as [18]:

$$\eta_p = \Delta_n^2 \Delta_{\omega_p}^2 \geq \frac{1}{4}, \quad \text{for } \|x\|_0 > 1, \quad (6.5)$$

The condition $\|x\|_0 > 1$ in (6.5) is necessary to exclude signals with one sample, as for such signals we would have $\Delta_n^2 = 0$. Equation (6.5) provides a lower bound on the solution of (6.4). However, we saw in Figure 6.2 that this bound is not tight for Gaussians. Let us rewrite the mathematical formulation of the problem that we aim to solve:

$$\begin{aligned} \Delta_{n,\text{opt}}^2 &= \underset{x_n}{\text{minimize}} && \Delta_n^2 \\ &\text{subject to} && \Delta_{\omega_p}^2 = \sigma^2, \end{aligned} \quad (6.6)$$

where σ^2 is the fixed and given frequency spread of the sequence. Although finding a tight bound for maximally compact sequences is an important task, the more interesting problem is to constructively design sequences which achieve such a bound. The ideal answer to this question would be to find analytic closed form solutions — like Gaussian signals for the continuous-time problem — achieving the tight bound. In the following we will address these problems.

We start with some properties of maximally compact sequences. These properties will greatly facilitate the task of solving (6.6).

6.2.1 Properties of Maximally Compact Sequences

In the definitions of time and frequency spreads in Table 6.3 we considered complex sequences and their DTFTs. In the following, we establish two lemmas that make the search for maximally compact sequences easier.

Lemma 6.1

Maximally compact sequences are generalized linear-phase sequences derived from real-valued, positive maximally compact sequences; i.e. x_n is a maximally compact sequence only if

$$\exists \varphi, \psi \in [0, 2\pi[\quad \text{such that} \quad x_n = |x_n| e^{-j(\varphi n + \psi)},$$

where $|x|$ is a maximally compact sequence.

Proof.

See Appendix 6.A.1. □

Consider also the shift operator

$$x_{n+\nu} \xleftrightarrow{\text{DTFT}} e^{j\omega\nu} X(e^{j\omega}), \quad \nu \in \mathbb{R}, \quad (6.7)$$

whose principal effect is to shift the time centre of a sequence

$$\mu_n(x_{n-\nu}) = \mu_n(x) + \nu.$$

Notice that ν might be non-integer, in which case $x_{n-\nu}$ is a shorthand for sinc resampling on a grid shifted by ν in the time domain.

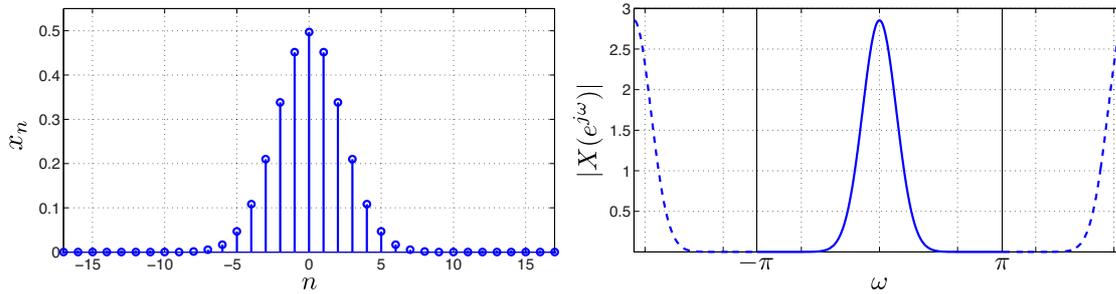


Figure 6.3: An example solution of (6.8). The output of the SDP in (6.8) with $\sigma^2 = 0.1$ using `cvx` in Example 6.3. The optimal value for Δ_n^2 is found to be 2.62 which results in a time-frequency spread of $\eta_p = 0.262$.

Example 6.3

Take $\sigma^2 = 0.1$ to be the fixed and given frequency spread of the sequence. We can use `cvx` to solve the semi-definite program (6.8) and find the optimal value of $\Delta_n^2 = 2.62$. This results in the time-frequency spread of $\eta_p = 0.262$. The simple code in MATLAB is:

```
cvx_begin
variable X(n,n);
minimize(trace(A*X))
subject to
    trace(B*X) == 1/sqrt(1+0.1)
    trace(X) == 1
    X == semi-definite(n)
cvx_end;
```

Note that in contrast to continuous-time signals, we cannot reach the 0.25 lower bound for sequences. The resulting sequence and its DTFT are shown in Figure 6.3.

The dual of SDP (6.8) is [16]:

$$\begin{aligned} & \underset{\lambda_1, \lambda_2}{\text{maximize}} && \alpha \lambda_1 + \lambda_2 \\ & \text{subject to} && \mathbf{A} - \lambda_1 \mathbf{B} - \lambda_2 \mathbf{I} \succeq 0 \end{aligned} \quad (6.10)$$

Lemma 6.3

For the primal problem (6.8) and the dual (6.10), strong duality holds.

Proof.

See Appendix 6.A.3. □

Thus, for finding the time-frequency spread of maximally compact sequences, solving the dual problem suffices.

Note that although Theorem 6.1 provides a constructive way for finding maximally compact sequences, it does not specify the closed form for these sequences. One would be interested to see

if — in analogy to continuous-time — sampled gaussians are maximally compact? The answer is negative, as shown by this theorem:

Theorem 6.2

The DTFT spectra, $X(e^{j\omega})$ of maximally compact sequences are Mathieu functions. More specifically,

$$X(e^{j\omega}) = \gamma_0 \cdot \text{ce}_0(-2\lambda_1; (\omega - \omega_0)/2) e^{j\mu\omega},$$

where $|\gamma_0| = \|\text{ce}_0(-2\lambda_1; (\omega - \omega_0)/2)\|^{-1}$, ω_0 and μ are shifts in frequency or time and λ_1 is the optimal solution of the dual problem (6.10). The function $\text{ce}_0(q; \omega)$ is Mathieu's harmonic cosine function of order zero.

Proof.

For the proof of the theorem and further insights about Mathieu functions, see Appendix 6.A.4. \square

Using the constructive method presented in Theorem 6.1, we can find the achievable (and tight) uncertainty principle bound for discrete sequences. This is shown and discussed more in Section 6.3 and Figure 6.4. However, a numerically computed bound may not always be satisfying, and even though the numerical solution exactly solves the problem, its accuracy may be challenged. Therefore, we characterize the asymptotic behavior of the time-frequency bound:

Theorem 6.3

If x_n is maximally compact for a given $\Delta_{\omega_p}^2 = \sigma^2$, then

$$\eta_p = \Delta_n^2 \Delta_{\omega_p}^2 \geq \sigma^2 \left(1 - \sqrt{\frac{\sigma^2}{1 + \sigma^2}} \right). \quad (6.11)$$

Proof.

The proof for this theorem is provided in Appendix 6.A.5. \square

This fundamental result states that for a given frequency spread, we cannot design sequences that achieve the classic Heisenberg uncertainty bound. We will see how this curve compares to the classic Heisenberg bound in Section 6.3.

The lower bound in (6.11) converges to 1/2 as the value of σ^2 grows, and “pushes up” the time-frequency spread of maximally compact sequences towards 1/2 which is also an asymptotic upper bound on the time-frequency spread as $\Delta_{\omega_p}^2 \rightarrow \infty$; indeed, one may construct the unit-norm sequence $x_n^{(\varepsilon)} = \varepsilon \delta_{n+1} + \sqrt{1 - 2\varepsilon^2} \delta_n + \varepsilon \delta_{n-1}$, which verifies $\lim_{\varepsilon \rightarrow 0} \eta_p(x^{(\varepsilon)}) = 1/2$.

Theorem 6.4

For small values of σ^2 , maximally compact sequences satisfy

$$\eta_p = \Delta_n^2 \Delta_{\omega_p}^2 \leq \frac{\sigma^2}{8} \left(\frac{\sqrt{1 + \sigma^2}}{\sqrt{1 + \sigma^2} - 1} - \frac{1}{2} \right). \quad (6.12)$$

Proof.

See Appendix 6.A.6. □

For small values of σ^2 , the upper bound in (6.12) converges from above to $1/4$, thus “pushing down” the time-frequency spread of maximally compact sequences towards the Heisenberg uncertainty bound $1/4$ from above.

Finite-Length Sequences

The theory that we have provided so far holds for infinite sequences. For computational purposes, we have to assume finite length for the sequences in the time domain, which is not an issue if the sequence length is chosen to be long enough. As a side benefit, a length constraint on the sequence may be put at will without changing the design algorithm.

6.3 Numerical Analysis

In order to show the behavior of the results obtained in Theorems 6.1, 6.3 and 6.4, we ran some simulations. To this end, we assumed that the designed filter is of finite length with 201 taps in the time domain (the length is long enough not to pose restrictions on the solution). For different values of $\Delta_{\omega_p}^2 = \sigma^2$, we solved the semi-definite program (6.8) using the cvx toolbox in MATLAB.

The resulting values of Δ_n^2 were then multiplied with the corresponding $\Delta_{\omega_p}^2$ to produce the time-frequency spread of maximally compact sequences. The time-frequency spread of maximally compact sequences versus their frequency spread is shown with the solid curve in Figure 6.4. This means (numerically) that any time-frequency spread under this curve is not achievable. The dotted line in this figure shows the classic Heisenberg uncertainty bound. Comparing the two curves shows the gap between the classic Heisenberg principle and what is achievable in practice. The dashed lines represent analytical lower and upper bounds for the time-frequency spread of maximally compact sequences (found in Theorems 6.3 and 6.4, respectively).

Further, to give an insight on how the time-frequency spread of some common filters compare to that of maximally compact sequences, we plot their time-frequency spread together with the new uncertainty bound in Figure 6.5. By changing the length of each filter in time, we can find its time and frequency spreads which results in a point on the figure. We observe that as shown by Prestin et al. in [99], asymptotically when the frequency spread of sequences are very small, sampled Gaussians converge to the lower bound for maximally compact sequences.

6.4 Summary

In this chapter we first redefined the periodic frequency spread for discrete-time sequences. We then proposed a constructive way for designing (exactly and not approximately) sequences with minimum time-frequency spreads. We called such sequences maximally compact. We saw that in contrast to continuous-time signals, the time-frequency spread of sequences cannot achieve the classic Heisenberg bound. We further provided a new tight bound for sequences. Theorem 6.2 show that sampled Gaussians are not maximally compact, although they are very close to being so!

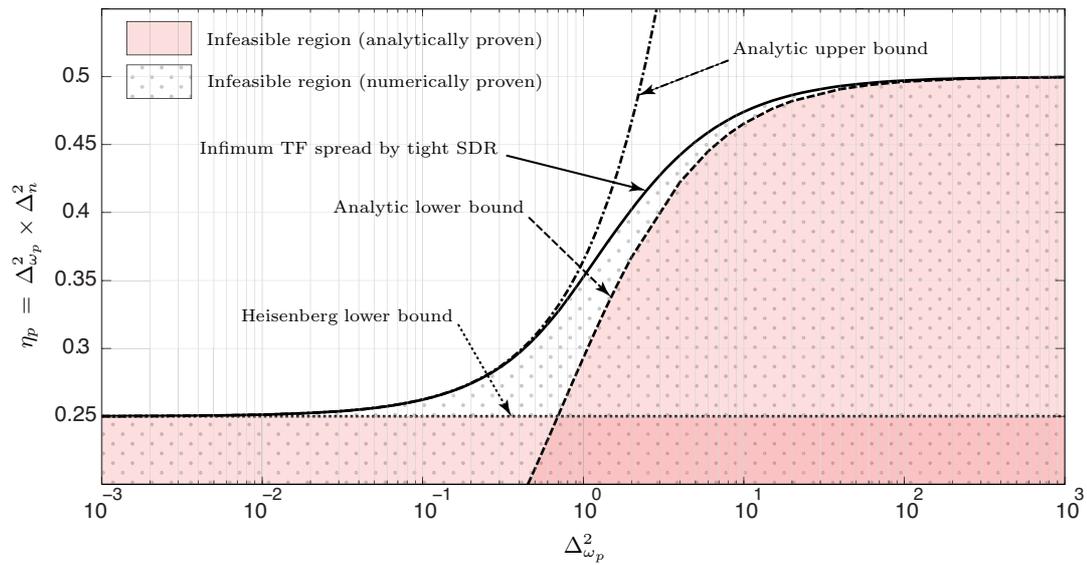


Figure 6.4: New uncertainty bounds. The solid line shows the results of solving the SDP in (6.8). The dotted line shows the classic Heisenberg uncertainty principle. The dashed lines show the analytic lower and upper bounds found in Theorems 6.3 and 6.4, respectively.

We can use the theory and bounds proved in this chapter as a benchmark for different filters and wavelet function to quantify how compact they are in time and frequency.

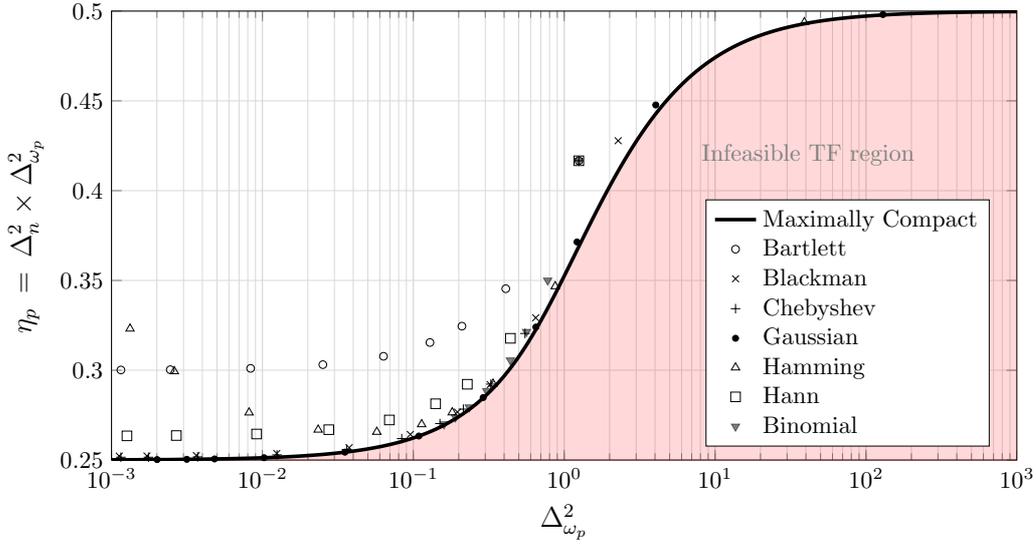


Figure 6.5: Time-frequency spread of common FIR filters. By changing the length of the filters in time, we compute the time and frequency spreads for each type of filters. For small values of frequency spread, Gaussian filters are good approximations of Mathieu functions (as shown also in [99]).

6.A Analysis

6.A.1 Proof of Lemmas 6.1 and 6.2

Proof of Lemma 6.1

Let x be maximally compact for a given time-spread $\Delta_n^2(x)$.

$$\Delta_n^2(x) = \Delta_n^2(|x|).$$

$$\begin{aligned} \Delta_{\omega_p}^2(|x|) &= \left| \sum_{n \in \mathbb{Z}} |x_n| |x_{n+1}| \right|^{-2} - 1, \\ &\leq \left| \sum_{n \in \mathbb{Z}} x_n x_{n+1}^* \right|^{-2} - 1 = \Delta_{\omega_p}^2(x). \end{aligned} \quad (6.13)$$

Since x is maximally compact, equality must be met in (6.13) to avoid contradiction, which amounts to

$$x_n x_{n+1}^* = |x_n| |x_{n+1}| e^{j\varphi}, \quad \varphi \in [0, 2\pi), \quad \forall n \in \mathbb{Z}.$$

This condition is equivalent to (6.1).

For maximally compact sequences, if Δ_n^2 strictly monotonically varies in function of $\Delta_{\omega_p}^2$, then fixing Δ_n^2 or $\Delta_{\omega_p}^2$ is equivalent and proves the lemma. In the following lemma we show that for maximally compact sequences Δ_n^2 changes monotonically with $\Delta_{\omega_p}^2$.

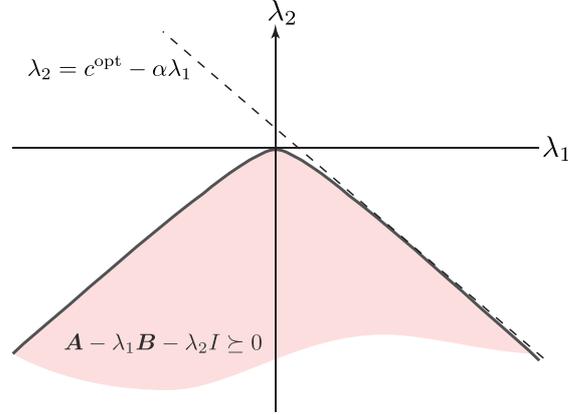


Figure 6.6: The feasible set of the dual problem (6.10) and the supporting line. As α increases, we need to elevate the line more to support the feasible set, which means that the optimal value of Δ_n^2 increases.

Lemma 6.4

For maximally compact sequences, Δ_n^2 is a decreasing function of $\Delta_{\omega_p}^2$.

Proof.

For proving this lemma, we use the dual formulation in (6.10). The feasible region of the dual problem is shown in Figure 6.6. We can write the dual as

$$\begin{aligned} & \underset{\lambda_1, \lambda_2}{\text{maximize}} && c \\ & \text{subject to} && \lambda_2 = c - \alpha \lambda_1, \\ & && \mathbf{A} - \lambda_1 \mathbf{B} - \lambda_2 \mathbf{I} \succeq 0. \end{aligned}$$

Note that α changes between 0 and 1 (see Figure 6.6). For a fixed α , the maximum c^{opt} is found by elevating the corresponding line $\lambda_2 = c - \alpha \lambda_1$ until it supports the feasible set (it is tangent to it). Since the feasible set is convex, as α grows (which means $\Delta_{\omega_p}^2$ decreases), we need a higher elevation of the line to support the convex set, thus c^{opt} (equivalently Δ_n^2) increases. \square

Proof of Lemma 6.2

Consider the shift operator in (6.7). Since the shift operation does not change the norm of a sequence, we will assume a unit norm sequence without loss of generality. We can show that

$$\begin{aligned} \mu_n(x_{n-\nu}) &= \frac{\langle j \frac{d}{d\omega} (e^{-j\omega\nu} X(e^{j\omega})), e^{-j\omega\nu} X(e^{j\omega}) \rangle}{2\pi} \\ &= \frac{\langle j \frac{d}{d\omega} X(e^{j\omega}), X(e^{j\omega}) \rangle}{2\pi} + \frac{\nu \langle X(e^{j\omega}), X(e^{j\omega}) \rangle}{2\pi} \\ &= \mu_n(x) + \nu, \end{aligned} \tag{6.14}$$

where we used the DTFT domain definition of the time center [99]:

$$\mu_n(x) = \frac{\langle j \frac{d}{d\omega} X(e^{j\omega}), X(e^{j\omega}) \rangle}{\|X\|^2}.$$

The proof for Lemma 6.2 — trivial for $\nu \in \mathbb{Z}$ — is not obvious for arbitrary shifts. Let x be maximally compact with time center $\mu_n(x)$, then according to (6.14), $x_{n-\nu}$ is centered at $\mu_n(x) + \nu$ and

$$\begin{aligned} 2\pi\Delta_n^2(x_{n-\nu}) &= 2\pi \left[\sum_{n \in \mathbb{Z}} n^2 (x_{n-\nu})^2 - |\mu_n(x_{n-\nu})|^2 \right] \\ &= \left\langle j \frac{d}{d\omega} e^{-j\omega\nu} X(e^{j\omega}), j \frac{d}{d\omega} e^{-j\omega\nu} X(e^{j\omega}) \right\rangle - 2\pi |\mu_n(x) + \nu|^2 \\ &= \langle -j\nu X(e^{j\omega}) + X'(e^{j\omega}), -j\nu X(e^{j\omega}) + X'(e^{j\omega}) \rangle - 2\pi |\mu_n(x) + \nu|^2 \\ &= 2\pi |\nu|^2 \langle X(e^{j\omega}), X(e^{j\omega}) \rangle + \langle X'(e^{j\omega}), X'(e^{j\omega}) \rangle \\ &\quad + \langle -j\nu X(e^{j\omega}), X'(e^{j\omega}) \rangle + \langle X'(e^{j\omega}), -j\nu X(e^{j\omega}) \rangle - 2\pi |\mu_n(x) + \nu|^2 \\ &= 2\pi |\nu|^2 + \langle X'(e^{j\omega}), X'(e^{j\omega}) \rangle + 2\pi \operatorname{Real}[\mu_n(x)\nu^*] - 2\pi |\mu_n(x) + \nu|^2 \\ &= \langle X'(e^{j\omega}), X'(e^{j\omega}) \rangle - 2\pi |\mu_n(x)|^2 \\ &= 2\pi\Delta_n^2(x). \end{aligned}$$

This shows that a time shift does not affect the time spread of a sequence. Thus, if x is a maximally compact sequence, then $x_{n-\mu_n(x)}$ is also maximally compact (note that a time shift does not change the frequency characteristics of the sequence). \square

6.A.2 Proof of Theorem 6.1

By using Lemmas 6.1 and 6.2, we can write problem (6.6) as

$$\begin{aligned} \Delta_{n,\text{opt}}^2 &= \underset{x_n}{\text{minimize}} && \sum_{n \in \mathbb{Z}} n^2 x_n^2 \\ &\text{subject to} && \sum_{n \in \mathbb{Z}} x_n x_{n+1} = \frac{1}{\sqrt{1 + \sigma^2}}, \\ &&& \sum_{n \in \mathbb{Z}} x_n^2 = 1. \end{aligned} \tag{6.15}$$

We can rewrite (6.15) in a matrix form as a quadratically constrained quadratic program (QCQP) [16]

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} && \mathbf{x}^T \mathbf{A} \mathbf{x} \\ &\text{subject to} && \mathbf{x}^T \mathbf{B} \mathbf{x} = \alpha, \\ &&& \mathbf{x}^T \mathbf{x} = 1, \end{aligned} \tag{6.16}$$

where \mathbf{A} and \mathbf{B} are defined in (6.9) and $\alpha = 1/\sqrt{1 + \sigma^2}$. This problem can be further reformu-

lated as follows:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \text{tr}(\mathbf{A}\mathbf{x}\mathbf{x}^T) \\ & \text{subject to} && \text{tr}(\mathbf{B}\mathbf{x}\mathbf{x}^T) = \alpha \\ & && \text{tr}(\mathbf{x}\mathbf{x}^T) = 1. \end{aligned}$$

Replacing $\mathbf{x}\mathbf{x}^T$ by \mathbf{X} , we can write equivalently

$$\begin{aligned} & \underset{\mathbf{X}}{\text{minimize}} && \text{tr}(\mathbf{A}\mathbf{X}) \\ & \text{subject to} && \text{tr}(\mathbf{B}\mathbf{X}) = \alpha \\ & && \text{tr}(\mathbf{X}) = 1 \\ & && \mathbf{X} \succeq 0, \text{rank}(\mathbf{X}) = 1. \end{aligned} \tag{6.17}$$

We further relax the above formulation to reach the semi-definite program

$$\begin{aligned} & \underset{\mathbf{X}}{\text{minimize}} && \text{tr}(\mathbf{A}\mathbf{X}) \\ & \text{subject to} && \text{tr}(\mathbf{B}\mathbf{X}) = \alpha \\ & && \text{tr}(\mathbf{X}) = 1, \mathbf{X} \succeq 0. \end{aligned} \tag{6.18}$$

In Lemma 6.5 we show that the semi-definite relaxation is tight. □

Lemma 6.5

The semi-definite relaxation (SDR) from (6.17) to (6.18) is tight.

Proof.

Shapiro and then Barnivok and Pataki [9, 81, 96, 116] show that if the SDP in (6.17) is feasible, then

$$\text{rank}(\mathbf{X}^{\text{opt}}) \leq \lfloor (\sqrt{8m+1} - 1)/2 \rfloor, \tag{6.19}$$

where m is the number of constraints of the SDP and \mathbf{X}^{opt} is its optimal solution. For our semi-definite program in (6.17), $m = 2$. Thus, (6.19) implies that the solution has rank 1. Using this fact, one can see that the semi-definite relaxation is in fact tight. Note that from the nature of the problem, (6.17) is clearly feasible; we can always find a periodic signal in the Fourier domain with a unit norm and a desired frequency spread, although not having an optimal time spread. □

6.A.3 Proof of Lemma 6.3

We use the following Lemma for the proof:

Lemma 6.6 ([125, 127])

For a semi-definite program and its dual: If the primal is feasible and the dual is strictly feasible, then strong duality holds.

It is easy to see the primal is feasible. One can always find a signal \mathbf{x} with a certain frequency spread and norm one. Using this signal we can construct $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ which shows the feasibility of the primal. For the dual, one can use the Gershgorin's circle theorem and show that a sufficient condition for $\mathbf{A} - \lambda_1 \mathbf{B} - \lambda_2 \mathbf{I} \succ 0$ to hold is $\lambda_2 < -\lambda_1$ and $\lambda_1 > 0$. Thus, the dual problem is strictly feasible. □

6.A.4 Proof of Theorem 6.2

If a sequence is a solution to the dual SDP problem (6.10), the dual constraint is active. Therefore, maximally compact sequences lie on the boundary of the quadratic cone

$$\mathbf{A} - \lambda_1 \mathbf{B} - \lambda_2 \mathbf{I} \succeq 0.$$

A maximally compact sequence \mathbf{x} is thus solution of the eigenvalue problem

$$(\mathbf{A} - \lambda_1 \mathbf{B})\mathbf{x} = \lambda_2 \mathbf{x}, \quad (6.20)$$

where λ_1 and λ_2 are the dual variables of the SDP problem, and λ_2 is also the minimal eigenvalue of $\mathbf{A} - \lambda_1 \mathbf{B}$ and x is the associated eigenvector (this can be also seen by forcing the derivative of the Lagrangian in (6.16) to zero).

This explicit link between the dual variables and the sequence, yields a differential equation for which the DTFT spectrum of maximally compact sequences is the solution. In the DTFT domain (6.20) becomes (expanding the matrix multiplications)

$$\begin{aligned} -X''(e^{j\omega}) - \lambda_1 \cos(\omega)X(e^{j\omega}) &= \lambda_2 X(e^{j\omega}), \\ \Leftrightarrow X''(e^{j\omega}) + (\lambda_2 + \lambda_1 \cos(\omega))X(e^{j\omega}) &= 0, \end{aligned} \quad (6.21)$$

which is *Mathieu's differential equation* ([1] 20.1.1):

$$\frac{\partial^2 y(\omega)}{\partial \omega^2} + (a - 2q \cos(2\omega)) \cdot y(\omega) = 0. \quad (6.22)$$

The solutions of Mathieu's equation are called Mathieu functions, and they assume an odd and even form

$$\begin{aligned} \text{Mathieu's Cosine (even)} & \quad ce(a, q; \omega), \\ \text{Mathieu's Sine (odd)} & \quad se(a, q; \omega). \end{aligned}$$

Taking into account the periodicity of (6.21), it appears not all pairs of parameters (a, q) will lead to a periodic solution. Mathieu functions can be restricted to be 2π periodic:

Definition 6.1

The solutions of Mathieu's harmonic differential equation — equation (6.22) with a solution y 2π -periodic — are defined as

$$\begin{aligned} \text{Mathieu's harmonic Cosine (even, periodic)} & \quad ce_m(q; \omega) = ce(a_m(q), q; \omega), \quad m \in \mathbb{N}. \\ \text{Mathieu's harmonic Sine (odd, periodic)} & \quad se_m(q; \omega) = ce(b_m(q), q; \omega), \quad m \in \mathbb{N}^+. \end{aligned}$$

It is immediately visible that the spectrum of maximally compact sequences may only have the form

$$\begin{aligned} X(e^{j\omega}) &= \gamma_0 \cdot ce_m(-2\lambda_1; \omega/2) + \gamma_1 \cdot se_m(-2\lambda_1; \omega/2), \quad \text{for } m \in \mathbb{N}^+, \\ X(e^{j\omega}) &= \gamma_0 \cdot ce_m(-2\lambda_1; \omega/2), \quad \text{for } m = 0, \end{aligned} \quad (6.23)$$

for any constants γ_0 and γ_1 such that $\|X(e^{j\omega})\| = 1$. More specifically, for any $\lambda_1 \geq 0$, the dual SDP problem can be posed and any solution would have the form (6.23).

Characteristic numbers of Mathieu's equation are ordered [37], such that

$$\begin{aligned} a_0(-2\lambda_1) < a_1(-2\lambda_1) < b_1(-2\lambda_1) < b_2(-2\lambda_1) < a_2(-2\lambda_1) < \cdots, & \lambda_1 > 0, \\ a_0(-2\lambda_1) < b_1(-2\lambda_1) \leq a_1(-2\lambda_1) < b_2(-2\lambda_1) \leq a_2(-2\lambda_1) < \cdots, & \lambda_1 \leq 0, \end{aligned}$$

By (6.22) and with the substitution $\omega \rightarrow \omega/2$ one obtains $a_m(-2\lambda_1) = 4\lambda_2$. Because λ_2 is the minimal eigenvalue, we conclude that $m = 0$.

Note that this result validates the one in [120] which stated that asymptotically Mathieu functions minimize the time-frequency product. Here, with a different approach, we showed that only Mathieu's harmonic cosine of order 0 minimizes this product for any given frequency-spread. \square

6.A.5 Proof of Theorem 6.3

In order to prove the theorem we first provide the following lemma.

Lemma 6.7

$$\begin{array}{|l} \text{If} \\ \hline \lambda_2 \leq 1 - \sqrt{1 + \lambda_1^2}, \\ \hline \text{then } \mathbf{P} = \mathbf{A} - \lambda_1 \mathbf{B} - \lambda_2 \mathbf{I} \succ 0. \end{array} \quad (6.24)$$

Proof.

Note that if $\lambda_1 = 0$, then the matrix \mathbf{P} is positive semi-definite with the given condition. So in the proof we will assume that $\lambda_1 \neq 0$.

Consider the following matrices \mathbf{P}_1 and \mathbf{P}_2 :

$$\mathbf{P}_1 = \begin{bmatrix} 0 - \lambda_2 & -\lambda_1 & 0 & 0 & 0 \\ -\lambda_1/2 & 1 - \lambda_2 & -\lambda_1/2 & 0 & 0 \\ 0 & -\lambda_1/2 & 4 - \lambda_2 & -\lambda_1/2 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} 1 - \lambda_2 & -\lambda_1/2 & 0 & 0 \\ -\lambda_1/2 & 4 - \lambda_2 & -\lambda_1/2 & 0 \\ 0 & \ddots & \ddots & \ddots \end{bmatrix}.$$

Call \mathcal{I} the set of eigenvalues of \mathbf{P} , \mathcal{I}_1 the set of eigenvalues of \mathbf{P}_1 and \mathcal{I}_2 the set of eigenvalues of \mathbf{P}_2 . It is trivial to see that $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$.

We show that if condition (6.24) is satisfied, then both \mathbf{P}_1 and \mathbf{P}_2 have positive eigenvalues.

1. \mathbf{P}_1

One can decompose \mathbf{P}_1 as

$$\mathbf{P}_1 = \begin{bmatrix} 2 & & & & 0 \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ 0 & & & & 1 \end{bmatrix} \times \begin{bmatrix} -\lambda_2/2 & -\lambda_1/2 & 0 & 0 & 0 \\ -\lambda_1/2 & 1 - \lambda_2 & -\lambda_1/2 & 0 & 0 \\ 0 & -\lambda_1/2 & 4 - \lambda_2 & -\lambda_1/2 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots \end{bmatrix} = \mathbf{D} \times \mathbf{P}_1^{(s)}.$$

Note that both \mathbf{D} and $\mathbf{P}_1^{(s)}$ are symmetric. Now define

$$\begin{aligned}\mathbf{S} &\stackrel{\text{def}}{=} \mathbf{D}^{-1/2} \mathbf{P}_1 \mathbf{D}^{1/2} \\ &= \mathbf{D}^{-1/2} \left(\mathbf{D} \mathbf{P}_1^{(s)} \right) \mathbf{D}^{1/2} \\ &= \mathbf{D}^{1/2} \mathbf{P}_1^{(s)} \mathbf{D}^{1/2}.\end{aligned}$$

First observe that \mathbf{S} is symmetric, thus it has real eigenvalues.

Let λ be an eigenvalue of \mathbf{S} corresponding to eigenvector \mathbf{v} . Then

$$\begin{aligned}\mathbf{S} \mathbf{v} &= \lambda \mathbf{v} \\ \Rightarrow \mathbf{D}^{-1/2} \mathbf{P}_1 \mathbf{D}^{1/2} \mathbf{v} &= \lambda \mathbf{v} \\ \Rightarrow \mathbf{P}_1 \mathbf{D}^{1/2} \mathbf{v} &= \mathbf{D}^{1/2} \lambda \mathbf{v} \\ \Rightarrow \mathbf{P}_1 \left(\mathbf{D}^{1/2} \mathbf{v} \right) &= \lambda \left(\mathbf{D}^{1/2} \mathbf{v} \right).\end{aligned}$$

Thus, eigenvalues of \mathbf{S} and \mathbf{P}_1 are equal. Further, if \mathbf{v} is an eigenvector of \mathbf{S} , then $\mathbf{D}^{1/2} \mathbf{v}$ is an eigenvector of \mathbf{P}_1 . Therefore, it suffices to consider eigenvalues of \mathbf{S} ; If \mathbf{S} is positive-definite then all the eigenvalues of \mathbf{P}_1 are positive. We have

$$\mathbf{S} = \begin{bmatrix} -\lambda_2 & -\lambda_1/\sqrt{2} & 0 & 0 & 0 \\ -\lambda_1/\sqrt{2} & 1 - \lambda_2 & -\lambda_1/2 & 0 & 0 \\ 0 & -\lambda_1/2 & 4 - \lambda_2 & -\lambda_1/2 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots \end{bmatrix}.$$

Sylvester's criterion states that a symmetric matrix is positive definite if and only if its principal minors are all positive. We can use Gaussian elimination on the matrix \mathbf{S} to compute its principal minors.

$$\mathbf{S}^U = \begin{bmatrix} 0 - \lambda_2 & -\lambda_1/\sqrt{2} & 0 & 0 & 0 \\ 0 & s_1 & -\lambda_1/2 & 0 & 0 \\ 0 & 0 & s_2 & -\lambda_1/2 & 0 \\ 0 & 0 & 0 & \ddots & \ddots \end{bmatrix},$$

where $s_1 = 1 - \lambda_2 + \lambda_1^2/2\lambda_2$ and $s_2 = 4 - \lambda_2 - \lambda_1^2/4s_1$. In general one can write

$$s_{i+1} = (i+1)^2 - \lambda_2 - \frac{\lambda_1^2}{4s_i}, \quad i \geq 1.$$

Note that

$$s_{i+2} - s_{i+1} = 2i + 3 + \lambda_1^2 \frac{s_{i+1} - s_i}{4s_i s_{i+1}}.$$

If $s_{i+1} > s_i$ then $s_{i+2} > s_{i+1}$. Thus if we have $s_0 > 0$ and $s_2 > s_1$, then by induction, $s_i > 0$ and thus \mathbf{S} is positive-definite. We show that under condition (6.24), this is true. One can easily check that under this condition:

- (i) $s_0 > 0$.
- (ii) $s_1 = \frac{\sqrt{1+\lambda_1^2}-1}{2} > 0$.

(iii) $s_2 - s_1 = 3 > 0$.

This shows that under condition (6.24), \mathbf{S} is positive definite and thus \mathbf{P}_1 has only positive eigenvalues.

2. \mathbf{P}_2 :

As \mathbf{P}_2 is already symmetric, we can immediately use Sylvester's criterion on it. One can write

$$\mathbf{P}_2^U = \begin{bmatrix} 1 - \lambda_2 & -\lambda_1/2 & 0 & 0 & 0 \\ 0 & s_2 & -\lambda_1/2 & 0 & 0 \\ 0 & 0 & s_3 & -\lambda_1/2 & 0 \\ 0 & 0 & 0 & \ddots & \ddots \end{bmatrix},$$

where

$$s_{i+1} = (i+1)^2 - \lambda_2 - \frac{\lambda_1^2}{4s_i}, \quad i \geq 1.$$

It is easy to see that under condition (6.24), $s_i > |\lambda_1|/2 > 0$:

(i)

$$s_1 = 1 - \lambda_2 = 1 - (1 - \sqrt{1 + \lambda_1^2}) = \sqrt{1 + \lambda_1^2} > \frac{|\lambda_1|}{2}.$$

(ii)

$$s_{i+1} = (i+1)^2 - \lambda_2 - \frac{\lambda_1^2}{4s_i} = i^2 + 2i + \sqrt{1 + \lambda_1^2} - \frac{\lambda_1^2}{2s_i}.$$

Thus, by induction if $s_i > |\lambda_1|/2$, then

$$\begin{aligned} s_{i+1} &> i^2 + 2i + |\lambda_1| - \frac{\lambda_1^2}{4|\lambda_1|/2} \\ &= i^2 + 2i + |\lambda_1| - \frac{|\lambda_1|}{2} \\ &> \frac{|\lambda_1|}{2} \\ &> 0. \end{aligned}$$

Thus, \mathbf{P}_2 is also positive-definite. Putting these together, one can conclude that under condition (6.24), the matrix \mathbf{P} is positive-definite. This finishes the proof of Lemma 6.7. \square

Note that condition (6.24) gives a sufficient (but not necessary) condition on the feasible set of the dual problem (6.10). Thus, it provides a lower bound for the maximum value of the dual. Consider the restricted dual problem

$$\begin{aligned} &\underset{\lambda_1, \lambda_2}{\text{maximize}} && \alpha \lambda_1 + \lambda_2 \\ &\text{subject to} && \lambda_2 \leq 1 - \sqrt{1 + \lambda_1^2}. \end{aligned}$$

The solution to this problem is simply $1 - \sqrt{1 - \alpha^2}$. If we rewrite α in terms of σ^2 , we have at the end

$$\Delta_{n,\text{opt}}^2 \geq 1 - \sqrt{\frac{\sigma^2}{1 + \sigma^2}}.$$

This concludes the proof for the lower bound in Theorem 6.3. \square

6.A.6 Proof of Theorem 6.4

It is easy to see that for small values of σ^2 (equivalently α close to 1), the maximum of the dual problem is achieved for large values of λ_1 (remember that λ_1 needs to be positive). We saw in Appendix 6.A.4 that for maximally compact sequences (i.e. sequences that result in the maximum of the dual problem) we have $\lambda_2 = 1/4 a_0(2\lambda_1)$ (note that $a_0(-q) = a_0(q)$). McLachlan in [82] shows that for large enough values of q , we have

$$\begin{aligned} a_0(q) &= -2q + 2q^{\frac{1}{2}} + \frac{1}{4} - \frac{1}{32}q^{-\frac{1}{2}} - \frac{48}{27}q^{-1} - \frac{848}{2^{17}}q^{-\frac{3}{2}} - \frac{4,752}{2^{20}}q^{-2} - \frac{126,752}{2^{20}}q^{-\frac{5}{2}} - \dots \\ &= -2q + 2q^{\frac{1}{2}} + \frac{1}{4} - \frac{1}{32}q^{-\frac{1}{2}} + O(q^{-1}). \end{aligned}$$

Thus, we have for large q ,

$$a_0(q) \leq -2q + 2q^{\frac{1}{2}} + \frac{1}{4}.$$

After replacing q by $2\lambda_1$ and $a_0(2\lambda)$ by $4\lambda_2$, we have

$$\lambda_2 = \frac{1}{4}a_0(2\lambda_1) \leq -\lambda_1 + \frac{1}{\sqrt{2}}\sqrt{\lambda_1} - \frac{1}{16}.$$

Because this set contains the original feasible set of the dual problem, it will give an upper bound on the optimal value of the dual. In other words, $\Delta_n^2 \leq \sigma_r^2$, where

$$\begin{aligned} \sigma_r^2 &= \underset{\lambda_1, \lambda_2}{\text{maximize}} && \alpha \lambda_1 + \lambda_2 \\ &\text{subject to} && \lambda_2 \leq -\lambda_1 + \frac{1}{\sqrt{2}}\sqrt{\lambda_1} - \frac{1}{16}. \end{aligned} \tag{6.25}$$

It is easy to see that the maximum of (6.25) is achieved for $\lambda_1 = \frac{1}{8(1-\alpha)^2}$. Replacing λ_1 in (6.25) and using the fact that $\Delta_n^2 \leq \sigma_r^2$, leads to

$$\Delta_n^2 \leq \frac{1}{8} \left(\frac{\sqrt{1+\sigma^2}}{\sqrt{1+\sigma^2}-1} - \frac{1}{2} \right).$$

□

Chapter 7

Conclusion

A large part of this thesis focused on the Euclidean distanced matrices. We started by a brief review of EDMs' key properties. In particular, we emphasized the importance of their low rank nature. Using EDM properties, we proposed three algorithms for attacking different related problems in signal processing. Let us review these algorithms again:

Algorithm I) EDM Completion: Thanks to the low rank of EDMs, we used a state-of-the-art matrix completion method, OPTSPACE, to estimate missing distance entries in an EDM. Although completion of distance matrices has already been addressed in the literature, to the best of our knowledge, all of these treatments lacked the analysis of the completion and error bounds. We provided analytic error bounds on the completion of distance matrices in the presence of noise, random missing entries and also structured missing entries. Later, we proposed a modification to this algorithm to incorporate more properties of EDMs in the matrix completion framework (e.g. symmetry, zero diagonal and positivity). We introduced a Cadzow-like algorithm to alternately satisfy each of these properties.

Algorithm II) Echo Labeling: Again based on the rank property of EDMs, we proposed an algorithm for echo labeling. Knowing the distance matrix coming from five points, the algorithm chooses collection of echoes and augments this known distance matrix with the chosen collection. If the echoes come from a single physical point in space, the rank of the EDM remains the same, otherwise it increases.

Algorithm III) Distributed Sensor Localization: By utilizing a non-linear cost function called s-stress, we introduced an algorithm to find a set of points in space that generate a given noisy and incomplete distance matrix. We decomposed the s-stress function into disjoint pieces that could be minimized using an alternate coordinate descent approach. This algorithm can be used both for finding the configuration that produces a given distance matrix, or for finding the closest EDM (in the sense of Frobenius norm) to a given matrix. Through a randomized set of experiments, we saw that in more than 99% of cases, the proposed algorithm converges to the global minimum of the s-stress function. Further, the algorithm is easily adaptable to use variable weights for different pairs in the cost function.

Using the algorithms mentioned above, we addressed four applications in signal processing:

Application I) Calibration in Ultrasound Tomography: In ultrasound tomography devices normally several hundreds of sensors are embedded on a structure (e.g., a ring) that

send and receive ultrasound signals to estimate the sound speed distribution. We used *Algorithm I* to complete and denoise the distance matrix coming from these devices and then estimate the sensor positions. The analysis derived for this algorithm enabled us to provide analytic error bounds on the calibration.

Application II) Hearing The Shape of a Room: Using the image source model, we transformed the problem of estimating the geometry of a room to the problem of finding the location of image sources of the loudspeaker. We used *Algorithm II* to label the echoes recorded by each microphone and were able to reconstruct the shape of a room with five microphones and one loudspeaker. In the noisy case, with the help of *Algorithm III*, we found how much the augmented matrix is close to an EDM. Thus, a fitness score was assigned to each echo combination and the echo collections with the smallest scores were chosen. The same algorithms were applied to find the position of a microphone inside a known room with only one loudspeaker.

Application III) Microphone Array Calibration In Diffuse Noise Fields: In this application, using the inherent properties of diffuse noise fields, we estimated the distances between close-by microphones, and larger distances were estimated by *Algorithm I*. We used *Algorithm III* to find the locations of the central microphones in our test setup. These microphones then served as anchor points in finding the location of the rest of microphones.

Application IV) Optimal English Keyboard Design: In this example we took every pair of alphabet characters in English and computed their pair-wise distance in English words. We used the frequency of each pair as a weight in *Algorithm III* and found the best embedding of English keys in 2D. If one would be interested in designing 3D keyboards, it could be easily done by a simple change of one parameter in the estimation procedure!

We further studied the problem of designing maximally compact sequences. We revisited the definitions of variance for periodic functions. Using these definitions, we formulated the search for maximally compact sequences as a semi-definite program. Solving this program gave us a tight bound on the time-frequency spread of discrete-time sequences. This analysis further showed that the classic Heisenberg uncertainty bound is not tight for sequences. We showed that contrary to continuous-time signals, Gaussians are not the minimizers of time-frequency spread in discrete domain. The uncertainty lower bound is achieved by sequences that have as their Fourier transform, Mathieu's harmonic cosine function of order zero.

Bibliography

- [1] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. National Bureau of Standards Applied Mathematics Series 55. Tenth Printing.* ERIC, 1972.
- [2] A. Agaskar and Y. Lu, “Uncertainty principles for signals defined on graphs: Bounds and characterizations,” in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3493–3496, 2012.
- [3] 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, Jan. 1999.
- [4] B. Alipanahi Ramandi, “New approaches to protein NMR automation,” Ph.D. dissertation, University of Waterloo, 2011.
- [5] J. B. Allen and D. A. Berkley, “Image method for efficiently simulating small-room acoustics,” *Journal of the Acoustical Society of America*, vol. 60(s1), 1979.
- [6] —, “Image Method For Efficiently Simulating Small-room Acoustics,” *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.
- [7] F. Antonacci, J. Filos, M. R. P. Thomas, E. A. P. Habets, A. Sarti, P. 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.
- [8] F. Antonacci, D. Lonoce, M. Motta, A. Sarti, and S. Tubaro, “Efficient source localization and tracking in reverberant environments using microphone arrays,” in Proc. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.
- [9] A. Barvinok, “Problems of distance geometry and convex properties of quadratic maps,” *Discrete & Computational Geometry*, vol. 13, no. 1, pp. 189–202, 1995.
- [10] A. Beck, P. Stoica, and J. Li, “Exact and approximate solutions of source localization problems,” *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1770–1778, 2008.
- [11] S. 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.

-
- [12] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in Proc. *Third International Symposium on Information Processing in Sensor Networks*, pp. 46–54, 2004.
- [13] P. Biswas, T. chen Liang, K. chuan Toh, T. chung Wang, and Y. Ye, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 4, pp. 360–371, 2006.
- [14] I. Borg and P. J. Groenen, *Modern multi-dimensional scaling theory and application*. Chapman-Hall, 2001.
- [15] J. Borish, "Extension of the image model to arbitrary polyhedra," *Journal of the Acoustical Society of America*, vol. 75, no. 6, pp. 1827–1836, 1984.
- [16] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.
- [17] W. Braun, C. Bösch, L. R. Brown, N. G, 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.
- [18] E. Breitenberger, "Uncertainty measures and uncertainty relations for angle observables," *Foundations of Physics*, vol. 15, no. 3, pp. 353–364, 1985.
- [19] J. Cadzow, "Signal enhancement—a composite property mapping algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, 1988.
- [20] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.
- [21] S. Čapkun, M. Hamdi, and J. Hubaux, "GPS-free positioning in mobile ad hoc networks," *Cluster Computing*, vol. 5, no. 2, pp. 157–167, 2002.
- [22] S. J. Chapman, "Drums that sound the same," *The American Mathematical Monthly*, vol. 102, no. 2, pp. 124–138, Feb. 1995.
- [23] J. Chen, Y. Huang, and J. Benesty, "Time delay estimation," in *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, Y. Huang and J. Benesty, eds. Springer US, 2004, ch. 8, pp. 197–227.
- [24] M. Chen, Z. Liu, L. He, P. Chou, and Z. Zhang, "Energy-based position estimation of microphones and speakers for ad-hoc microphone arrays," in Proc. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2007.
- [25] R. K. Cook, R. V. Waterhouse, R. D. Berendt, S. Edelman, and M. C. Thompson, "Measurement of correlations coefficients in reverberant sound fields," *Journal of the Acoustical Society of America*, vol. 27, 1955.
- [26] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*. Meboo, 2005.

-
- [27] J. De Leeuw, “Applications of convex analysis to multidimensional scaling,” in *Recent Developments in Statistics*, J. Barra, F. Brodeau, G. Romier, and B. V. Cutsem, eds. North Holland Publishing Company, 1977, pp. 133–146.
- [28] —, “Convergence of the majorization method for multidimensional scaling,” *Journal of Classification*, vol. 5, no. 2, pp. 163–180, 1988.
- [29] L. Doherty, K. pister, and L. El Ghaoui, “Convex position estimation in wireless sensor networks,” in Proc. *IEEE INFOCOM*, vol. 3, pp. 1655–1663, 2001.
- [30] I. Dokmanic, Y. M. Lu, and M. Vetterli, “Can one hear the shape of a room: The 2-D polygonal case,” in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 321–324, 2011.
- [31] I. Dokmanic, 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.
- [32] I. Dokmanic and M. Vetterli, “Room helps: Acoustic localization with finite elements,” in Proc. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2617–2620, 2012.
- [33] D. L. Donoho and P. B. Stark, “Uncertainty principles and signal recovery,” *SIAM Journal on Applied Mathematics*, vol. 49, no. 3, pp. 906–931, 1989.
- [34] P. Drineas, A. Javed, M. Magdon-Ismail, G. Pandurangant, R. Virrankoski, and A. Savvides, “Distance matrix reconstruction from incomplete distance information for sensor network localization,” in Proc. *Sensor and Ad Hoc Communications and Networks*, vol. 2, pp. 536–544, 2006.
- [35] 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.
- [36] G. Ekman, “Dimensions of color vision,” *The Journal of Psychology*, vol. 38, no. 2, pp. 467–474, 1954.
- [37] A. Erdélyi, W. Magnus, F. Oberhettinger, F. Tricomi, and H. Bateman, *Higher transcendental functions*. McGraw-Hill New York, 1955, vol. 3.
- [38] A. Farina, “Simultaneous measurement of impulse response and distortion with a swept-sine technique,” in Proc. *Audio Engineering Society Convention 108*, pp. 1–24, 2000.
- [39] B. P. Flanagan and K. L. Bell, “Array self-calibration with large sensor position errors,” *Signal Processing*, vol. 81, 2001.
- [40] M. Fréchet, “Sur quelques points du calcul fonctionnel,” *Rendiconti del Circolo Matematico di Palermo*, vol. 22, no. 1, pp. 1–72, 1906.
- [41] N. Gaffke and R. Mathar, “A cyclic projection algorithm via duality,” *Metrika*, vol. 36, no. 1, pp. 29–54, 1989.

-
- [42] W. Glunt, T. L. Hayden, S. Hong, and J. Wells, “An alternating projection algorithm for computing the nearest Euclidean distance matrix,” *SIAM Journal on Matrix Analysis and Applications*, vol. 11, no. 4, pp. 589–600, 1990.
- [43] W. Glunt, T. L. Hayden, and W.-M. Liu, “The embedding problem for predistance matrices,” *Bulletin of Mathematical Biology*, vol. 53, no. 5, pp. 769–796, 1991.
- [44] C. Gordon and D. Webb, “You can’t hear the shape of a drum,” *American Scientist*, vol. 84, pp. 46–55, 1996.
- [45] C. Gordon, D. Webb, and S. Wolpert, “Isospectral plane domains and surfaces via Riemannian orbifolds,” *Inventiones mathematicae*, vol. 110, no. 1, pp. 1–22, Dec. 1992.
- [46] —, “One cannot hear the shape of a drum,” *Bulletin of the American Mathematical Society*, vol. 27, no. 1, pp. 134–138, 1992.
- [47] J. C. Gower, “Euclidean distance geometry,” *The Mathematical Scientist*, vol. 7, pp. 1–14, 1982.
- [48] —, “Properties of Euclidean and non-Euclidean distance matrices,” *Linear Algebra and its Applications*, vol. 67, pp. 81–97, 1985.
- [49] M. Grant and S. Boyd, “cvx: MATLAB software for disciplined convex programming, version 1.21,” 2010.
- [50] 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.
- [51] E. Habets, “Generating sensor signals in isotropic noise fields,” *Journal of the Acoustical Society of America*, vol. 122 (6), 2007.
- [52] T. F. Havel, “Distance geometry: Theory, algorithms and chemical applications,” in *Proc. Encyclopedia of Computational Chemistry*, pp. 723–742. John Wiley & Sons, 1998.
- [53] 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,” *Journal of Molecular Biology*, vol. 182, no. 2, pp. 281–294, 1985.
- [54] T. L. Hayden, J. Wells, W.-M. Liu, and P. Tarazaga, “The cone of distance matrices,” *Linear Algebra and its Applications*, vol. 144, no. 0, pp. 153–169, 1990.
- [55] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in *Proc. MobiCom '03*, pp. 81–95, 2003.
- [56] W. Heisenberg, “The actual content of quantum theoretical kinematics and mechanics,” *Physikalische Z.*, vol. 43, p. 172, 1927, (Translated).
- [57] 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*, vol. 6, no. 2, pp. 1–32, 2010.

-
- [58] A. K. Jain and D. Zongker, "Representation and recognition of handwritten digits using deformable templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1386–1391, 1997.
- [59] V. Jain and L. Saul, "Exploratory analysis and visualization of speech and music by locally linear embedding," in Proc. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, 2004.
- [60] A. Javanmard and A. Montanari, "Localization from incomplete noisy distance measurements," *Foundations of Computational Mathematics*, vol. 13, no. 3, pp. 297–345, 2013.
- [61] S. Jennings, *Artist's Color Manual: The Complete Guide to Working with Color*. Chronicle Books, 2003.
- [62] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM*, vol. 24, pp. 1–13, January 1977.
- [63] I. Jovanović, "Inverse problems in acoustic tomography," Ph.D. dissertation, EPFL, Lausanne, 2008.
- [64] I. Jovanović, A. Hormati, L. Sbaiz, and M. Vetterli, "Efficient and stable acoustic tomography using sparse reconstruction methods," in Proc. *19th International Congress on Acoustics*, 2007.
- [65] I. Jovanović, L. Sbaiz, and M. Vetterli, "Acoustic tomography for scalar and vector fields: theory and application to temperature and wind estimation," *Journal of Atmospheric and Oceanic Technology*, vol. 26, no. 8, pp. 1475–1492, 2009.
- [66] M. Kac, "Can one hear the shape of a drum," *The American Mathematical Monthly*, vol. 73, pp. 1–23, 1966.
- [67] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," in Proc. *Advances in Neural Information Processing Systems*, December 2009.
- [68] —, "Matrix completion from a few entries," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2980–2998, 2010.
- [69] R. H. Keshavan and S. Oh, "OptSpace: A gradient descent algorithm on the grassman manifold for matrix completion," *online*, 2009, [arXiv:0910.5260](https://arxiv.org/abs/0910.5260).
- [70] Z. Khalid, S. Durrani, P. Sadeghi, and R. Kennedy, "Concentration uncertainty principles for signals on the unit sphere," in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3717–3720, 2012.
- [71] E. Kreyszig, *Introductory Functional Analysis with Applications*. Wiley, 1989.
- [72] N. Krislock and H. Wolkowicz, "Euclidean distance matrices and applications," in *Handbook of Semidefinite, Cone and Polynomial Optimization*, 2010.
- [73] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

-
- [74] —, “Nonmetric multidimensional scaling: A numerical method,” *Psychometrika*, vol. 29, no. 2, pp. 115–129, 1964.
- [75] J. B. Kruskal and M. Wish, *Multidimensional scaling*. SAGE Publications, 1978.
- [76] E. Larsson and D. Danev, “Accuracy comparison of LS and squared-range LS for source localization,” *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 916–923, 2010.
- [77] S. H. Lee, S. Lee, H. Song, and H. S. Lee, “Wireless sensor network design for tactical military applications: remote large-scale environments,” in Proc. *Proceedings of the 28th IEEE conference on Military communications*, MILCOM’09, pp. 911–917, 2009.
- [78] C. Li, L. Huang, N. Duric, H. Zhang, and C. Rowe, “An improved automatic time-of-flight picker for medical ultrasound tomography,” *Ultrasonics*, vol. 49, no. 1, pp. 61–72, 2009.
- [79] C. Liu, K. Wu, and T. He, “Sensor localization with ring overlapping based on comparison of received signal strength indicator,” *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 516–518, 2004.
- [80] T. Lokki and V. Pulkki, “Evaluation of geometry-based parametric auralization,” *22nd International Conference: Virtual, Synthetic, and Entertainment Audio*, 2002.
- [81] Z. Luo, W. Ma, A. M. C. So, Y. Ye, and S. Zhang, “Semidefinite relaxation of quadratic optimization problems,” *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.
- [82] N. W. MacLachlan, *Theory and Applications of Mathieu Functions*. Dover Publications, 1964, p. 232.
- [83] S. W. Malone and M. W. Trosset, “A study of the stationary configurations of the stress criterion for metric multidimensional scaling,” Duke University, Tech. Rep., 2000.
- [84] K. V. Mardia and P. E. Jupp, *Directional Statistics*. Wiley, 2009.
- [85] S. Massar and P. Spindel, “Uncertainty relation for the discrete Fourier transform,” *Physical Review Letters*, vol. 100, p. 190401, May 2008.
- [86] R. Mathar, “Algorithms in multidimensional scaling,” in *Conceptual and Numerical Analysis of Data*, O. Opitz, ed. Springer, 1989, pp. 159–177.
- [87] R. Mathar and P. J. F. Groenen, “Algorithms in convex analysis applied to multidimensional scaling,” in *Symbolic-numeric data analysis and learning*, E. Diday and Y. Lechevalier, eds. Nova Science, 1991, pp. 45–56.
- [88] I. McCowan, M. Lincoln, and I. Himawan, “Microphone array shape calibration in diffuse noise fields,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16(3), 2008.
- [89] D. Moore and I. McCowan, “The multichannel overlapping numbers corpus,” IDIAP, Tech. Rep., 2003.
- [90] F. Natterer, “Acoustic mammography in the time domain,” University of Münster, Germany, Tech. Rep., 2008.

-
- [91] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AoA," in Proc. *IN-FOCOM*, vol. 3, pp. 1734–1743, 2003.
- [92] M. Nilges, G. Clore, and A. M. Gronenborn, "Determination of three-dimensional structures of proteins from interproton distance data by hybrid distance geometry-dynamical simulated annealing calculations," *FEBS Letters*, vol. 229, no. 2, pp. 317–324, 1988.
- [93] S. Oh, A. Montanari, and A. Karbasi, "Sensor network localization from local connectivity: Performance analysis for the MDS-MAP algorithm," in Proc. *ITW 2010*, pp. 1–5, 2010.
- [94] R. Parhizkar, Y. Barbotin, and M. Vetterli, "Sequences with minimal time-frequency uncertainty," *Submitted to Applied and Computational Harmonic Analysis (ACHA)*, March 2013.
- [95] R. Parhizkar, A. Karbasi, S. Oh, and M. Vetterli, "Calibration using matrix completion with application to ultrasound tomography," *IEEE Transactions on Signal Processing*, July 2013.
- [96] G. Pataki, "On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues," *Mathematics of Operations Research*, pp. 339–358, 1998.
- [97] P. Pertila, M. Mieskolainen, and M. Hamalainen, "Closed-form self-localization of asynchronous microphone arrays," in Proc. *Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, pp. 139–144, 2011.
- [98] J. Prestin and E. Quak, "Optimal functions for a periodic uncertainty principle and multiresolution analysis," *Proceedings of the Edinburgh Mathematical Society*, vol. 42, no. 2, pp. 225–242, 1999.
- [99] J. Prestin, E. Quak, H. Rauhut, and K. Selig, "On the connection of uncertainty principles for functions on the circle and on the real line," *Journal of Fourier Analysis and Applications*, vol. 9, no. 4, pp. 387–409, 2003.
- [100] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," MIT, Tech. Rep., 2003.
- [101] T. Przebinda, V. DeBrunner, and M. Ozaydin, "Using a new uncertainty measure to determine optimal bases for signal representations," in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1999.
- [102] V. C. Raykar and R. Duraiswami, "Automatic position calibration of multiple microphones," in Proc. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, pp. 69–72, 2004.
- [103] V. C. Raykar, I. V. Kozintsev, and R. Lienhart, "Position calibration of microphones and loudspeakers in distributed computing platforms," *IEEE Transactions on Speech and Audio Processing*, vol. 13(1), 2005.
- [104] B. Recht, "A simpler approach to matrix completion," *The Journal of Machine Learning Research*, vol. 12, pp. 3413–3430, Dec. 2011.

-
- [105] B. Recht, W. Xu, and B. Hassibi, “Null space conditions and thresholds for rank minimization,” *Mathematical Programming*, vol. 127, no. 1, pp. 175–202, Mar 2011.
- [106] F. Ribeiro, D. E. Ba, and C. Zhang, “Turning enemies into friends: Using reflections to improve sound source localization,” *Proceedings of the 2010 IEEE International Conference on Multimedia and Expo (ICME)*, 2010.
- [107] F. Ribeiro, D. A. Florencio, D. E. Ba, and C. Zhang, “Geometrically constrained room modeling with compact microphone arrays,” *IEEE Transactions on Acoustics, Speech and Language Processing*, vol. 20, no. 5, pp. 1449–1460, 2012.
- [108] F. Ribeiro, C. Zhang, D. A. Florencio, and D. E. Ba, “Using reverberation to improve range and elevation discrimination for small array sound source localization,” *IEEE Transactions on Acoustics, Speech and Language Processing*, vol. 18, no. 7, pp. 1781–1792, 2010.
- [109] Y. Rue and S. Mylvaganam, “Performance monitoring of ultrasonic transducers with laser vibrometers,” in Proc. *IEEE Sensors Applications Symposium (SAS)*, pp. 64–69, Feb. 2011.
- [110] J. M. Sachar, H. F. Silverman, and W. R. Patterson, “Microphone position and gain calibration for a large-aperture microphone array,” *IEEE Transactions on Speech and Audio Processing*, vol. 13(1), 2005.
- [111] C. Savarese and J. M. Rabaey, “Locationing in distributed ad-hoc wireless sensor networks,” in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2037–2040, 2001.
- [112] A. Savvides, C. Han, and M. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” in Proc. *7th MobiCom*, pp. 166–179, 2001.
- [113] J. B. Saxe, “Embeddability of weighted graphs in k-space is strongly NP-hard,” in Proc. *Proceedings of the 17th Allerton Conference on Communications, Control, and Computing*, pp. 480–489, 1979.
- [114] I. J. Schoenberg, “Remarks to Maurice Frechet’s article “Sur la definition axiomatique d’une classe d’espace distances vectoriellement applicable sur l’espace de Hilbert,”” *Annals of Mathematics*, vol. 36, no. 3, pp. 724–732, 1935.
- [115] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, “Localization from mere connectivity,” in Proc. *MobiHoc ’03*, pp. 201–212, 2003.
- [116] A. Shapiro, “Rank-reducibility of a symmetric matrix and sampling theory of minimum trace factor analysis,” *Psychometrika*, vol. 47, no. 2, pp. 187–199, 1982.
- [117] X. Sheng and Y.-H. Hu, “Energy based acoustic source localization,” in *Information Processing in Sensor Networks*, Lecture Notes in Computer Science, F. Zhao and L. Guibas, eds. Springer Berlin Heidelberg, 2003, vol. 2634, pp. 285–300.
- [118] D. Slepian, “Prolate spheroidal wave functions, Fourier analysis, and uncertainty. V: The discrete case,” *The Bell Systems Technical Journal*, vol. 57, no. 5, pp. 1371–1430, 1978.

-
- [119] J. Smith and J. Abel, "Closed-form least-squares source location estimation from range-difference measurements," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 12, 1987.
- [120] S. Song Goh and C. A. Micchelli, "Uncertainty principles in Hilbert spaces," *Journal of Fourier Analysis and Applications*, vol. 8, pp. 335–374, 2002.
- [121] M. J. Taghizadeh, R. Parhizkar, P. Garner, and H. Bourlard, "Euclidean distance matrix completion for ad-hoc microphone array calibration," in Proc. *International Conference on Digital Signal Processing*, Santorini, Greece, 2013.
- [122] Y. Takane, F. 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.
- [123] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [124] O. Thiergart, G. Del Galdo, and E. Habets, "On the spatial coherence in mixed sound fields and its application to signal-to-diffuse ratio estimation." *Journal of the Acoustical Society of America*, vol. 132, no. 4, pp. 2337–2346, 2012.
- [125] M. Todd, "Semidefinite optimization," *Acta Numerica*, vol. 10, pp. 515–560, 2001.
- [126] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, pp. 401–419, 1952.
- [127] M. Trnovská, "Strong duality conditions in semidefinite programming," *Journal of Electrical Engineering*, vol. 56, no. 12, 2005.
- [128] M. W. Trosset, "Applications of multidimensional scaling to molecular conformation," *Computing Science and Statistics*, vol. 29, pp. 148–152, 1998.
- [129] M. Trosset and R. Mathar, "On the existence of nonglobal minimizers of the stress criterion for metric multidimensional scaling," *American Statistical Association: Proceedings Statistical Computing Section*, pp. 158–162, 1997.
- [130] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [131] M. Vetterli, J. Kovacevic, and V. K. Goyal, *Foundations of Signal Processing*. online, 2012, <http://www.fourierandwavelets.org/>.
- [132] R. von Mises, "Über die "Ganzzahligkeit" der Atomgewichte und verwandte Fragen," *Physikalische Z.*, vol. 19, pp. 490–500, 1918.
- [133] A. Žilinskas and A. Podlipskytė, "On multimodality of the stress criterion for metric multidimensional scaling," *Informatika*, vol. 14, no. 1, pp. 121–130, 2003.
- [134] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," in Proc. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

- [135] M. P. Williamson, T. F. Havel, and K. Wüthrich, “Solution conformation of proteinase inhibitor IIA from bull seminal plasma by 1h nuclear magnetic resonance and distance geometry,” *Journal of Molecular Biology*, vol. 182, no. 2, pp. 295–315, 1985.
- [136] Y. Xu, Y. Ouyang, Z. Le, J. Ford, and F. Makedon, “Mobile anchor-free localization for wireless sensor networks,” in Proc. *Distributed Computing in Sensor Systems*, 2007.
- [137] G. Young and A. Householder, “Discussion of a set of points in terms of their mutual distances,” *Psychometrika*, vol. 3, no. 1, pp. 19–22, 1938.
- [138] T. Zhu, S. Cheadle, A. Petrella, and S. Gray, “First-arrival tomography: Method and application,” in Proc. *70th Annual International Meeting: Society of Exploration Geophysics*, pp. 2028–2031, 2000.

Curriculum Vitæ

Reza Parhizkar

Audiovisual Communications Laboratory (LCAV)
École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland

Email: reza.parhizkar@epfl.ch
Web-page: <http://lcav.epfl.ch/parhizkar>

Personal

Date of Birth January 23, 1985
Marital Status Single

Education

Sep. 2009–present Ph.D. candidate in School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, Supervisor: Prof. Martin Vetterli
Sep. 2007–Aug. 2009 M.Sc. in Communication Systems, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
Sep. 2003–Aug. 2007 B.Sc. in Electrical Engineering, Sharif University of Technology, Tehran, Iran

Professional experience

Apr. 2013–Jul. 2013 **Interim Engineering Intern**, Qualcomm Inc., San Diego, CA, USA
Feb. 2009–Aug. 2009 **Research Intern**, Nokia research center-Lausanne, Lausanne, Switzerland
Sep. 2007–present **Teaching assistant**, Various master and doctoral level courses, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
Oct. 2006–Jun. 2007 **R&D Member**, Micromodje Industries, Tehran, Iran
Jun. 2006–Sep. 2006 **R&D Intern**, Mahva Technologies, Tehran, Iran

Publications

Journal papers

1. R. Parhizkar, A. Karbasi, S. Oh and M. Vetterli. Calibration Using Matrix Completion with Application to Ultrasound Tomography, *IEEE Transactions on Signal Processing*, Jul. 2013.
2. I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu and M. Vetterli. Acoustic Echoes Reveal Room Shape, *Proceedings of the National Academia of Science (PNAS)*, Jun. 2013.
3. R. Parhizkar, Y. Barbotin, and M. Vetterli. Sequences with Minimal Time-Frequency Uncertainty, Submitted to *Applied and Computational Harmonic Analysis (ACHA)*, Mar. 2013.
4. R. Parhizkar, I. Dokmanić, and M. Vetterli. Euclidean Distance Matrices For Signal Processing, In Preparation for *Signal Processing Magazine*.
5. R. Parhizkar, S. HaghghatShoar, and M. Vetterli. Least-Squares Sensor Localization: Theory and Distributed Algorithm, In Preparation for *IEEE Transactions on Signal Processing*.

Conference papers

1. O. Roy, I. Jovanovic, A. Hormati, R. Parhizkar and M. Vetterli. Sound Speed Estimation Using Wave-based Ultrasound Tomography: Theory and GPU Implementation. *SPIE Medical Imaging, San Diego*. 2010
2. A. Karbasi, S. Oh, R. Parhizkar and M. Vetterli. Ultrasound Tomography Calibration Using Structured Matrix Completion. *The 20th International Congress on Acoustics, Sydney, Australia*. 2010
3. R. Parhizkar, A. Karbasi and M. Vetterli. Calibration in Circular Ultrasound Tomography Devices. *The 36th International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic*. 2011
4. R. Parhizkar, Y. Barbotin and M. Vetterli. Sequences with Minimal Time-Frequency Spreads. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, Canada*. 2013
5. M. Taghizadeh, R. Parhizkar, P. Garner and H. Boursard. Euclidean Distance Matrix Completion for Ad-hoc Microphone Array Calibration. *International Conference on Digital Signal Processing, Santorini, Greece*. 2013
6. R. Parhizkar, and M. Vetterli. Euclidean Distance Matrices Can Help You Design Keyboards. In Preparation for *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014), Florence, Italy*.
7. R. Parhizkar, I. Dokmanic, and M. Vetterli. In-Room Localization Using Euclidean Distance Matrices. In Preparation for *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014), Florence, Italy*.

Reports

1. R. Parhizkar. Estimating Physical Parameters in a Diffusive Field. *Technical Report, EPFL, Lausanne, Switzerland.* 2008
2. R. Parhizkar. Acoustic Tomography Based on Wave Equation. *Technical Report, EPFL, Lausanne, Switzerland.* 2008
3. R. Parhizkar. Spatial Error Concealment in Ad-hoc Audio Conferencing Systems. *Master Thesis, EPFL, Lausanne, Switzerland.* 2009
4. G. Baechler, R. Parhizkar and M. Vetterli. Multi-Modal Sensor Localization. *Technical Report, EPFL, Lausanne, Switzerland.* 2013

Patents

1. I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu and M. Vetterli, A Method and a System for Determining the Geometry and/or the Localization of an Object, , Filed: *US13/828,761*, April 2013.
2. G. Baechler, R. Parhizkar and M. Vetterli, Indoor Localization Method and System, Filed: *2013CH-0611*, March 2013.

Awards and honors

- 2013 Co-author of a research that appeared in more than 50 news websites including [BBC's front page](#), [NBC News](#), [TIME](#) and [Discovery News](#).
- 2011-2012 President of the Iranian Students Association at EPFL (IRSA).
- 2011 Recipient of the Award for research with exceptional value from the dean of the Communications and Computer Sciences Department, EPFL.
- 2011 Recipient of the Best Student Paper Award in the International Conference on Acoustics, Speech and Signal Processing, ICASSP for the paper Calibration in Circular Ultrasound Tomography Devices, Prague.
- 2007-2009 Awarded scholarship for the M.Sc. program in Communication Systems Department, EPFL, Lausanne, Switzerland.
- 2008 Recipient of the MICS (Mobile Information & Communication Systems) Program scholarship for an internship, Switzerland.
- 2003 Ranked 21st among over 400,000 applicants in Iran's Nationwide University Entrance Exam (Konkour) for Engineering and Applied Mathematics, Iran.

Computer skills

C/C++ , Java, MATLAB, Mathematica, L^AT_EX, Microsoft Office

Languages

Azeri, Farsi (Native), English (Fluent), French, Arabic (Intermediate)