

# RELAX AND UNFOLD: MICROPHONE LOCALIZATION WITH EUCLIDEAN DISTANCE MATRICES

*Ivan Dokmanić, Juri Ranieri, and Martin Vetterli*

School of Computer and Communication Sciences  
Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland  
{ivan.dokmanic, martin.vetterli}@epfl.ch

## ABSTRACT

Recent methods for microphone position calibration work with sound sources at *a priori* unknown locations. This is convenient for *ad hoc* arrays, as it requires little additional infrastructure. We propose a flexible localization algorithm by first recognizing the problem as an instance of multidimensional unfolding (MDU)—a classical problem in Euclidean geometry and psychometrics—and then solving the MDU as a special case of Euclidean distance matrix (EDM) completion. We solve the EDM completion using a semidefinite relaxation. In contrast to existing methods, the semidefinite formulation allows us to elegantly handle missing pairwise distance information, but also to incorporate various prior information about the distances between the pairs of microphones or sources, bounds on these distances, or ordinal information such as “microphones 1 and 2 are more apart than microphones 1 and 15”. The intuition that this should improve the localization performance is justified by numerical experiments.

**Index Terms**—Microphone localization, array calibration, microphone array, Euclidean distance matrix, semidefinite relaxation, multidimensional unfolding

## 1. INTRODUCTION

In many applications involving multiple microphones we need to know their relative positions. Modern methods for microphone localization use arbitrarily positioned sound sources, as opposed to using specialized calibration rigs with fixed geometry [2, 3, 4, 5]. Some of these methods can also deal with unknown times at which the acoustic events appear [2, 3].

On the other hand, the mentioned methods (with the notable exception of [5]) require a full set of distance measurements. Furthermore, it is not straightforward to add *a priori* information about the relative geometry of the microphones or of the acoustic events. Such side information may take any of the following forms:

1. Distance between microphones 5 and 10 is 10 cm;
2. Calibration events 3, 4, and 5 are all within a distance of 20 cm;
3. Distance between microphones 1 and 2 is between 20 and 30 cm;
4. Distance between microphones 1 and 2 is larger than the distance between microphones 1 and 3.

This work was supported by an ERC Advanced Grant—Support for Frontier Research—SPARSAM Nr: 247006, and a Google PhD Fellowship.

Portions of this work are part of a longer paper accepted for publication in IEEE Signal Processing Magazine [1].

Intuition suggests that prior knowledge should improve calibration performance. As we show later in numerical experiments, this is indeed observed in practice.

To address potentially missing distances between the sources and the microphones, and to incorporate the prior information about the geometry, we propose to address microphone localization as a special case of Euclidean distance matrix (EDM) completion.

We do it in two steps. First, we recognize the calibration problem as an instance of multidimensional unfolding (MDU)—a set of tools used for data visualization in psychometrics. The MDU was addressed in detail by Schönemann in 1970 [6]. Second, we solve the MDU as a Euclidean distance matrix (EDM) completion problem.

EDMs are simply matrices of squared distances between points. Algebraic manipulation allows us to establish a linear relationship between the Gram matrix and the EDM, and then to use semidefinite programming to solve the completion problem. The full semidefinite program has a rank constraint, so what we actually use is the semidefinite relaxation (SDR).

A constraint in our approach is that to use EDMs we need to work with distances, so we need to assume some synchronization between the microphones and the sources. The more challenging problem of non-synchronized microphones and sources (see, *e.g.*, [3]) is beyond the scope of this paper.

## 2. MICROPHONE CALIBRATION AND MULTIDIMENSIONAL UNFOLDING

We define the microphone localization problem as the task of finding the locations of  $m$  microphones given their distances to  $k$  acoustic events.

**Problem 1.** Denote by  $\mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_m] \in \mathbb{R}^{d \times m}$  the unknown microphone locations, and by  $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_k] \in \mathbb{R}^{d \times k}$  the unknown source locations. The ambient dimensionality is denoted by  $d$  (usually  $d = 2$  or  $d = 3$ ). The distance between the  $i$ th microphone and  $j$ th source is

$$\delta_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\|^2, \quad (1)$$

where  $\|\cdot\|$  denotes the Euclidean norm. We collect these distances in the matrix  $\Delta$ , and the task is then to recover  $\mathbf{R}$  from  $\Delta$ .

This standard scenario is described for example in [5]. The first key step we make is that we recognize this problem as an instance of metric multidimensional unfolding (MDU) [6]. Indeed, MDU is defined as the problem of localizing a set of points partitioned into two subsets, where we can measure the distances only between the points belonging to different subsets.

One of the early approaches to metric MDU is described by Schönemann [6]. We go through the steps of the algorithm, and then explain how we can solve the problem using EDMs. The goal is to make a comparison between the MDU-specific approach and a more general EDM formalization, and to emphasize the universality and simplicity of the EDM approach.

We can write the definition of  $\delta_{ij}$  (1) in the matrix form as

$$\Delta = \mathcal{D}(\mathbf{R}, \mathbf{S}) \stackrel{\text{def}}{=} \text{diag}(\mathbf{R}^\top \mathbf{R}) \mathbf{1}^\top - 2\mathbf{R}^\top \mathbf{S} + \mathbf{1} \text{diag}(\mathbf{S}^\top \mathbf{S}), \quad (2)$$

Consider now two geometric centering matrices of sizes  $m$  and  $k$ , denoted  $\mathbf{J}_m$  and  $\mathbf{J}_k$ . The geometric centering matrix of size  $n$  is defined as

$$\mathbf{J} = \mathbf{I} - \frac{1}{n} \mathbf{1} \mathbf{1}^\top. \quad (3)$$

Using this definition, we have

$$\mathbf{R} \mathbf{J}_m = \mathbf{R} - \mathbf{r}_c \mathbf{1}^\top, \quad \mathbf{S} \mathbf{J}_k = \mathbf{S} - \mathbf{s}_c \mathbf{1}^\top, \quad (4)$$

where  $\mathbf{r}_c$  is the geometric center of the microphones in  $\mathbf{R}$ , and  $\mathbf{s}_c$  is the geometric center of the sources in  $\mathbf{S}$ . This means that

$$\mathbf{J}_m \Delta \mathbf{J}_k = \tilde{\mathbf{R}}^\top \tilde{\mathbf{S}} \stackrel{\text{def}}{=} \tilde{\mathbf{G}} \quad (5)$$

is a matrix of inner products between vectors  $\tilde{\mathbf{r}}_i$  and  $\tilde{\mathbf{s}}_j$ . We used tildes to differentiate this from *real* inner products between  $\mathbf{r}_i$  and  $\mathbf{s}_j$ , because in (5), the points in  $\tilde{\mathbf{R}}$  and  $\tilde{\mathbf{S}}$  are referenced to different coordinate systems. The centroids  $\mathbf{r}_c$  and  $\mathbf{s}_c$  generally do not coincide. There are different ways to decompose  $\tilde{\mathbf{G}}$  into a product of two full rank matrices  $\mathbf{A}$  and  $\mathbf{B}$ ,

$$\tilde{\mathbf{G}} = \mathbf{A}^\top \mathbf{B}. \quad (6)$$

We could for example use the SVD,  $\tilde{\mathbf{G}} = \mathbf{U} \Sigma \mathbf{V}^\top$ , and set  $\mathbf{A}^\top = \mathbf{U}$  and  $\mathbf{B} = \Sigma \mathbf{V}^\top$ . Any two such decompositions are linked by some invertible transformation  $\mathbf{T} \in \mathbb{R}^{d \times d}$ ,

$$\tilde{\mathbf{G}} = \mathbf{A}^\top \mathbf{B} = \tilde{\mathbf{R}}^\top \mathbf{T}^{-1} \mathbf{T} \tilde{\mathbf{S}}. \quad (7)$$

We can now write down the conversion rule between the sought microphone positions and the matrices  $\mathbf{A}$  and  $\mathbf{B}$  which we can compute from  $\Delta$ ,

$$\begin{aligned} \mathbf{R} &= \mathbf{T}^\top \mathbf{A} + \mathbf{r}_c \mathbf{1}^\top \\ \mathbf{S} &= (\mathbf{T}^{-1})^\top \mathbf{B} + \mathbf{s}_c \mathbf{1}^\top, \end{aligned} \quad (8)$$

where  $\mathbf{A}$  and  $\mathbf{B}$  can be computed according to (6). Because we cannot reconstruct the absolute position of the point configuration, we can arbitrarily set  $\mathbf{r}_c = 0$ , and  $\mathbf{s}_c = \alpha \mathbf{e}_1$ . Recapitulating, we have that

$$\Delta = \mathcal{D}(\mathbf{T}^\top \mathbf{A}, (\mathbf{T}^{-1})^\top \mathbf{B} + \alpha \mathbf{e}_1 \mathbf{1}^\top), \quad (9)$$

and the problem is reduced to computing  $\mathbf{T}$  and  $\alpha$  so that (9) hold, or in other words, so that the right hand side be consistent with the data  $\Delta$ . We reduced MDU to a relatively small problem: in 3D, we need to compute only ten scalars. Schönemann [6] gives an algebraic method to find these parameters, and mentions the possibility of least squares, while Crocco, Bue and Murino [5] propose a different approach using non-linear least squares.

The described procedure seems quite convoluted. Rather, we see MDU as a special case of matrix completion. To explain how, we first introduce Euclidean distance matrices.

### 3. EUCLIDEAN DISTANCE MATRICES

An EDM corresponding to the point set  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$  is the matrix  $\mathbf{D}$  defined as

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2, \quad (10)$$

[1] where  $\|\cdot\|$  is the Euclidean norm. By expanding the norm, we see that  $\mathbf{D}$  can be computed as

$$\mathbf{D} = \mathcal{K}(\mathbf{G}) \stackrel{\text{def}}{=} \text{diag}(\mathbf{G}) \mathbf{1}^\top + \mathbf{1} \text{diag}(\mathbf{G})^\top - 2\mathbf{G}, \quad (11)$$

where  $\mathbf{G} \stackrel{\text{def}}{=} \mathbf{X}^\top \mathbf{X}$  is the Gram matrix. A straightforward computation shows that  $\mathbf{G}$  can be obtained from  $\mathbf{D}$  as

$$\mathbf{G} = -\frac{1}{2} \mathbf{J} \mathbf{D} \mathbf{J}, \quad (12)$$

where  $\mathbf{J}$  is the geometric centering matrix of size  $n$ . This holds because the nullspace of  $\mathbf{J}$  contains the rank-one matrices that appear in (11); that is, the terms  $\text{diag}(\mathbf{G}) \mathbf{1}^\top + \mathbf{1} \text{diag}(\mathbf{G})^\top$  are annihilated by  $\mathbf{J}$ . We note that many different Gramians  $\mathbf{G}$  (corresponding to different translations of the point set) yield the same EDM  $\mathbf{D}$ . Relation (12) then computes a particular  $\mathbf{G}$ , namely that one for which the centroid of the point set is at the origin (hence the name for  $\mathbf{J}$ ). For further details and intuitions about this, see [1].

#### 3.1. EDM Completion

In a typical EDM problem, we have to localize the point set, but the distances are noisy and some are missing. TOA measurements (hence the distances) are subject to noise, sampling errors and model mismatch.

Missing entries arise because of limited range or occlusions, or because the nodes are asymmetric by definition. In microphone calibration we have two types: microphones and calibration sources. This results in a particular block structure of the missing entries (we will come back to this later, but you can fast-forward to Fig. 1 for an illustration). We summarize the denoising and the completion problems as follows:

**Problem 2.** *Let  $\mathbf{D}$  be an EDM corresponding to a point set  $\mathbf{X}$ . We are given a noisy observation of the distances between  $p \leq \binom{n}{2}$  pairs of points from  $\mathbf{X}$ . That is, we have a noisy measurement of  $2p$  entries in  $\mathbf{D}$ ,*

$$\tilde{d}_{ij} = d_{ij} + \epsilon_{ij}, \quad (13)$$

for  $(i, j) \in E$ , where  $E$  is some index set, and  $\epsilon_{ij}$  absorbs all errors. The goal is to reconstruct the point set  $\hat{\mathbf{X}}$  in the given embedding dimension, so that the entries of  $\mathcal{K}(\hat{\mathbf{X}}^\top \hat{\mathbf{X}})$  are close in some metric to the observed entries  $\tilde{d}_{ij}$ .

To concisely write down completion problems, we define the mask matrix  $\mathbf{W}$  as follows,

$$w_{ij} \stackrel{\text{def}}{=} \begin{cases} 1, & (i, j) \in E \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

This matrix then selects elements of an EDM through a Hadamard (entrywise) product. For example, to compute the norm of the difference between the observed entries in  $\mathbf{A}$  and  $\mathbf{B}$ , we write  $\|\mathbf{W} \circ (\mathbf{A} - \mathbf{B})\|$ .

## 4. SEMIDEFINITE RELAXATION

### 4.1. Semidefinite Programming

A characterization of EDMs by Gower [7] states that  $\mathbf{D}$  is an EDM if and only if the corresponding geometrically centered Gram matrix  $-\frac{1}{2}\mathbf{J}\mathbf{D}\mathbf{J}$  is positive-semidefinite. Thus, it establishes a one-to-one correspondence between the cone of EDMs, denoted by  $\mathbb{EDM}^n$ , and the intersection of the symmetric positive-semidefinite cone  $\mathbb{S}_+^n$  with the geometrically centered cone  $\mathbb{S}_c^n$ . The latter is defined as the set of all symmetric matrices whose column sum vanishes,

$$\mathbb{S}_c^n = \left\{ \mathbf{G} \in \mathbb{R}^{n \times n} \mid \mathbf{G} = \mathbf{G}^\top, \mathbf{G}\mathbf{1} = \mathbf{0} \right\}. \quad (15)$$

We can use this correspondence to cast EDM completion and approximation as semidefinite programs. While the above characterization describes an EDM of an  $n$ -point configuration in any dimension, we are often interested in situations where  $d \ll n$ . It is easy to adjust for this case by requiring that the rank of the centered Gram matrix be bounded. One can verify that

$$\left. \begin{array}{l} \mathbf{D} = \mathcal{K}(\mathbf{X}^\top \mathbf{X}) \\ \text{affdim}(\mathbf{X}) \leq d \end{array} \right\} \iff \left\{ \begin{array}{l} -\frac{1}{2}\mathbf{J}\mathbf{D}\mathbf{J} \succeq 0 \\ \text{rank}(\mathbf{J}\mathbf{D}\mathbf{J}) \leq d, \end{array} \right. \quad (16)$$

when  $n \geq d$ . That is, EDMs with a particular embedding dimension  $d$  are completely characterized by the rank and definiteness of  $\mathbf{J}\mathbf{D}\mathbf{J}$ .

Now we can write the following rank-constrained semidefinite program for solving Problem 2,

$$\begin{aligned} & \underset{\mathbf{G}}{\text{minimize}} && \left\| \mathbf{W} \circ (\tilde{\mathbf{D}} - \mathcal{K}(\mathbf{G})) \right\|_F^2 \\ & \text{subject to} && \text{rank}(\mathbf{G}) \leq d \\ & && \mathbf{G} \in \mathbb{S}_+^n \cap \mathbb{S}_c^n, \end{aligned} \quad (17)$$

where  $\mathbf{W}$  is the mask matrix that selects only the observed entries. The cost function computes how far  $\tilde{\mathbf{D}}$  is from the estimated EDM  $\mathcal{K}(\mathbf{G})$  on the observed entries. The second constraint is just a shorthand for writing  $\mathbf{G} \succeq 0$ ,  $\mathbf{G}\mathbf{1} = \mathbf{0}$ . We note that this corresponds to minimizing the well-known s-stress cost function [8].

### 4.2. Relaxation

Unfortunately, the rank property makes the feasible set in (17) non-convex, and solving it exactly becomes difficult. This makes sense, as we know that s-stress is not convex. Nevertheless, we may *relax* the hard problem, by simply omitting the rank constraint, and hope to obtain a solution with the correct dimensionality,

$$\begin{aligned} & \underset{\mathbf{G}}{\text{minimize}} && \left\| \mathbf{W} \circ (\tilde{\mathbf{D}} - \mathcal{K}(\mathbf{G})) \right\|_F^2 \\ & \text{subject to} && \mathbf{G} \in \mathbb{S}_+^n \cap \mathbb{S}_c^n. \end{aligned} \quad (18)$$

We call (18) a semidefinite relaxation (SDR) of the rank-constrained program (17). If the embedding dimension of the solution is larger than desired, we project it in the lower-dimensional space by thresholding the eigenvalues of  $\mathbf{G}$ .

The constraint  $\mathbf{G} \in \mathbb{S}_c^n$ , or equivalently,  $\mathbf{G}\mathbf{1} = \mathbf{0}$ , means that there are no strictly positive definite solutions ( $\mathbf{G}$  has a nullspace, so at least one eigenvalue must be zero). In other words, there exist no strictly feasible points [9]. This may pose a numerical problem, especially for various interior point methods. The idea is then to reduce the size of the Gram matrix through an invertible transformation, somehow removing the part of it responsible for the nullspace.

---

### Algorithm 1 Semidefinite Relaxation (Matlab/CVX)

---

```

1 function [EDM, X] = sdr_complete_edm(D, W, lambda)
2
3 n = size(D, 1);
4 x = -1/(n + sqrt(n));
5 y = -1/sqrt(n);
6 V = [y*ones(1, n-1); x*ones(n-1) + eye(n-1)];
7 e = ones(n, 1);
8
9 cvx_begin sdp
10     variable G(n-1, n-1) symmetric;
11     B = V*G*V';
12     E = diag(B)*e' + e*diag(B)' - 2*B;
13     maximize trace(G) ...
14         - lambda * norm(W .* (E - D), 'fro');
15     subject to
16         G >= 0;
17 cvx_end
18
19 [U, S, V] = svd(B);
20 EDM = diag(B)*e' + e*diag(B)' - 2*B;
21 X = sqrt(S)*V';

```

---

In what follows, we describe how to construct this smaller Gram matrix.

A different, equivalent way to phrase Gower's EDM characterization [7] is by the following statement: a symmetric hollow matrix  $\mathbf{D}$  is an EDM if and only if it is negative semidefinite on  $\{\mathbf{1}\}^\perp$  (on all vectors  $\mathbf{t}$  such that  $\mathbf{t}^\top \mathbf{1} = 0$ ). Let us construct an orthonormal basis for this orthogonal complement—a subspace of dimension  $(n-1)$ —and arrange it in the columns of matrix  $\mathbf{V} \in \mathbb{R}^{n \times (n-1)}$ . We demand

$$\begin{aligned} \mathbf{V}^\top \mathbf{1} &= \mathbf{0} \\ \mathbf{V}^\top \mathbf{V} &= \mathbf{I}. \end{aligned} \quad (19)$$

There are many possible choices for  $\mathbf{V}$ , but all of them obey that  $\mathbf{V}\mathbf{V}^\top = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top = \mathbf{J}$ . The following choice is given in [10],

$$\mathbf{V} = \begin{bmatrix} p & p & \cdots & p \\ 1+q & q & \cdots & q \\ q & 1+q & \cdots & q \\ \vdots & \cdots & \ddots & \vdots \\ q & q & \cdots & 1+q \end{bmatrix}, \quad (20)$$

where  $p = -1/(n + \sqrt{n})$  and  $q = -1/\sqrt{n}$ .

With the help of the matrix  $\mathbf{V}$ , we can now construct the sought Gramian with reduced dimensions. For an EDM  $\mathbf{D} \in \mathbb{R}^{n \times n}$ ,

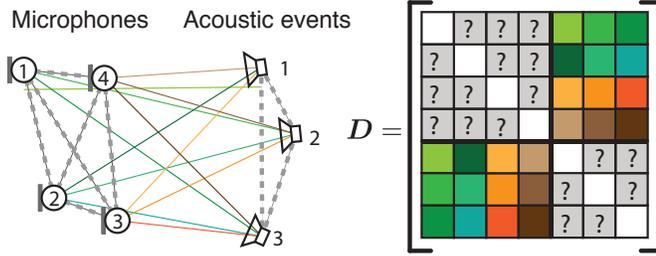
$$\mathcal{G}(\mathbf{D}) \stackrel{\text{def}}{=} -\frac{1}{2}\mathbf{V}^\top \mathbf{D}\mathbf{V} \quad (21)$$

is an  $(n-1) \times (n-1)$  PSD matrix. This can be verified by substituting (21) in (11). Additionally, we have that

$$\mathcal{K}(\mathbf{V}\mathcal{G}(\mathbf{D})\mathbf{V}^\top) = \mathbf{D}. \quad (22)$$

Indeed,  $\mathbf{H} \mapsto \mathcal{K}(\mathbf{V}\mathbf{H}\mathbf{V}^\top)$  is an invertible mapping from  $\mathbb{S}_+^{n-1}$  to  $\mathbb{EDM}^n$  whose inverse is exactly  $\mathcal{G}$ . Using these notations we can write down an equivalent optimization program that is numerically more stable than (18) [10]:

$$\begin{aligned} & \underset{\mathbf{H}}{\text{minimize}} && \left\| \mathbf{W} \circ (\tilde{\mathbf{D}} - \mathcal{K}(\mathbf{V}\mathbf{H}\mathbf{V}^\top)) \right\|_F^2 \\ & \text{subject to} && \mathbf{H} \in \mathbb{S}_+^{n-1}. \end{aligned} \quad (23)$$



**Fig. 1:** Microphone calibration as an example of MDU. We can measure only the propagation times from acoustic sources at unknown locations, to microphones at unknown locations. The corresponding revealed part of the EDM has a particular off-diagonal structure, leading to a special case of EDM completion.

On the one hand, with the above transformation the constraint  $\mathbf{G}\mathbf{1} = \mathbf{0}$  became implicit in the objective, as  $\mathbf{V}\mathbf{H}\mathbf{V}^T\mathbf{1} \equiv \mathbf{0}$  by (19); on the other hand, the feasible set is now the full semidefinite cone  $\mathbb{S}_+^{n-1}$ . Still, as Krislock & Wolkowicz mention [9], by omitting the rank constraint we allow the points to move about in a larger space, so we may end up with a higher-dimensional solution even if there is a completion in dimension  $d$ .

There exist various heuristics for promoting lower rank. One such heuristic involves the trace norm—the convex envelope of rank. The trace or nuclear norm is studied extensively by the compressed sensing community. In contrast to the common wisdom in compressed sensing, the trick here is to maximize the trace norm, not to minimize it. The mechanics are as follows: maximizing the sum of squared distances between the points will stretch the configuration as much as possible, subject to available constraints. But stretching favors smaller affine dimensions (imagine pulling out a roll of paper, or stretching a bent string). Maximizing the sum of squared distances can be rewritten as maximizing the sum of norms in a centered point configuration—but that is exactly the trace of the Gram matrix  $\mathbf{G} = -\frac{1}{2}\mathbf{J}\mathbf{D}\mathbf{J}$  [11]. This idea has been successfully put to work by Weinberger and Saul [11] in manifold learning, and by Biswas *et al.* in SNL [12].

Noting that  $\text{trace}(\mathbf{H}) = \text{trace}(\mathbf{G})$  because  $\text{trace}(\mathbf{J}\mathbf{D}\mathbf{J}) = \text{trace}(\mathbf{V}^T\mathbf{D}\mathbf{V})$ , we write the following SDR,

$$\begin{aligned} & \underset{\mathbf{H}}{\text{maximize}} && \text{trace}(\mathbf{H}) - \lambda \left\| \mathbf{W} \circ (\tilde{\mathbf{D}} - \mathcal{K}(\mathbf{V}\mathbf{H}\mathbf{V}^T)) \right\|_F \\ & \text{subject to} && \mathbf{H} \in \mathbb{S}_+^{n-1} \end{aligned} \quad (24)$$

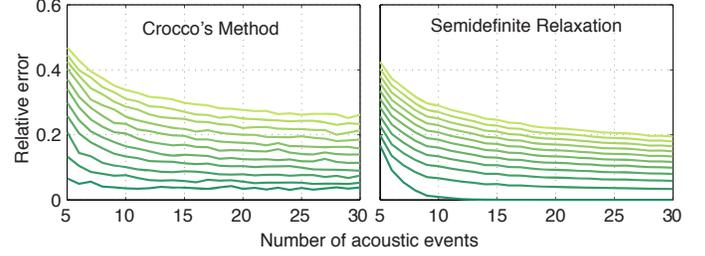
Here we opted to include the data fidelity term in the Lagrangian form, as proposed by Biswas [12], but it could also be moved to constraints. Matlab/CVX code for this relaxation is given in Algorithm 1 [13].

### 4.3. Unfolding as Matrix Completion

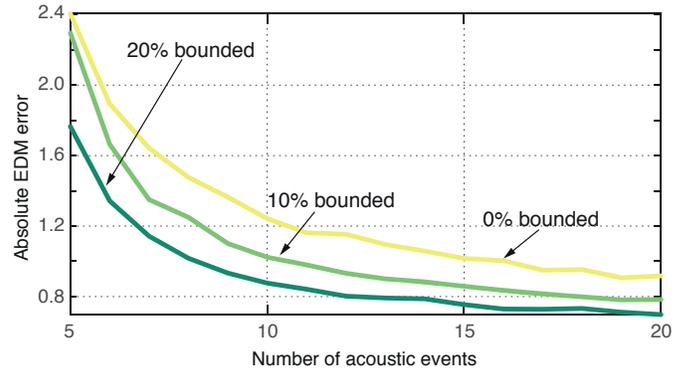
We now frame Problem 1 (MDU) as a special case of Problem 2 (EDM completion). Let the microphones and the sources be represented by a set of  $n = k + m$  points, ascribed to the columns of matrix  $\mathbf{X} = [\mathbf{R} \ \mathbf{S}]$ . Then  $\mathcal{K}(\mathbf{X}^T\mathbf{X})$  has a special structure as seen in Fig. 1,

$$\mathcal{K}(\mathbf{X}) = \begin{bmatrix} \mathcal{K}(\mathbf{R}) & \mathcal{D}(\mathbf{R}, \mathbf{S}) \\ \mathcal{D}(\mathbf{S}, \mathbf{R}) & \mathcal{K}(\mathbf{S}) \end{bmatrix}, \quad (25)$$

where the upper-right and the lower-left blocks are measured, and the diagonal blocks are unknown. Thus we define the mask matrix



**Fig. 2:** Comparison of two algorithms applied to multidimensional unfolding with varying number of acoustic events  $k$  and noisy distances. For every number of acoustic events, we generated 1000 realizations of  $m = 20$  microphone locations uniformly at random in a unit cube. In addition to the number of acoustic events, we varied the amount of random jitter added to the distances. Jitter was drawn from a centered uniform distribution, with the level increasing in the direction of the arrow, from  $\mathcal{U}[0, 0]$  (no jitter) for the darkest curve at the bottom, to  $\mathcal{U}[-0.15, 0.15]$  for the lightest curve at the top, in 11 increments. For every jitter level, we plotted the mean relative error  $\|\hat{\mathbf{D}} - \mathbf{D}\|_F / \|\mathbf{D}\|_F$  for all algorithms.



**Fig. 3:** Comparison of the influence of prior information on the reconstruction accuracy. For every number of acoustic events, we generated 500 realizations of 15 microphones inside the unit cube (acoustic event were also generated inside the unit cube). Uniform jitter between  $\pm 7$  cm was added to the distances measurements and then varying percentage of microphone pairwise distances was revealed within  $\pm 15\%$  of the true distance. The curves show the absolute reconstruction error  $\|\hat{\mathbf{D}} - \mathbf{D}\|_F$ . Mean value of the Frobenius norm of the true EDM was around 13 for 20 acoustic events, so the absolute error of 1 can be considered as successful localization.

for MDU as

$$\mathbf{W}_{\text{MDU}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{0}_{m \times m} & \mathbf{1}_{m \times k} \\ \mathbf{1}_{k \times m} & \mathbf{0}_{k \times k} \end{bmatrix}. \quad (26)$$

With this matrix, we can simply invoke the SDR in Algorithm 1.

### 4.4. Integrating Prior Knowledge

The attractiveness of the formulation (24) lies in its flexibility: it is easy to insert many useful constraints. Unlike with the methods that are developed specifically for microphone calibration, if we miss one or more source-microphone distances, (24) still works with a trivial modification of the mask matrix  $\mathbf{W}$ . Such situations arise commonly in practice.

We can also easily insert estimates or bounds on the distances. For example, we may have a rough idea of the distances between some pairs of microphones, as well as a rough idea of the distances between some pairs of acoustic events. All of these constraints are simply linear constraints on  $\mathbf{D} = \mathcal{K}(\mathbf{V}\mathbf{H}\mathbf{V}^\top)$ . Let us give some examples. Upper and lower bounds can be specified as follows

$$\begin{aligned} \mathbf{W}_L \circ [\mathcal{K}(\mathbf{V}\mathbf{H}\mathbf{V}^\top) - \mathbf{L}] &\geq 0 \\ \mathbf{W}_U \circ [\mathcal{K}(\mathbf{V}\mathbf{H}\mathbf{V}^\top) - \mathbf{U}] &\leq 0, \end{aligned} \quad (27)$$

where matrices  $\mathbf{W}_L$  and  $\mathbf{W}_U$  select the elements for which we have bounds, and  $\mathbf{L}$  and  $\mathbf{U}$  contain the bounds.

We can also plug in ordinal information about the distances. For example, we can require that the distance between the microphones  $i$  and  $j$  be smaller than the distance between microphones  $i$  and  $k$ . This is written simply as

$$\mathbf{e}_i^\top \mathcal{K}(\mathbf{V}\mathbf{H}\mathbf{V}^\top)(\mathbf{e}_j - \mathbf{e}_k) \leq 0, \quad (28)$$

where  $\mathbf{e}_i$  is the  $i$ th canonical basis vector. Many other useful constraints can be modeled as linear constraints on  $\mathcal{K}$ .

## 5. NUMERICAL EXPERIMENTS

We demonstrate the usefulness of the semidefinite relaxation in two different scenarios. First, we address the problem of localizing the microphone array from the set of pairwise distance measurements in the presence of noise, and compare the SDR with the method of Crocco, Bue and Murino [5]. The results are illustrated in Fig. 2. We observe that for the noiseless case, the SDR performs better, giving perfect reconstruction for all but the lowest number of acoustic events. It also yields more accurate reconstructions at higher noise levels. When the number of acoustic events is low, the method from [5] performs better.

Second, we explore the effect of adding prior information on the distances between the microphones in Fig. 3. We add considerable jitter to measurements ( $\pm 7$  cm within a meter is surpassed by any method for measuring distance using times of flight). We then include additional constraints: upper and lower bounds at 15% of the true distance for varying percentage of the pairwise microphone distances. It is clearly observed that adding prior information can improve the localization performance. A comparison with other approaches is not possible as they do not allow to easily integrate such constraints.

## 6. CONCLUSION

We have shown how to rephrase the microphone localization problem as a Euclidean distance matrix completion problem. We propose to solve it using a semidefinite relaxation. Our formulation can handle missing source-microphone distance, and it allows us to easily integrate arbitrary linear constraints on the distance matrix. These constraints may represent various bounds and relations between the distances in the source-microphone system.

An important challenge is to extend this method to unknown source emission times. The matrix that corresponds to this case exhibits no straightforward rank property that we could exploit.

Another challenge is to better enforce the rank property. Trace maximization does a good job when the number of microphones and the number of calibration events is sufficiently large. But for small arrays and large noise, the solution obtained is not realizable in the

desired embedding dimension, so the performance of the method deteriorates, unlike in [5]. We should seek efficient methods to enforce the correct rank in these smaller-size problems.

## References

- [1] I. Dokmanić, R. Parhizkar, J. Ranieri, and M. Vetterli, “Euclidean Distance Matrices: Essential Theory, Algorithms, and Applications,” *IEEE Signal Process. Mag.*, 2015.
- [2] M. Pollefeys and D. Nister, “Direct Computation of Sound and Microphone Locations From Time-Difference-of-Arrival Data,” in Proc. *IEEE ICASSP*, Las Vegas, 2008, pp. 2445–2448, IEEE.
- [3] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens, “Auto-Localization in Ad-Hoc Microphone Arrays,” in Proc. *IEEE ICASSP*, Vancouver, 2013, pp. 106–110, IEEE.
- [4] M. Crocco, A. Del Bue, M. Bustreo, and V. Murino, “A Closed Form Solution to the Microphone Position Self-Calibration Problem,” in Proc. *IEEE ICASSP*, Kyoto, 2012, pp. 2597–2600.
- [5] M. Crocco, A. D. Bue, and V. Murino, “A Bilinear Approach to the Position Self-Calibration of Multiple Sensors,” *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 660–673, 2012.
- [6] P. H. Schönemann, “On Metric Multidimensional Unfolding,” *Psychometrika*, vol. 35, no. 3, pp. 349–366, 1970.
- [7] J. C. Gower, “Euclidean Distance Geometry,” *Math. Sci.*, vol. 7, no. 1, pp. 1–14, 1982.
- [8] W. S. Torgerson, “Multidimensional scaling: I. Theory and method,” *Psychometrika*, vol. 17, no. 4, pp. 401–419, Dec. 1952.
- [9] N. Krislock and H. Wolkowicz, “Euclidean Distance Matrices and Applications,” in Proc. *Handbook on Semidefinite, Conic and Polynomial Optimization*, pp. 879–914. Springer US, Boston, MA, Jan. 2012.
- [10] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, “Solving Euclidean Distance Matrix Completion Problems via Semidefinite Programming,” *Computational Optimization and Applications*, vol. 12, no. 1-3, pp. 13–30, 1999.
- [11] K. Q. Weinberger and L. K. Saul, “Unsupervised Learning of Image Manifolds by Semidefinite Programming,” *Int. J. Comput. Vision*, vol. 70, no. 1, pp. 77–90, 2006.
- [12] P. Biswas, T. C. Liang, K. C. Toh, Y. Ye, and T. C. Wang, “Semidefinite Programming Approaches for Sensor Network Localization With Noisy Distance Measurements,” *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, 2006.
- [13] M. Grant and S. Boyd, “CVX: Matlab Software for Disciplined Convex Programming, version 2.1,” <http://cvxr.com/cvx>, Mar. 2014.