SIAM J. MATRIX ANAL. APPL.
Vol. 41, No. 1, pp. 332–363

COORDINATE DIFFERENCE MATRICES*

GILLES BAECHLER†¶, FREDERIKE DÜMBGEN‡¶, GOLNOOSH ELHAMI‡¶,
MIRANDA KREKOVIĆ‡¶, AND MARTIN VETTERLI§

Abstract. In many problems such as phase retrieval, molecular biology, source localization, and sensor array calibration, one can measure vector differences between pairs of points and attempt to recover the position of these points; this class of problems is called vector geometry problems (VGPs). A closely related field studies distance geometry problems (DGPs), where only the Euclidean distance between pairs of points is available. This has been extensively studied in the literature and is often associated with Euclidean distance matrices (EDMs). Although similar to DGPs, VGPs have received little attention in the literature; our goal is to fill in this gap and introduce a framework to solve VGPs. Inspired by EDM-related approaches, we arrange the differences in what we call a coordinate difference matrix (CDM) and introduce a methodology to reconstruct a set of points from CDM entries. We first propose a reconstruction scheme in 1D and then generalize it to higher dimensions. We show that our algorithm is optimal in the least-squares sense, even when we have only access to partial measurements. In addition, we provide necessary and sufficient conditions on the number and structure of measurements needed for a successful reconstruction, as well as a comparison with EDMs. In particular we show that compared to EDMs, CDMs are simpler objects, both from an algorithmic and a theoretical point of view. Therefore, CDMs should be favored over EDMs whenever vector differences are available. In the presence of noise, we provide a statistical analysis of the reconstruction error. Finally, we apply the established knowledge to five practical problems to demonstrate the versatility of this theory and showcase the wide range of applications covered by the CDM framework.

Key words. coordinate difference matrices, Euclidean distance matrices, assigned vector geometry problem, range and angle measurements, array calibration, source localization, incomplete data, measurement uncertainty, molecular conformation, ranking

AMS subject classifications. 15A30, 15B48

DOI. 10.1137/18M123428X

1. Introduction.

Many problems in science and engineering take the following form: Given a dissimilarity measure between entities, can one find an entity embedding in which similar objects are close to each other, and dissimilar objects are far? In the field of medical imaging, for instance, the entities can be retina images, and we might want to find an embedding such that retina images of the healthy patients are grouped together, while images of patients with diabetic retinopathy can be distinguished [37, 77]. In text-based information retrieval, we try to find word embeddings such that words that are related are close [78]. In this case, the similarity could be the number of word co-occurrences in training documents. The embedding of such high dimensional entities is often estimated using variants of principal component analysis

*Received by the editors December 19, 2018; accepted for publication (in revised form) by J.M. Bardsley January 14, 2020; published electronically March 25, 2020.

https://doi.org/10.1137/18M123428X

Funding: The work of the authors was supported by the Swiss National Science Foundation grant 20FP-1 151073, "Inverse Problems Regularized by Sparsity," and grant 200021 181978/1, "SESAM - Sensing and Sampling: Theory and Algorithms."

†Google Switzerland, CH 8002 Zürich, Switzerland (baechler@google.com, http://ai.google).

‡AudioVisual Communications Laboratory (LCAV), École Polytechnique Fédérale de Lausanne, Lausanne, Vaud, 1015, Switzerland (frederike.duembgen@epfl.ch, golnoosh.elhami@epfl.ch, miranda.krekovic@epfl.ch).

§School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland (martin.vetterli@epfl.ch).

¶The authors contributed equally to this work and are listed alphabetically.

[72], linear discriminant analysis [57], or neural networks [58, 37, 77]. In the present work, we focus on point embeddings of arbitrary dimension where dissimilarities are measured by coordinate differences, which model many relevant problems.

A vast and mature literature looks at a framework similar to ours, where the dissimilarity measures are Euclidean distances between points. Such problems form the category of distance geometry problems (DGPs), of which a thorough overview is given in [53]. One broadly distinguishes between assigned DGPs (aDGPs) and unassigned DGPs (uDGPs) [26]. In aDGPs, the indices of the pairs of points from which the measured distances originate are known, so we say that the distances are labeled. In such a framework, it is a common practice to arrange the measurements in a Euclidean distance matrix (EDM), where the element at position (i, j) is the squared Euclidean distance between points i and j [20]. By exploiting their various properties, EDMs can be completed and denoised, and many algorithms for recovering admissible point embeddings exist [40, 30, 21, 65, 23].

Recently, interest has grown for vector geometry problems (VGPs), which are an extension of distance geometry to vector measurements. In vector geometry, the dissimilarity between points is given not only by their distance but also by the orientation of the edge between them. The term *vector geometry* was coined in [5] in the context of molecular conformation reconstruction. In this problem, certain measurement methods can provide the relative orientation of the atoms, induced, for example, from bond and torsion angles. DGP build-up methods, which have been widely used for molecular conformation reconstruction to date [59], do not exploit this additional information.

To establish and further develop the foundations of VGPs, and inspired by EDMs, we introduce new objects that we call coordinate difference matrices (CDMs). Similar to their Euclidean distance counterpart, CDMs contain differences between the coordinates of pairs of points. In this work, we put them in perspective with respect to EDMs in terms of properties, characteristics, and algorithms. We propose methods to recover the point embeddings from a variable number of coordinate differences, and we discuss the uniqueness and optimality of the embeddings. We provide conditions on the number and structure of measurements for the recovery to be well defined, as well as statistical analyses on the reconstruction error.

The main building block for point recovery in any dimension is the CDM constructed from 1D coordinates: given D-dimensional difference vectors between D-dimensional points, we can simply create one CDM for each coordinate and recover the coordinates of the points independently. When we are given the difference vectors of the points projected onto $K \geq D$ frame vectors, we can rely on the same idea and obtain a suboptimal but efficient approach. Interestingly, we show that this method in certain cases leads to the optimal solution in the least-squares sense, even in the presence of noise and missing measurements.

Furthermore, we illustrate the usefulness of CDMs and show that their field of application ranges from molecular conformation to ranking problems. In Figure 1 we depict six different applications that can be solved by our algorithms. Besides the already discussed molecular conformation problem (Figure 1a), we can also measure distances and angles in sensor networks (Figure 1b). In these two applications, we can project the measurements onto a standard basis. In microphone array calibration (Figure 1c), the measurements can be interpreted as coordinate differences in $K \geq D$ directions, each corresponding to the location of one far-field speaker. In phase retrieval (Figure 1d), a problem arising in X-ray tomography, we obtain unlabeled coordinate differences. While resolving the labeling problem, partial solutions

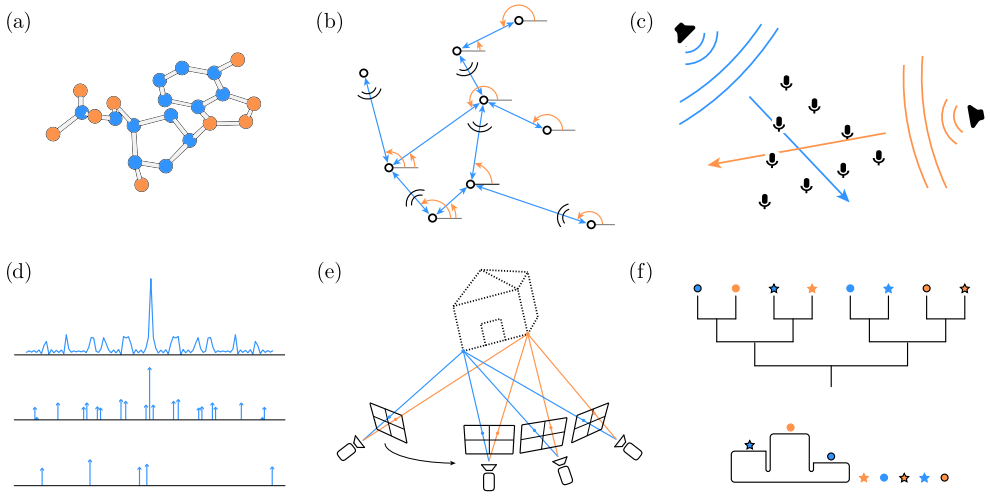


FIG. 1. *Overview of applications of CDMs:* (a) *molecular conformation,* (b) *sensor array localization,* (c) *geometric microphone calibration,* (d) *phase retrieval,* (e) *structure from motion,* (f) *ranking and rating estimation.*

can be denoised using CDMs. In affine structure from motion (Figure 1e), we recover coordinates of points from their projections onto image planes. Finally, in ranking sports teams (Figure 1f), the net scores of games can be viewed as coordinate differences from which we reconstruct relative rankings of teams [56].

To summarize, we lay down important mathematical concepts regarding VGPs. Our contributions are threefold. First, we introduce and analyze CDMs. We study their properties in section 2, provide a simple point recovery algorithm as well as a reconstruction uniqueness result in section 3, and compare the result with the classic EDM-based approach in section 5. Second, we propose two algorithms for recovering point embeddings from coordinate differences in higher dimensions in section 4: one which is optimal in the least-squares sense, and one which is computationally more efficient. We also identify cases in which these two algorithms coincide and conduct an error analysis in section 6. Third, we showcase and solve a variety of applications which can be formulated as coordinate difference problems in section 7, hoping that it will convince the reader of the practical relevance of a research topic that has received little attention so far.

2. Coordinate difference matrices. Consider N 1D points $\boldsymbol{x} = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix}^\top \in \mathbb{R}^N$. We define a *coordinate difference matrix* (CDM) $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ with entries $C_{ij} = x_i - x_j$ as the matrix that contains the pairwise coordinate differences of the points in $\boldsymbol{x}$. It can be expressed as

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

where $\mathbb{1}$ is the all-one vector.

In Table 1, we provide a nonexhaustive list of properties of CDMs that are leveraged later to reconstruct a point set from a given CDM (for derivations, see Appendix A). It is worth mentioning that the special case of equally spaced points results in the CDM being a Toeplitz matrix. These ten properties are necessary conditions for a matrix $\boldsymbol{C}$ to be a CDM. Additionally, we can derive a sufficient condition.

TABLE 1
Properties of CDMs.

	Properties	**Description**
P.1	Rank-2	$\operatorname{rank}(\boldsymbol{C}) = 2$ for $N > 1$
P.2	Triangular equality	$C_{ij} = C_{ik} + C_{kj}$
P.3	Skew-symmetry	$\boldsymbol{C} = -\boldsymbol{C}^\top$
P.4	Hollowness	$\operatorname{diag}(\boldsymbol{C}) = \mathbf{0}$
P.5	Zero-sum	$\mathbf{1}^\top \boldsymbol{C} \mathbf{1} = \mathbf{0}$
P.6	Columns as a solution set	CDM of $\boldsymbol{x} = \{C_{ij} \mid \forall i \leq N\}$ is $\boldsymbol{C}$
P.7	Row averaging	$\boldsymbol{x} + c = \frac{1}{N}\boldsymbol{C}\mathbb{1}$, where c is a constant
P.8	Translation invariance	$\boldsymbol{x}$ and $\boldsymbol{x} + c$ have the same CDM $\boldsymbol{C}$
P.9	Imaginary eigenvalues	$\Re(\lambda_i) = 0$ for $i = \{1, 2\}$
P.10	Anti-symmetric eigenvalues	$\lambda_1 = -\lambda_2$

PROPOSITION 2.1. *A matrix $\boldsymbol{C}$ is a CDM if and only if its elements satisfy the triangular equality $C_{ij} = C_{ik} + C_{kj}$ for all triples (i, j, k).*

Proof. A matrix $\boldsymbol{C}$ whose entries satisfy the triangular equality implies both hollowness and skew-symmetry: $C_{ij} = C_{ii} + C_{ij}$ ensures that $C_{ii} = 0$, while $C_{ii} = C_{ij} + C_{ji} = 0$ implies $C_{ij} = -C_{ji}$. Combining the triangular equality with the skew-symmetry, we can express all elements as $C_{ij} = C_{ik} - C_{jk}$. Without loss of generality, we define $\boldsymbol{x} = \begin{bmatrix} C_{1k} & C_{2k} & \dots & C_{Nk} \end{bmatrix}^\top$. Using this definition, we can rewrite C_{ij} as $C_{ij} = x_i - x_j$ for every i and j; therefore, $\boldsymbol{C}$ is of the form (1). ∎

An alternative way to check if the sufficient condition is satisfied is based on consistent positive reciprocal matrices [73], described below for completeness.

DEFINITION 2.2. *The matrix $\boldsymbol{R} \in \mathbb{R}^{N \times N}$ is positive reciprocal if $R_{ij} > 0$ and $R_{ij} = R_{ji}^{-1}$ for any $i, j = 1, \dots, N$. If $R_{ik} = R_{ij} R_{jk}$, it is said to be consistent.*

We can transform a CDM $\boldsymbol{C}$ into a reciprocal matrix $\boldsymbol{R}$ with elementwise exponentiation, $\boldsymbol{R} = \exp(\boldsymbol{C})$. The consistency property is then the natural extension of the triangular equality to reciprocal matrices, and it provides an alternative way of testing if a matrix is a CDM.

It is known that a reciprocal matrix is consistent if and only if it has $N - 1$ zero eigenvalues and one eigenvalue equal to N [73]. As a consequence, we can verify whether a matrix is a CDM by examining the eigenvalues of its corresponding reciprocal matrix.

3. Recovering point embeddings from CDMs. In general, CDMs can be incomplete. To take into account missing entries, we introduce a symmetric weight matrix $\boldsymbol{W}$ with nonnegative entries, where $W_{ij} = 0$ indicates that the entry (i, j) is missing. Moreover, $W_{ij} > 0$ denotes the importance of each difference C_{ij}, and it can encompass, for example, multiple measurements of the same difference or the certainty about each measurement. To simplify the notation, we adopt the convention that $W_{ii} = 0$ for all i. The idea of assigning different priorities to measurements was first introduced in [35].

In addition to being missing, the measured differences can also be noisy; in that regard, we introduce the noise matrix $\boldsymbol{Z}$, whose entries are independent noise realizations. We define an incomplete and noisy CDM as

$$\widetilde{\boldsymbol{C}} = (\boldsymbol{C} + \boldsymbol{Z}) \circ \boldsymbol{W}. \tag{2}$$

In case we have multiple measurements between the points x_i and x_j, the elements $\widetilde{C}_{ij}$ and Z_{ij} are the weighted average of the measurements and the noise realizations, respectively.

The inverse problem that arises naturally from (1) and (2) is formalized as follows.

PROBLEM 1. *Given a set of noisy* 1*D differences* $\widetilde{C}_{ij}$ *for some* i, j*, recover the set of points* $\{x_i\}_{i=1}^N$ *that generated them.*

3.1. Reconstruction algorithm. To solve Problem 1, we propose estimating the points from a measured subset of their pairwise differences as

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}} \left\| \boldsymbol{W} \circ (\boldsymbol{x}\mathbb{1}^\top - \mathbb{1}\boldsymbol{x}^\top - \widetilde{\boldsymbol{C}}) \right\|_F^2. \tag{3}$$

In Appendix B we prove that $f(\boldsymbol{x})$ is convex. Thus, we can find the optimal solution by setting the first derivative of $f(\boldsymbol{x})$ to zero:

$$x_i = \frac{1}{\Lambda_{ii}} \sum_{j=1}^{N} (x_j + \widetilde{C}_{ij}) W_{ij}, \tag{4}$$

where

$$\Lambda_{ij} = \begin{cases} \sum_{k=1}^N W_{ik}, & i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

We can rewrite this result in matrix form $\boldsymbol{A}\boldsymbol{x} = \widetilde{\boldsymbol{v}}$, where $\boldsymbol{A} = \boldsymbol{\Lambda} - \boldsymbol{W}$ and $\widetilde{\boldsymbol{v}} = (\widetilde{\boldsymbol{C}} \circ \boldsymbol{W})\mathbb{1}$.

The matrix $\boldsymbol{A}$ has a particular structure and belongs to the class of so-called *M-matrices* [68, 46]. We study its invertibility in the following section.

3.2. Invertibility of $\boldsymbol{A}$. Let us define the weighted graph $G = (\boldsymbol{x}, \boldsymbol{W})$, where vertices are represented by the points $\boldsymbol{x}$ and their connecting edges are given by $\boldsymbol{W}$. In general, graphs provide an interesting alternative representation for CDMs, but in the scope of this paper we solely leverage them to study the invertibility of $\boldsymbol{A}$.

Observe that $\boldsymbol{A}$ is the Laplacian matrix of its corresponding graph G, as it is the difference of the degree matrix $\boldsymbol{\Lambda}$ and the adjacency matrix $\boldsymbol{W}$. Hence, $\operatorname{rank}(\boldsymbol{A})$ is at most $N - 1$ [41], and it is not invertible.

This result is not surprising; indeed, Property P.8 states that we can recover the original points only up to a translation. In order to anchor the translation, we can arbitrarily fix one point of the embedding; in this paper we set x_1 to zero, without loss of generality. To that end, we remove the first entry of $\boldsymbol{x}$ and $\widetilde{\boldsymbol{v}}$ and denote the new vectors by $\boldsymbol{x}'$ and $\widetilde{\boldsymbol{v}}'$. Similarly, we remove the first row and column of $\boldsymbol{A}$ to get $\boldsymbol{A}'$, and we define the matrices $\boldsymbol{\Lambda}'$, $\boldsymbol{W}'$, and $\widetilde{\boldsymbol{C}}'$ analogously. This yields the following linear system:

$$\boldsymbol{A}'\boldsymbol{x}' = \widetilde{\boldsymbol{v}}'. \tag{6}$$

3.2.1. Complete CDM. In the special case where we measure all pairwise differences and assign them the same weight—we call it a *complete CDM*—an analytic solution for $(\boldsymbol{A}')^{-1}$ exists. Indeed, with $\boldsymbol{\Lambda}' = (N-1)\boldsymbol{I}$ and $\boldsymbol{W}' = \mathbb{1}\mathbb{1}^\top - \boldsymbol{I}$, we obtain

$$(\boldsymbol{A}')^{-1} = \left(\boldsymbol{I} - \frac{\mathbb{1}\mathbb{1}^\top - \boldsymbol{I}}{N-1} \right)^{-1} \frac{1}{N-1} = \frac{\boldsymbol{I} + \mathbb{1}\mathbb{1}^\top}{N}. \tag{7}$$

This can be easily verified by direct computation. To recover $\boldsymbol{x}$ from a complete CDM, we substitute (7) into (6):

$$\widehat{\boldsymbol{x}}' = \left(\boldsymbol{A}'\right)^{-1}\widetilde{\boldsymbol{v}}' = \frac{\boldsymbol{I} + \mathbb{1}\mathbb{1}^\top}{N}\widetilde{\boldsymbol{C}}'\mathbb{1} = \frac{1}{N}\widetilde{\boldsymbol{C}}'\mathbb{1} + c\mathbb{1}, \tag{8}$$

where $c = \frac{1}{N}\sum_{ij}\widetilde{C}'_{ij}$ is a constant that only translates the solution; this result confirms Property P.7. We conclude that the optimal point recovery in the complete case corresponds to a simple average of the rows of a CDM.

3.2.2. Incomplete and weighted CDM. In the following, we study the invertibility of $\boldsymbol{A}'$ when some entries of the CDM are missing or/and they have different assigned weights.

We say that the CDM $\boldsymbol{C}$ is *connected* if and only if its underlying graph G is connected, or, in other words, if for all indices $i \neq j$ there is a path of indices $i_1, i_2, \ldots, i_m$ such that $W_{ii_1} \neq 0, W_{i_1i_2} \neq 0, \ldots, W_{i_{m-1}i_m} \neq 0, W_{i_mj} \neq 0$. According to Kirchhoff's matrix tree theorem [13], the number of spanning trees of G is given by $\kappa(G) = \det\left(\boldsymbol{A}'\right)$. Obviously, G is connected if and only if $\kappa(G) \neq 0$, or, equivalently, if and only if $\boldsymbol{A}'$ is nonsingular.

It is not hard to see that $\boldsymbol{A}'$ is weakly diagonally dominant, $|A'_{ii}| \geq \sum_{j=1,j\neq i}^{N-1}|A'_{ij}|$, as $|A'_{ii}| = \sum_{j=1}^{N} W_{i+1,j}$, $|A'_{ij}| = W_{i+1,j+1}$, and $\sum_{j=1}^{N} W_{i+1,j} \geq \sum_{j=2}^{N} W_{i+1,j}$ for every row i. When $\boldsymbol{C}$ is connected, then $\boldsymbol{A}'$ is reducible, and for at least one row i we have a strict inequality $\sum_{j=1}^{N} W_{ij} > \sum_{j=2}^{N} W_{ij}$; hence, $\boldsymbol{A}'$ is irreducibly diagonally dominant. This proves the same result as above, that $\boldsymbol{A}'$ is nonsingular [69], but it provides two additional insights: it shows that we can solve (6) in nearly linear time in N [74], and it enables a simple derivation for the inverse of $\boldsymbol{A}'$; for a connected matrix $\boldsymbol{C}$, it is given by Neumann series

$$(\boldsymbol{A}')^{-1} = \left(\sum_{k=0}^{\infty}((\boldsymbol{\Lambda}')^{-1}\boldsymbol{W}')^k\right)(\boldsymbol{\Lambda}')^{-1}. \tag{9}$$

The proof follows directly from the Gershgorin circle theorem [33], which proves that the eigenvalues of $(\boldsymbol{\Lambda}')^{-1}\boldsymbol{W}'$ are inside the unit circle, and the Neumann series theorem [44]. In Appendix C we confirm that the infinite sum (9) converges to (7) in the case of complete measurements.

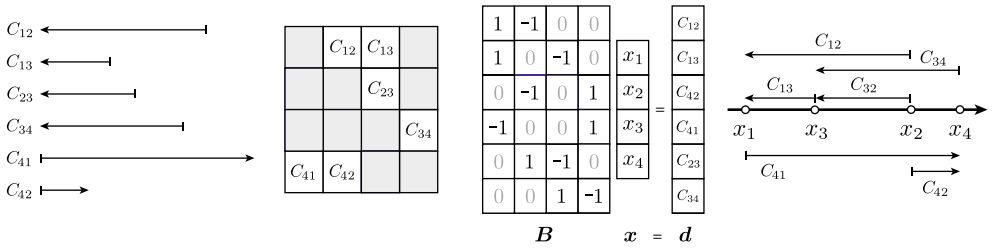
The invertibility of $\boldsymbol{A}'$ enables us to reconstruct the points $\boldsymbol{x}$ from (6) as

$$\boldsymbol{x} = \begin{bmatrix} 0 \\ (\boldsymbol{A}')^{-1}\widetilde{\boldsymbol{v}}' \end{bmatrix}. \tag{10}$$

Clearly, we cannot apply (10) when $\boldsymbol{C}$ is not connected. What we can do in such a case is recover the points by invoking (10) within each *connected component* of $\boldsymbol{C}$. As the recovered connected components can shift independently, we have an infinite number of solutions.

4. Generalization to higher dimensions. In practice, we can often measure the vectors between multidimensional points projected onto some measurement directions; examples of such setups are given in subsection 7.3 and subsection 7.4. This naturally motivates a generalization of CDMs to higher dimensions.

4.1. Vector form in 1D. Before discussing the multidimensional case, we introduce an alternative way to represent a CDM by arranging its entries in a vector



(a) measurements (b) incomplete CDM (c) vectorized form (d) reconstructed points

FIG. 2. *An instance of Problem* 1. (a) *Assume that we can measure a subset of noisy* 1*D differences between the points.* (b) *We arrange them in a CDM and either use it directly, or* (c) *use its vectorized form to* (d) *reconstruct the points that give rise to measurements.*

$\boldsymbol{d} \in \mathbb{R}^M$, where M denotes the total number of measurements. Then, $d_m = x_i - x_j$ are the pairwise differences between the points, and m is indexing the observed entries of the flattened CDM. To assign the difference d_m to the pair of points x_i and x_j, we introduce a sparse measurement matrix $\boldsymbol{B} \in \mathbb{R}^{M \times N}$ with $B_{mi} = 1$, $B_{mj} = -1$, and 0 otherwise. Using this notation, we can write

$$\boldsymbol{B}\boldsymbol{x} = \boldsymbol{d}. \tag{11}$$

In the case of noisy measurements, we define $\widetilde{\boldsymbol{d}} = \boldsymbol{d} + \boldsymbol{z}$, where $\boldsymbol{z}$ contains independent noise realizations. In the case of multiple measurements, $\boldsymbol{B}$ has identical rows for every repeated measurement, and $\widetilde{\boldsymbol{d}}$ contains all of their realizations. An instance of Problem 1, both in matrix and vectorized form, is illustrated in Figure 2.

Recall that in section 3 we defined $\boldsymbol{W}$ to be a nonnegative matrix whose entries are the weights of each difference measurement. To be aligned with such a definition, we should allow not only multiple measurements of each difference (identical rows in $\boldsymbol{B}$) but also different weights for these measurements. To do so, we can simply scale rows of $\boldsymbol{B}$ and $\widetilde{\boldsymbol{d}}$ with the same factors. To keep the notation clean, in the rest of the paper we assume that $\boldsymbol{B}$ and $\widetilde{\boldsymbol{d}}$ encompass these weights.

We can estimate $\boldsymbol{x}$ by solving the normal equations, $\boldsymbol{B}^\top \boldsymbol{B}\boldsymbol{x} = \boldsymbol{B}^\top \widetilde{\boldsymbol{d}}$. Due to translation ambiguity, a system is noninvertible. We resolve it by removing the first column of $\boldsymbol{B}$ to get $\boldsymbol{B}'$, which corresponds to setting $x_1 = 0$. This brings us to our well-studied problem (6), where $\boldsymbol{A}' = \boldsymbol{B}'^\top \boldsymbol{B}'$, $\widetilde{\boldsymbol{v}}' = \boldsymbol{B}'^\top \widetilde{\boldsymbol{d}}$, and the points are reconstructed as

$$\widehat{\boldsymbol{x}}' = (\boldsymbol{B}')^\dagger \widetilde{\boldsymbol{d}} = ((\boldsymbol{B}')^\top \boldsymbol{B}')^{-1} (\boldsymbol{B}')^\top \widetilde{\boldsymbol{d}}. \tag{12}$$

Such a problem definition is not novel; it appears in the broad literature on statistical ranking from pairwise comparisons. For instance, it is used in [56] to rank a collection of sport teams based on their scores, and in [62] to design tournaments that maximally improve the informativeness of a ranking for a given number of future comparisons. The reason we introduce it here is twofold: (a) As we are the first to show the connection of (11) with CDMs, it may be useful to recast the existing problems such as [56, 62] that rely on (11) to the CDM framework and take advantage of the devised properties, bounds on the reconstruction error, more efficient implementations, and connections to other applications. (b) The vector form proves to be beneficial in the error analysis.

4.2. Vector form in higher dimensions. We extend points and their pairwise differences to D dimensions and consider $K \geq D$ frame vectors $\{\boldsymbol{\varphi}_k\}_{k=1}^K$ [79]. Then, we formulate the generalization of Problem 1 and expand the formulation introduced in subsection 4.1.

PROBLEM 2. *Given a subset of noisy D-dimensional coordinatewise differences observed in the frame $\{\boldsymbol{\varphi}_k\}_{k=1}^K$, recover the set of points that generated them.*

We assume that we measure M_k differences in each frame direction k, $k = 1, \ldots, K$, and construct the measurement matrix $\boldsymbol{B}_k \in \mathbb{R}^{M_k \times N}$ and the vector of differences $\widetilde{\boldsymbol{d}}_k \in \mathbb{R}^{M_k}$ for every k, analogously to $\boldsymbol{B}$ and $\widetilde{\boldsymbol{d}}$ in (11). The total number of differences is denoted by $M = \sum_{k=1}^K M_k$. For every frame vector $\boldsymbol{\varphi}_k$, we create the matrix $\boldsymbol{\Phi}_k \in \mathbb{R}^{N \times ND}$, such that

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{\varphi}_k^\top & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varphi}_k^\top & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \boldsymbol{\varphi}_k^\top \end{bmatrix}. \tag{13}$$

Then, we can formalize Problem 2 as follows:

$$\begin{bmatrix} \boldsymbol{B}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{B}_K \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \boldsymbol{\Phi}_2 \\ \vdots \\ \boldsymbol{\Phi}_K \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{d}}_1 \\ \widetilde{\boldsymbol{d}}_2 \\ \vdots \\ \widetilde{\boldsymbol{d}}_K \end{bmatrix}, \tag{14}$$

or, in matrix form,

$$\boldsymbol{B}\boldsymbol{\Phi}\boldsymbol{x} = \widetilde{\boldsymbol{d}}, \tag{15}$$

where $\widetilde{\boldsymbol{d}} \in \mathbb{R}^M$ contains all M measured K-dimensional differences in all frames arranged in one vector, $\boldsymbol{\Phi} \in \mathbb{R}^{NK \times ND}$ contains the matrices $\boldsymbol{\Phi}_k$ stacked in a tall matrix, and $\boldsymbol{x} \in \mathbb{R}^{DN}$ is the vector of all N D-dimensional points $\boldsymbol{x}_n$ that we want to recover, $n = 1, \ldots, N$. Figure 3 presents a simple example of three points in a 2D space to illustrate the notation and clarify the above expressions.

4.3. Optimal solution. Analogously to (11), the system (15) is noninvertible, so we fix the first coordinate of every dimension to zero. This corresponds to removing every Kth column of $\boldsymbol{B}$, as well as every Kth row and the first D columns of $\boldsymbol{\Phi}$, resulting in the new matrices $\boldsymbol{B}' \in \mathbb{R}^{M \times (N-1)K}$ and $\boldsymbol{\Phi}' \in R^{(N-1)K \times (N-1)D}$, respectively. Then, we can reconstruct the canonical coordinates of the points as

$$\widehat{\boldsymbol{x}}' = (\boldsymbol{B}'\boldsymbol{\Phi}')^\dagger \widetilde{\boldsymbol{d}}. \tag{16}$$

This approach is optimal in the least-squares sense.

4.4. Splitting algorithm. To reduce the complexity of the solution, we propose and study an alternative approach that decomposes Problem 2 into many instances of Problem 1 of smaller dimension. We divide the multidimensional problem into K 1D CDM recovery problems and estimate the points from their differences separately in each frame direction. Therefore, we first recover the expansion coefficients

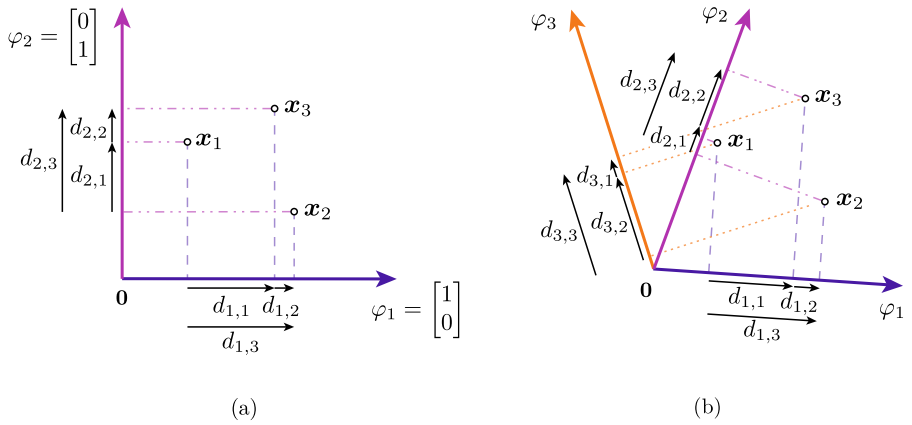


FIG. 3. *An example of the point recovery in* 2*D.* (a) *The simple case of the Cartesian coordinate system. We use* $d_{k,m}$ *for* $k = 1, 2$ *and* $m = 1, 2, 3$ *to indicate the measured pairwise differences between the Cartesian coordinates of the points* $\{\boldsymbol{x}_n\}_{n=1}^3$. (b) *A frame with* $K = 3$ *vectors. We assume we measure* $d_{k,m}$ *for* $k = 1, 2, 3$ *and* $m = 1, 2, 3$, *and we want to recover* $\{\boldsymbol{x}_n\}_{n=1}^3$.

of the points for each frame vector, given by $(\boldsymbol{B}')^\dagger \widetilde{\boldsymbol{d}}$, and then we find their canonical coordinates by a change of basis:

$$\widehat{\boldsymbol{x}}' = (\boldsymbol{\Phi}')^\dagger (\boldsymbol{B}')^\dagger \widetilde{\boldsymbol{d}}. \tag{17}$$

Note that in a more efficient implementation, (17) is solved in two steps. The computation of $(\boldsymbol{B}')^\dagger \widetilde{\boldsymbol{d}}$ in the first step is divided into K independent problems, each giving an estimate of $\boldsymbol{f}_k \in \mathbb{R}^K$, $\boldsymbol{f}_k = (\boldsymbol{B}'_k)^\dagger \widetilde{\boldsymbol{d}}_k$. Then, the canonical coordinates of the estimated points $\widehat{\boldsymbol{x}}'$ are in the columns of $(\boldsymbol{\Phi}'_0)^\dagger \boldsymbol{F}$, where $\boldsymbol{F} \in \mathbb{R}^{K \times N}$ contains vectors $\boldsymbol{f}_k$ in its columns. Besides the reduction in the size of the problem, the main benefit of this formulation is that we can take advantage of the structure of $\boldsymbol{B}'_k$ to solve the problem faster: indeed, as mentioned in subsection 3.2.2, $(\boldsymbol{B}'_k)^\top \boldsymbol{B}'_k$ is irreducibly diagonally dominant, which enables us to invert it in nearly linear time. The splitting formulation also allows for the method to be easily parallelized.

4.5. Uniqueness and number of solutions. Given a set of coordinate differences, there is an infinite number of valid point sets which are generated by translating the original point set. Here, we refer to a problem with a *nonunique* solution when at least two different point sets that are not translated versions of each other are both valid solutions.

For the multidimensional point reconstruction to be possible, we need to extend the connectivity requirement. At the coordinate level, every point needs to be connected with the others by at least D frame measurements. Since we have N points and therefore at least $N - 1$ connections, the minimum number of measurements is $D(N - 1)$. When using the splitting algorithm, this requirement is more restrictive: we require that the CDM corresponding to every frame vector is connected, and hence, we need at least $(N - 1)K$ measurements. We summarize the results on the uniqueness and number of solutions in the following proposition.

PROPOSITION 4.1. *We refer to the ensemble of all translated reconstructions as one single solution. Then, the CDM problem in* 1*D can have either one or an infinite number of solutions. It has one solution if and only if the CDM is connected. The*

TABLE 2
Comparison of CDMs $\boldsymbol{C}$ and EDMs $\boldsymbol{E}$.

Properties	**CDM**	**EDM**
Rank	$\operatorname{rank}(\boldsymbol{C}) = 2$	$\operatorname{rank}(\boldsymbol{E}) = D + 2$
Triangular (in)equality	$C_{ij} = C_{ik} + C_{kj}$	$\sqrt{E_{ij}} \leq \sqrt{E_{ik}} + \sqrt{E_{kj}}$
Symmetry	$\boldsymbol{C} = -\boldsymbol{C}^\top$	$\boldsymbol{E} = \boldsymbol{E}^\top$
Hollowness	$\operatorname{diag}(\boldsymbol{C}) = \boldsymbol{0}$	$\operatorname{diag}(\boldsymbol{E}) = \boldsymbol{0}$
Invariance to	translations	all rigid motions

CDM problem in D dimensions can have either one solution or an infinite number of solutions. If the CDMs of at least D independent frame vectors are connected, a unique solution exists.

Note that the uniqueness condition for $D > 1$ is sufficient, but not necessary. The study of the exact number of solutions is tightly connected with the (global) graph rigidity problem, and, to the best of our knowledge, no trivial solution exists at this point.

5. Comparison of CDMs with EDMs. In this section, we relate the introduced framework of CDMs with the more mature field of EDMs. The EDM and the CDM theories arise in similar applications, but one can be more useful than the other depending on the measured information. CDMs can be used to solve assigned VGPs (aVGPs), while EDMs play an important role in aDGPs, which are harder by nature (the difference vectors are *collapsed* into one dimension). Oftentimes, EDM is the tool of choice simply because the coordinate information is not available by design; however, due to its popularity, the EDM framework is sometimes used where CDMs would be more adequate, precise, and efficient. The molecular conformation application is one such example: even though coordinate information is available through different angle measurements, more focus was given to distance-based methods to this date [5]. We discuss this problem in more detail in subsection 7.5.

We compare below CDMs and EDMs with respect to (a) matrix properties, (b) minimal number of observed entries required for the reconstruction, (c) uniqueness of the reconstruction, and (d) applications other than point recovery.

There are some obvious algebraic differences between CDMs and EDMs, summarized in Table 2. The rank of both matrices is independent of the number of points: for CDMs constructed for 1D points, the rank is equal to 2, while for EDMs of D-dimensional points it is equal to $D + 2$. Furthermore, the entries of CDMs satisfy the triangular equality, while the elements of EDMs respect the triangular inequality. Moreover, CDMs are skew-symmetric, while EDMs are symmetric. Finally, when transitioning from a point set to a CDM, information about the absolute translation of the points is irremediably lost. When estimating points from EDMs, in addition to translation, we also lose the information about their rotation and reflection.

Let us now consider the recovery of points in 1D. Assuming a connected CDM, we require at least one measurement per point, that is, $N - 1$ coordinate differences to recover N points. To visualize this, we can think in terms of a build-up algorithm and fix the first point at 0. Then we can iteratively reconstruct the remaining points in one pass from the coordinate differences. The entries of EDMs in 1D are simply the squared values of the entries of CDMs, but this small difference makes the recovery problem significantly harder. In addition to fixing the first point at 0, we also need to pick the sign of the second point to fix the reflection. Then we can iteratively

build a solution, observing that for every newly added point we have two possibilities. To identify the correct one, we need at least one additional measurement from a previously recovered point.

Moving to point recovery in higher dimensions, the measurements in CDMs are given as K-dimensional coordinate differences between D-dimensional points, where $K \geq D$. The minimal number of measurements is achieved for $N-1$ connected measurements per each dimension, resulting in the total number of $D(N-1)$ measurements. For noiseless EDMs, there is no exact formula that describes the minimum number of measurements, but we can lower bound it by a counting argument. We have DN unknowns (N D-dimensional points) to recover, but these points are only recovered up to rigid motions, which encompass D degrees of freedom for translations and $D(D-1)/2$ degrees of freedom for orthogonal transformations. We conclude that the number of measurements is lower bounded by $DN-(D+1)D/2$. Moreover, to ensure that every point is rigidly connected to the others, at least $D+1$ measurements are needed for every point. As the dimension of the space in most applications is typically 2 or 3, the contribution of the second term is negligible, and the reconstruction of points from CDMs and noiseless EDMs requires the same number of measurements. When distances are noisy, there is no clear expression for the minimal number of measurements in the EDM reconstruction problem, and no existing algorithm guarantees an optimal solution [23]. On the other hand, our proposed algorithm within the CDM framework has an optimal closed-form solution even with imperfect measurements.

This leads to another important difference between CDMs and EDMs. As no algorithm guarantees an optimal solution for EDM reconstruction with incomplete matrices, many methods in the EDM literature split the problem into two independent steps: matrix completion and denoising, followed by point recovery. The goal of the matrix completion and denoising is to determine the closest EDM for a given incomplete matrix. Having the complete EDM, the point recovery is then obtained via a simple SVD. When using CDMs, the completion and denoising steps are not required because one can directly apply the point recovery algorithm on incomplete CDMs, which implicitly denoises and completes CDMs.

Lastly, although CDMs and EDMs are both designed to solve assigned distance problems, they prove to be useful as labeling and denoising tools in various applications. As an example, the rank property of EDMs has been used to recover distance labels in room geometry reconstruction from echoes [24]. When we measure unlabeled coordinate differences instead of unlabeled distances, CDMs could be used in the same way. Concretely, one could iterate over possible permutations and check if the resulting matrix satisfies Proposition 2.1, the sufficient condition of CDMs. Alternatively, EDMs and CDMs can be used in unassigned geometry problems to denoise a partial solution set during any iterative point recovery algorithm, as shown in subsection 7.2.

6. Reconstruction error. In this section, we analytically compute the expected value and variance of the estimation error of the points. For the multidimensional case, we derive the gap in the reconstruction accuracy between the optimal solution and the splitting algorithm. We perform numerical simulations to validate the theoretical analysis and illustrate the dependence of the estimation error on the amount of noise, the number of missing measurements, and the number of frame vectors.

6.1. 1D setup. We can rewrite (12) as $\widehat{\boldsymbol{x}}' = \boldsymbol{x}' + (\boldsymbol{B}')^{\dagger}\boldsymbol{z}$. As $\widehat{\boldsymbol{x}}'$ and $\boldsymbol{x}'$ contain $N-1$ points, we have to prepend the removed leading zero to $\widehat{\boldsymbol{x}}'$ and $\boldsymbol{x}'$ and align them before computing the estimation error. More precisely, we set their centroids to the origin, introduce the centering matrix $\boldsymbol{J} = \boldsymbol{I} - \frac{1}{N}\mathbb{1}\mathbb{1}^{\top} \in \mathbb{R}^{N\times N}$, and obtain

$\boldsymbol{J}' \in \mathbb{R}^{N\times N-1}$ from $\boldsymbol{J}$ by removing its first column. Then, we can compute the centered point sets as $\widehat{\boldsymbol{x}}_c = \boldsymbol{J}'\widehat{\boldsymbol{x}}'$ and $\boldsymbol{x}_c = \boldsymbol{J}'\boldsymbol{x}'$. Their difference is the desired estimation error vector, given as $\boldsymbol{e}_c = \boldsymbol{J}'(\widehat{\boldsymbol{x}}'_c - \boldsymbol{x}'_c)$. Thus,

$$\boldsymbol{e}_c \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{e}_c}), \text{ with } \boldsymbol{\Sigma}_{\boldsymbol{e}_c} = \sigma^2 \boldsymbol{J}' \left((\boldsymbol{B}')^\top \boldsymbol{B}'\right)^{-1} (\boldsymbol{J}')^\top. \tag{18}$$

It follows that the reconstructed points $\widehat{\boldsymbol{x}}_c$ are also normally distributed with mean $\boldsymbol{x}_c$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{e}_c}$.

We define the estimation error ϵ as the mean squared error (MSE) between $\widehat{\boldsymbol{x}}_c$ and $\boldsymbol{x}_c$, $\epsilon = \frac{1}{N}\|\boldsymbol{e}_c\|^2$, and we can find its expected value from

$$\mathbb{E}[\epsilon] = \frac{\sigma^2}{N} \operatorname{tr}\left(\boldsymbol{J}'' \left((\boldsymbol{B}')^\top \boldsymbol{B}'\right)^{-1}\right), \tag{19}$$

where we leverage the cyclic invariance of the trace and use $\boldsymbol{J}''$ to denote $\boldsymbol{J}$ without the first column and row, $\boldsymbol{J}'' = (\boldsymbol{J}')^\top \boldsymbol{J}'$.

This proves that the expected value of the error depends on the noise level σ^2 and the structure of measurements. To better understand the latter, we further rewrite (19) as

$$\begin{aligned}
\mathbb{E}[\epsilon] &= \frac{\sigma^2}{N}\left[\operatorname{tr}\left((\boldsymbol{A}')^{-1}\right) - \frac{1}{N}\mathbb{1}^\top (\boldsymbol{A}')^{-1}\mathbb{1}\right] \\
&= \frac{\sigma^2}{N}\sum_{k=0}^{\infty}\left[\operatorname{tr}\left(\left(\boldsymbol{W}'(\boldsymbol{\Lambda}')^{-1}\right)^k (\boldsymbol{\Lambda}')^{-1}\right) - \frac{1}{N}\mathbb{1}^\top \left(\boldsymbol{W}'(\boldsymbol{\Lambda}')^{-1}\right)^k (\boldsymbol{\Lambda}')^{-1}\mathbb{1}\right] \\
&\stackrel{\text{def}}{=} \frac{\sigma^2}{N}\sum_{k=0}^{\infty} f_k(\boldsymbol{W}', \boldsymbol{\Lambda}'),
\end{aligned} \tag{20}$$

where $(\boldsymbol{A}')^{-1}$ is from (9).

The expression $f_k(\boldsymbol{W}', \boldsymbol{\Lambda}')$ simplifies for $k = 0$ and $k = 1$ to $\frac{N-1}{N}\sum_{i=1}^{N-1} {\Lambda'_i}^{-1}$ and $-\frac{1}{N}\sum_{i,j=1}^{N-1} W'_{ij}{\Lambda'_i}^{-2}$, respectively.[1] For $k \geq 2$, we have

$$f_k(\boldsymbol{W}', \boldsymbol{\Lambda}') = \sum_{i,j=1}^{N-1} \frac{1}{\Lambda'_i} \sum_{m_1,\ldots,m_{k-1}} \frac{\left(\prod_{\ell=0}^{k-2} W'_{m_\ell, m_{\ell+1}}\right)\left(NW'_{i,m_{k-1}} - (N-1)W'_{m_{k-1},j}\right)}{\prod_{\ell=0}^{k-1}\Lambda'_{m_\ell}}, \tag{21}$$

where the second sum is over all $k-1$-tuples $(m_1, \ldots, m_{k-1})$ with $1 \leq m_\ell \leq N-1$ for $\ell = 1, \ldots, k-1$, and $m_0 = i$.

In what follows, we use (21) to prove that the smallest error is achieved for the uniform distribution of measurements. Let us consider any $\boldsymbol{W}^a$ with the corresponding $\boldsymbol{\Lambda}^a$ defined by (5). From the first row, we take two weights W^a_{1j} and W^a_{1k} such that $W^a_{1j} > W^a_{1k}$, and we choose Δ such that $0 < \Delta \leq (W^a_{1j} - W^a_{1k})/2$.

Additionally, let us define the matrix $\boldsymbol{W}^b$ such that it is equal to $\boldsymbol{W}^a$ except for $W^b_{1j} = W^b_{j1} = W^a_{1j} - \Delta$ and $W^b_{1k} = W^b_{k1} = W^a_{1k} + \Delta$. Using (5), we also associate the matrix $\boldsymbol{\Lambda}^b$ to $\boldsymbol{W}^b$. Furthermore, let us construct $\boldsymbol{W}^{a\prime}$ and $\boldsymbol{W}^{b\prime}$ from $\boldsymbol{W}^a$ and $\boldsymbol{W}^b$ by removing their first column and row.

Note that $\boldsymbol{W}^b$ brings the measurements closer to being uniformly distributed. Our goal is to show that $\boldsymbol{W}^b$ leads to a lower estimation error than $\boldsymbol{W}^a$. To see this,

[1] To ease the notation, we use Λ'_i instead of Λ'_{ii} in this section.

observe that, on the one hand, $\boldsymbol{W}^{a\prime} = \boldsymbol{W}^{b\prime}$, but, on the other hand, $\Lambda_j^b = \Lambda_j^a - \Delta$, $\Lambda_k^b = \Lambda_k^a + \Delta$, and $\Lambda_i^b = \Lambda_i^a$ for $i \neq j$, $i \neq k$. As a consequence, since the numerators in (21) depend only on the weights, they are identical for both $\boldsymbol{W}^{a\prime}$ and $\boldsymbol{W}^{b\prime}$, and only their denominators differ. From (21) we can therefore compute the difference of $f_k(\boldsymbol{W}^{a\prime}, \boldsymbol{\Lambda}^{a\prime})$ and $f_k(\boldsymbol{W}^{b\prime}, \boldsymbol{\Lambda}^{b\prime})$ as

$$
\begin{aligned}
&f_k(\boldsymbol{W}^{a\prime}, \boldsymbol{\Lambda}^{a\prime}) - f_k(\boldsymbol{W}^{b\prime}, \boldsymbol{\Lambda}^{b\prime}) \\
&\qquad = \sum_{i,j=1}^{N-1} \sum_{m_1,\ldots,m_{k-1}} h(\boldsymbol{W}') \left(\frac{1}{\Lambda_i^{a\prime} \prod_{\ell=0}^{k-1} \Lambda_{m_\ell}^{a\ \prime}} - \frac{1}{\Lambda_i^{b\prime} \prod_{\ell=0}^{k-1} \Lambda_{m_\ell}^{b\ \prime}} \right),
\end{aligned} \tag{22}
$$

where $h(\cdot)$ is used to shorten the expressions of the numerators in (21).

As $0 \leq \Lambda_j^b - \Lambda_k^b < \Lambda_j^a - \Lambda_k^a$ and $\sum_{i=1}^N \Lambda_i^a = \sum_{i=1}^N \Lambda_i^b$, it is not hard to show that (22) is positive for every k. In other words, we have shown that if we rearrange the entries of the first row in $\boldsymbol{W}$ such that they are more uniform and that the variance of $\boldsymbol{\Lambda}$ is smaller, then the estimation error decreases. With a derivation analogous to the above, we can show that the contrary is also true; by rearranging the entries of the first row of $\boldsymbol{W}$ such that the variance in $\boldsymbol{W}$ and $\boldsymbol{\Lambda}$ increases, then the estimation error also increases. As we can arbitrarily swap rows in $\boldsymbol{W}$, as long as we swap their corresponding columns, the entries in the first row of $\boldsymbol{W}$ can be associated to any point $i = 1, \ldots, N$. Therefore, the above proof is general and valid for any i. In fact, we can show that by maximizing expression (22) over the entries in $\boldsymbol{\Lambda}^{b\prime}$, the global maximum is achieved when $\Lambda_1^{b\prime} = \cdots = \Lambda_{N-1}^{b\ \prime}$; these values are equal to the sum of all entries in $\boldsymbol{W}^a$ divided by $N - 1$.

Lastly, we provide expressions for the best and the worst arrangements of measurements with the total sum of the weights fixed to $L(N^2 - N)$, $L \in \mathbb{N}$. As shown above, the smallest error is achieved when $W_{ij} = L$ for every (i, j), $i \neq j$,

$$
\epsilon_{min} = \frac{\sigma^2}{N} \sum_{k=0}^{\infty} \frac{N \operatorname{tr}(\boldsymbol{W}^k) - \mathbb{1}^\top \boldsymbol{W}^k \mathbb{1}}{N(L(N-1))^{k+1}} = \frac{\sigma^2}{N} \frac{L(N-1)}{L(N-1)+1}. \tag{23}
$$

On the other extreme, the largest error occurs when there exists p for which $W_{pj} = W_{jp} = LN/2$ for every j, $j \neq p$, while $W_{ij} = 0$ for every other entry (i, j), $i \neq p$,

$$
\epsilon_{max} = \frac{\sigma^2}{N} \operatorname{tr}((\boldsymbol{\Lambda}')^{-1}) - \frac{1}{N} \mathbb{1}^\top (\boldsymbol{\Lambda}')^{-1} \mathbb{1} = \sigma^2 \frac{2(N-1)^2}{LN^3}. \tag{24}
$$

Simulation results. The dependence of the error on the noise level and number of measurements is illustrated in Figure 4. The number of measured differences spreads from the minimum required for reconstruction, $M = N - 1$, to the complete case, $M = N(N-1)/2$. We compute the error ϵ for multiple realizations of the matrix $\boldsymbol{B}$, i.e., different connectivities between the points, and take their average to estimate the expected value of ϵ defined in (19). It is clear that the error increases with the amount of noise and number of missing entries.

6.2. Multidimensional setup. We extend the statistical analysis from subsection 6.1 to the multidimensional setting and provide a closed-form expression for the difference in the estimation errors of the optimal and the splitting algorithms.

We assume that we add independent Gaussian noise to all the differences, such that $\widetilde{\boldsymbol{d}} \sim \mathcal{N}(\boldsymbol{d}, \sigma^2 \boldsymbol{I})$. Analogous to the 1D case in subsection 6.1, we first estimate the

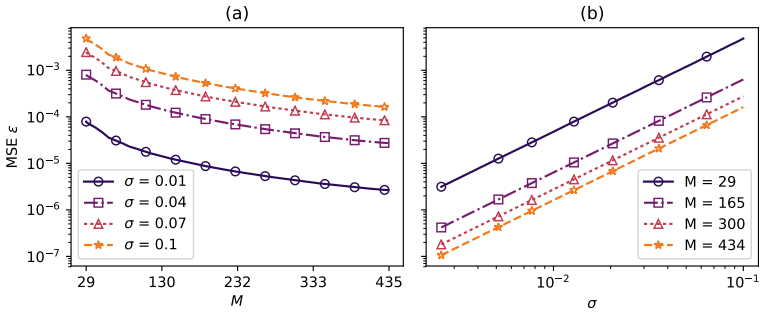


FIG. 4. *CDM estimation error ϵ. We consider $N = 30$ points drawn independently and uniformly from $[0, 1]$ and assume Gaussian noise on the differences with 0-mean and a variance of σ^2.* (a) *Dependence of ϵ on number of measurements M for fixed σ.* (b) *Dependence of ϵ on σ for fixed M.*

partial point sets $\widehat{\boldsymbol{x}}'_o$ and $\widehat{\boldsymbol{x}}'_s$ from (16) and (17), aligned such that their first points are $\mathbf{0}$. We create $\boldsymbol{J}'_D$ by generalizing the centering matrix $\boldsymbol{J}'$ to D dimensions, such that the sum of the point coordinates in every separate dimension is 0. It follows that the centered estimated points, $\widehat{\boldsymbol{x}}_o = \boldsymbol{J}'_D\widehat{\boldsymbol{x}}'_o$ and $\widehat{\boldsymbol{x}}_s = \boldsymbol{J}'_D\widehat{\boldsymbol{x}}'_s$, have Gaussian distributions with the following parameters:

$$\begin{aligned}\widehat{\boldsymbol{x}}_o &\sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_o}), \text{ where } \boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_o} = \sigma^2\boldsymbol{J}_D(\boldsymbol{B}\boldsymbol{\Phi})^\dagger((\boldsymbol{B}\boldsymbol{\Phi})^\dagger)^\top(\boldsymbol{J}_D)^\top,\\ \widehat{\boldsymbol{x}}_s &\sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_s}), \text{ where } \boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_s} = \sigma^2\boldsymbol{J}_D\boldsymbol{\Phi}^\dagger\boldsymbol{B}^\dagger(\boldsymbol{B}^\dagger)^\top(\boldsymbol{\Phi}^\dagger)^\top(\boldsymbol{J}_D)^\top.\end{aligned} \tag{25}$$

For a less cluttered notation, in (25) and the rest of the section, we omit the prime symbol $'$ on $\boldsymbol{J}_D, \boldsymbol{B}, \boldsymbol{\Phi}$.

We define the estimation error vectors of the optimal and splitting algorithms as $\boldsymbol{e}_o = \widehat{\boldsymbol{x}}_o - \boldsymbol{x}$ and $\boldsymbol{e}_s = \widehat{\boldsymbol{x}}_s - \boldsymbol{x}$, respectively. The expectations of the MSEs are $\mathbb{E}\left[\frac{1}{N}\|\boldsymbol{e}_o\|^2\right] = \frac{1}{N}\operatorname{tr}(\boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_o})$ and $\mathbb{E}\left[\frac{1}{N}\|\boldsymbol{e}_s\|^2\right] = \frac{1}{N}\operatorname{tr}(\boldsymbol{\Sigma}_{\widehat{\boldsymbol{x}}_s})$.

Simulation results. We consider Gaussian noise with 0-mean and $\sigma = 0.01$ added to the difference measurements, and we assume the complete case in 2D, such that both approaches are optimal. Then, for a given pair (N, K), we generate K directions of frame vectors uniformly at random from $[0, 2\pi)$ and the complete measurement matrix $\boldsymbol{B}$. Figure 5 shows that the estimation error decreases with the number of frame vectors K and the number of points N.

6.3. The cost of splitting. To evaluate the performance of the proposed splitting algorithm with respect to the optimal solution, we define the *cost of splitting* c as the normalized squared norm of the difference between the two estimators, $c = \frac{1}{N}\|\widehat{\boldsymbol{x}}_o - \widehat{\boldsymbol{x}}_s\|^2$. We can compute the expected cost of splitting as

$$\begin{aligned}\mathbb{E}[c] &= \frac{\sigma^2}{N}\operatorname{tr}\left[\left(\boldsymbol{J}_D\left((\boldsymbol{B}\boldsymbol{\Phi})^\dagger - \boldsymbol{\Phi}^\dagger\boldsymbol{B}^\dagger\right)\right)^\top\left(\boldsymbol{J}_D\left((\boldsymbol{B}\boldsymbol{\Phi})^\dagger - \boldsymbol{\Phi}^\dagger\boldsymbol{B}^\dagger\right)\right)\right]\\ &= \frac{\sigma^2}{N}\left(\operatorname{tr}\left[(\boldsymbol{J}_D)^\top\boldsymbol{J}_D\boldsymbol{\Phi}^\dagger\left(\boldsymbol{B}^\top\boldsymbol{B}\right)^{-1}(\boldsymbol{\Phi}^\dagger)^\top\right] - \operatorname{tr}\left[(\boldsymbol{J}_D)^\top\boldsymbol{J}_D\left((\boldsymbol{B}\boldsymbol{\Phi})^\top(\boldsymbol{B}\boldsymbol{\Phi})\right)^{-1}\right]\right)\\ &= \frac{\sigma^2}{N}\operatorname{tr}(\boldsymbol{\Psi}(\boldsymbol{\Phi}, \boldsymbol{B})) - \frac{\sigma^2}{N^2}\mathbb{1}^\top\boldsymbol{\Psi}(\boldsymbol{\Phi}, \boldsymbol{B})\mathbb{1},\end{aligned} \tag{26}$$

where $\boldsymbol{\Psi}(\boldsymbol{\Phi}, \boldsymbol{B}) = \boldsymbol{\Phi}^\dagger(\boldsymbol{B}^\top\boldsymbol{B})^{-1}(\boldsymbol{\Phi}^\dagger)^\top - (\boldsymbol{\Phi}^\top\boldsymbol{B}^\top\boldsymbol{B}\boldsymbol{\Phi})^{-1}$.

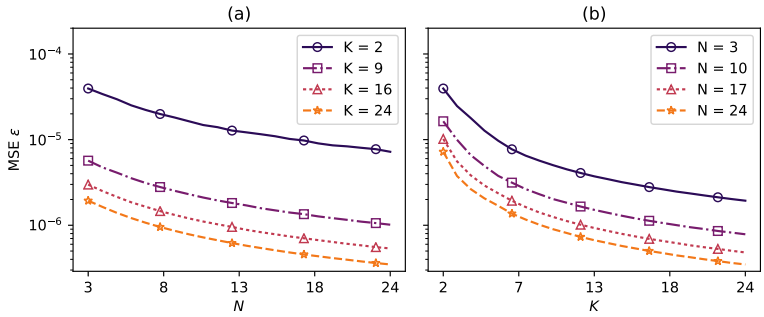


FIG. 5. *Estimation error ϵ in 2D. We assume complete CDM for every frame vector and Gaussian noise with 0-mean and $\sigma = 0.01$ on the measurements.* (a) *Dependence of ϵ on N for fixed K.* (b) *Dependence of ϵ on K for fixed N.*

As the splitting approach leads to a more efficient algorithm, identifying cases in which we can apply the splitting algorithm and still obtain an optimal solution is valuable. From (26) it follows that the cost is equal to 0 if $(\boldsymbol{B\Phi})^\dagger = \boldsymbol{\Phi}^\dagger \boldsymbol{B}^\dagger$ or, equivalently, if $\boldsymbol{\Psi}(\boldsymbol{\Phi}, \boldsymbol{B}) = \boldsymbol{0}$. For $\boldsymbol{B}$ and $\boldsymbol{\Phi}$ as defined in (14), we prove that there are two practical cases for which $\mathbb{E}[c] = 0$: (1) $K = D$ and (2) $\boldsymbol{B}_k = \boldsymbol{B}_0$ for every $k = 1, \dots, K$.

6.3.1. Case $K = D$. When the number of frame vectors is the same as the dimension of the space, $\boldsymbol{\Phi}$ is invertible; thus $\boldsymbol{\Phi}^\dagger = \boldsymbol{\Phi}^{-1}$. The splitting algorithm results in the optimal solution:

$$\begin{aligned}\widehat{\boldsymbol{x}}_o &= (\boldsymbol{B\Phi})^\dagger \widetilde{\boldsymbol{d}} \\ &= (\boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{B\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{B}^\top \widetilde{\boldsymbol{d}} \\ &= \boldsymbol{\Phi}^{-1} (\boldsymbol{B}^\top \boldsymbol{B})^{-1} \boldsymbol{B}^\top \widetilde{\boldsymbol{d}} = \widehat{\boldsymbol{x}}_s.\end{aligned}$$

6.3.2. Case $\boldsymbol{B_k} = \boldsymbol{B_0}$ for every $\boldsymbol{k}$. Assume that all measurement matrices for every frame direction are equal to the matrix $\boldsymbol{B}_0$. For instance, this is the case when we observe all pairwise differences on all frame vectors k. In that case, $\boldsymbol{B} = \boldsymbol{I} \otimes \boldsymbol{B}_0$, where $\otimes$ is the Kronecker product. We change the order of the entries in $\widetilde{\boldsymbol{d}}$ and the order of the corresponding columns in $\boldsymbol{\Phi}$, so that we can write $\boldsymbol{\Phi} = \boldsymbol{\Phi}_0 \otimes \boldsymbol{I}$, where the kth row of $\boldsymbol{\Phi}_0 \in \mathbb{R}^{K \times D}$ is $\boldsymbol{\varphi}_k^\top$ for every $k = 1, \dots, K$. Note that this does not influence the estimation of the points, but it provides a simpler expression for $\boldsymbol{\Phi}$ via the Kronecker product. Then, we can show that the optimal solution is equal to the solution of the splitting algorithm:

$$\begin{aligned}\widehat{\boldsymbol{x}}_o &= (\boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{B\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{d} \\ &= \Big((\boldsymbol{\Phi}_0^\top \otimes \boldsymbol{I}) \Big(\boldsymbol{I} \otimes (\boldsymbol{B}_0^\top \boldsymbol{B}_0) \Big) (\boldsymbol{\Phi}_0 \otimes \boldsymbol{I}) \Big)^{-1} \boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{d} \\ &= \Big((\boldsymbol{\Phi}_0^\top \boldsymbol{\Phi}_0)^{-1} \otimes (\boldsymbol{B}_0^\top \boldsymbol{B}_0)^{-1} \Big) \boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{d} \\ &= (\boldsymbol{I} \otimes (\boldsymbol{B}_0^\top \boldsymbol{B}_0)^{-1}) \Big((\boldsymbol{\Phi}_0^\top \boldsymbol{\Phi}_0)^{-1} \otimes \boldsymbol{I} \Big) (\boldsymbol{\Phi}_0 \otimes \boldsymbol{I}) \boldsymbol{B}^\top \boldsymbol{d} \\ &= \boldsymbol{\Phi}_0^\dagger \otimes \boldsymbol{B}_0^\dagger \boldsymbol{d} = \boldsymbol{\Phi}^\dagger \boldsymbol{B}^\dagger \boldsymbol{d} = \widehat{\boldsymbol{x}}_s.\end{aligned}$$

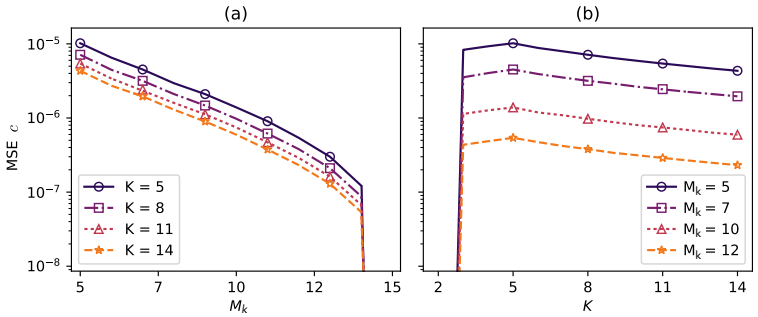


FIG. 6. *The cost of splitting c in 2D. We consider the setup of $N = 6$ points and assume Gaussian noise with 0-mean and $\sigma = 0.01$ on the measurements.* (a) *Dependence of c on M_k for fixed K.* (b) *Dependence of c on K for fixed M_k.*

Simulation results. To visualize the difference between the two approaches, in Figure 6 we plot the cost of splitting (26) for different number of measurements M_k and different number of frame vectors K. We consider $N = 6$ 2D points drawn from $[0, 1]^2$, so the number of measurements ranges from the smallest value $M_k = N - 1 = 5$ to the complete case $M_k = N(N-1)/2 = 15$. We assume that M_k is the same for every direction k. As shown in subsection 6.3.1 and subsection 6.3.2, the cost is equal to 0 for $K = D = 2$ and for the complete case; Figure 6 confirms these two special cases. In addition, we observe that the cost decreases with M_k and K.

6.4. Algorithmic considerations. In this section, we compare the proposed algorithms in terms of their numerical complexity and accuracy on simulated data. We experiment with three different implementations of the optimal and splitting methods. Two of the implementations are related to the vector form from subsection 4.2, and one is a direct implementation of our initial CDM formulation from subsection 3.1, generalized to higher dimensions. We describe them in more detail below.[2]

- *Standard implementation.* We implement the optimal and the splitting algorithms as described in subsection 4.3 and subsection 4.4, respectively. The matrices $\boldsymbol{B}$, $\boldsymbol{B}_k$, $\boldsymbol{\Phi}$ are represented with a standard linear algebra library. Least-squares solvers are implemented with the same library and use the SVD to solve the linear problem.
- *Sparse implementation.* The splitting and the optimal algorithms solve the same linear problem as in the standard implementation, but the matrices $\boldsymbol{B}$, $\boldsymbol{B}_k$, $\boldsymbol{\Phi}$ are represented as compressed sparse row matrices (CSRs).
- *CDM implementation.* To avoid costly computations resulting from the sparse and large matrices of the vector form, we leverage the matrix representation offered by CDMs. For the splitting strategy, we create K CDMs $\widetilde{\boldsymbol{C}}_k$, one for each frame vector. By counting the number of pairwise comparisons in every $\boldsymbol{B}_k$, we compose K weight matrices $\boldsymbol{W}_k$. For every k, we compute the matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ from $\widetilde{\boldsymbol{C}}_k$ and $\boldsymbol{W}_k$ as in subsection 3.1. Similarly, when implementing the optimal algorithm, we directly compute a matrix equivalent to the product $\boldsymbol{\Phi}^\top \boldsymbol{B}^\top \boldsymbol{B} \boldsymbol{\Phi} \in \mathbb{R}^{DN \times DN}$, without having to set up $\boldsymbol{\Phi}$ and $\boldsymbol{B}$, whose dimension M can be arbitrarily large. This is done through derivations analogous to subsection 3.1, where (3) is generalized to

[2]The methods are implemented using standard `Python` libraries and solvers on a Linux server with processor model *Intel(R) Xeon(R) CPU X5675 @ 3.07GHz*. The *standard* and *CDM* implementations are done using `numpy` and LAPACK's DGELSD solver. The *sparse* implementations use the `scipy.sparse` package and the sparse iterative LSQR solver [63].

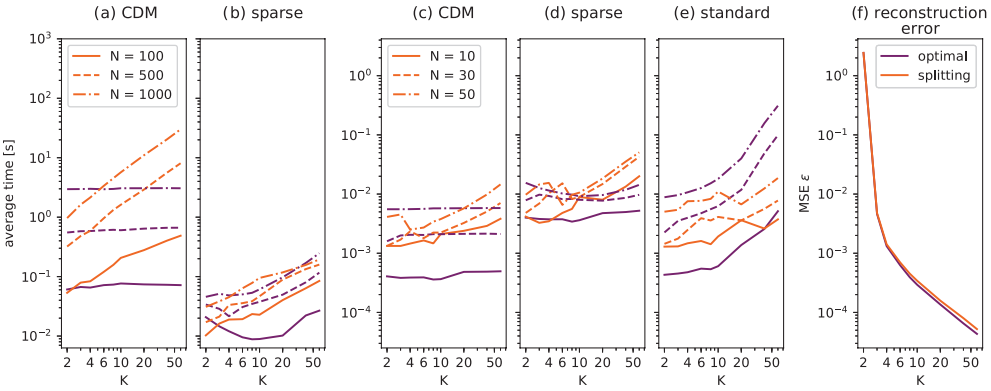


FIG. 7. *Evaluation of proposed methods in terms of efficiency (average running time in seconds) and reconstruction error (MSE). Figures* (a) *and* (b) *show results for high N, for which the standard implementation is infeasible due to memory issues. The number of directions ranges from the minimum required $K = D = 2$ to $K = 60$. For each implementation, the optimal and splitting approaches are highlighted in purple and orange, respectively. The reconstruction error in* (f) *is independent of the implementation strategy and the number of points N. The standard deviation of the Gaussian noise added to coordinate differences is $\sigma = 0.1$. (Color available online.)*

take into account all contributions from the frame,

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \left\| \boldsymbol{W}_k \circ (\boldsymbol{\Phi}_k \boldsymbol{x} \mathbb{1}^\top - \mathbb{1} \boldsymbol{x}^\top \boldsymbol{\Phi}_k^\top - \widetilde{\boldsymbol{C}}_k) \right\|_F^2 . \tag{27}$$

In all implementations, we exploit the distributed nature of the splitting approach by parallelizing it on eight CPU cores.

6.4.1. Simulation setup. Throughout all experiments, N 2D points are chosen uniformly at random from the interval $[0, \sqrt{N}]^2$, such that the average density is one point per unit area. The directions of K frame vectors are picked uniformly at random from $[0, 2\pi]$. We varied the number of points from $N = 10$ to $N = 1000$, and we chose an average of four measurements per frame vector and per point, while ensuring that the CDM for every k is connected. For such a connectivity value, the large memory requirement restricted us from simulating the standard implementations for $N \geq 100$.

The time and error averages are taken over 100 independent geometry and noise realizations. Note that the reported times do not include the time it takes to simulate the measurements. It is therefore a measure of how fast we can solve the different least squares problems with the proposed implementations.

6.4.2. Observations. The plots in Figure 7 show the simulation results of the three described implementations. There is no single method that outperforms others for every combination of N and K; thus we discuss below multiple important findings.

Figure 7a and Figure 7c confirm that the CDM implementation of the optimal method does not depend on K. This is a significant advantage over the implementation strategies that rely on the vector form, for which the execution times grow with K. Moreover, to optimally benefit from the CDM implementation, one should choose the splitting algorithm since it is faster than the optimal method when K is below a certain threshold dependent on N.

When N is larger than 100 and K is smaller than 60, the sparse implementations of the vector form exhibit shorter execution times than the CDM approach; this can

be seen by comparing Figure 7a and Figure 7b. However, as the execution times of sparse implementations increase with K, the CDM implementation of the optimal method surpasses its sparse implementation after a certain value of K.

In the standard implementation, the optimal solution relies on the SVD of a matrix whose number of rows equals the total number of measurements; this becomes prohibitively large with increasing K. As a consequence, the splitting algorithm becomes attractive for higher values of K; for more than 30 points splitting is on average always faster than the optimal solution. However, the standard implementation does not compare favorably either with the CDM implementation or with the sparse implementation for any N and K.

The last column shows that we lose little in terms of accuracy by applying the splitting algorithm instead of the optimal solution. The reconstruction error is independent of the number of points N and the type of implementation.

Finally, we would like to stress that more efficient and robust solvers for the systems of linear equations can be used. In particular, methods that avoid squaring the condition number and exploit the diagonally dominant structure of $\boldsymbol{A}$ are preferable for applications requiring fast performance. We expect that the splitting algorithm would specifically benefit from such methods, as the corresponding linear system involves the inversion of irreducibly diagonally dominant matrices, while this is not the case for the optimal formulation.

7. Applications and results. We showcase five seemingly different applications that can all be solved using CDMs; in particular, we show that the labeled vector geometry problem is present in many fields, ranging from sport ranking to molecular biology. We emphasize that our goal is not to demonstrate the superior performance of our method over state-of-the-art techniques but rather to demonstrate a wide range of applications of the CDM framework. For the first two applications, we provide a brief description of the problem as well as the connection with our proposed framework. The other three applications are described in more detail, along with simulations and experiments on real data.

7.1. 1D case: Rankings and ratings in sports. In the sports community, ranking teams based on their performances has been a long-standing question with a number of proposed solutions [45, 47, 17, 29, 14, 36, 15]. The problem statement is fairly simple—arrange teams in order such that the *better* team has a higher rank. Here, we focus on a more general problem called rating, which assigns to teams absolute scores that reflect their performance or strength. In fact, this is an instance of a 1D labeled vector problem that belongs to the CDM framework.

One of the most widely used rating algorithms in the sports community is called Massey's method [56]. It abstracts the problem by considering the strength of teams as points on a line. The strength of a team encompasses all its characteristics, including the quality of the players, points scored in the past games, statistics, etc. The greater the strength, the higher the likelihood that a team is going to win a game. Then, the net score of a game between two teams is the (noisy) difference between the strengths of the teams. Additionally, one might assign different importance to different games: for instance, the first game in a sport championship should probably have less weight than the final of the playoffs.

With this formulation, rating teams from their scores boils down to solving Problem 1. Massey's method tackles it by computing the solution to (11). In other words, Massey's method is a 1D case of the problem described in this manuscript and can be solved with (10).

7.2. Unlabeled vector geometry problems. We demonstrate that the CDM framework can also be leveraged in the context of the unlabeled vector geometry problem as a denoising tool.

A typical problem that arises in this field is the (noisy) turnpike problem [19], also known as partial digest. It finds applications in many fields, including X-ray crystallography and genetics, where the goal is to recover structures from measured distances. The noisy turnpike problem can be formulated as the reconstruction of points whose pairwise differences are the closest to a given set of measured differences. The distinction between this formulation and Problem 1 is that the labels are not known: in the context of CDMs, it corresponds to having access to the matrix elements but not their positions.

The main difficulty resides in inferring the proper labeling for the differences; in fact, it can be shown that this problem is NP-hard [16]. Nevertheless, approximative solutions exist. For instance, a greedy algorithm has been proposed in [4] as a building block to solve the problem of phase retrieval for sparse signals. This algorithm retrieves the labeling by iteratively selecting the point that is the most likely to generate a subset of the measured differences. For every partial estimate of the labeling, it leverages (10) to denoise the solution set. The authors show that this improves the final accuracy of the result.

7.3. 2D case with $K > D$: Sensor array calibration. We exploit CDMs for sensor array calibration, that is, the problem of determining the locations of sensors given measurements from external calibration sources. Sensor arrays have been employed in real-time monitoring and measurement for decades, and the importance of the accurate calibration of their relative positions is evident in numerous applications, such as source localization [55, 71, 64], source separation, and noise reduction [31].

We consider measurements from a number of sources placed in the far field at known locations. Even though the problem is of practical relevance, it has not received much attention in the literature. Many proposed methods consider the sources in the near field and assume their exact synchronization [8, 6, 34, 18, 32, 51, 80]. These approaches use measured distances between sources and sensors and are closely related to the framework of EDMs. Extensions to far-field calibration also exist [60, 28, 76, 50]; however, in these works the source locations are assumed to be unknown, and the algorithms iteratively estimate the direction of arrival along with the sensor locations, which is not necessary in the method we propose. We show that the knowledge of the source locations in the far field can be used to directly apply a CDM-based algorithm for point recovery in higher dimensions.

We aim to localize N sensors from the measurements of K calibration sources placed at angles $\{\boldsymbol{\varphi}_k\}_{k=1}^K$ in the far field. The far-field assumption implies that the sources emit plane waves, and the incident angle $\boldsymbol{\varphi}_k$ of some fixed source k is the same for all sensors. This notation is illustrated in Figure 8, along with the experimental setup in which we investigate the performance of the proposed method. The setup includes a microphone array called *Pyramic* [70, 3] and three loudspeakers in an anechoic chamber. Pyramic (see Figure 8b) is a pyramidal array composed of six branches, each containing eight microphones (see Figure 8c). The shortest distance between two microphones is 8 mm, while their maximum distance is 200 mm. As the loudspeakers are located as far away as possible to emulate far-field conditions, it is more practical to place the Pyramic array on a turntable and rotate it instead of the speakers, yielding the same desired relative orientation.

The measurements from the calibration sources are conducted in the following

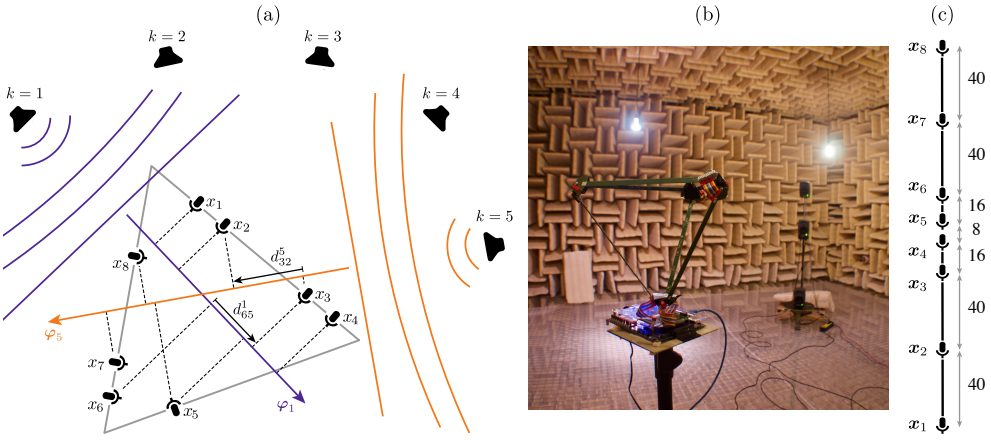


FIG. 8. *Calibration of sensor arrays: Geometry and real experiment.* (a) *Three edges of the Pyramic array with several microphones. The differences of the measured times of arrival for the calibration source k and two microphones i, j are denoted d^k_{ij}.* (b) *Pyramic array (foreground) and speakers (background) in an anechoic chamber.* (c) *Eight microphones placed in one branch of the Pyramic array, where distances are in mm.*

way: The speakers produce waves at unknown times, and the sensors register the absolute times of arrival (TOAs) of the waves, denoted τ_{nk} for the nth sensor and the kth source. If there is a sensor n that registered all the TOAs for every source k, then we can subtract τ_{nk} from the detected times τ_{mk} of all the sensors $m = 1, \dots, N$ for every direction k. This corresponds to fixing the nth sensor at the origin. However, if this is not the case, we cannot combine measurements from different sources k, as each of them has a different emission time. Nevertheless, we can compute the pairwise differences of the registered times, which brings us to the framework of CDMs. Such measurements can be seen as coordinate differences of the sensors' locations, projected onto the directions of the frame vectors $\boldsymbol{\varphi}_k$. In accordance with the theory developed in section 4, we can thus reconstruct the sensors' locations in D dimensions as long as we have $K \geq D$ noncollinear calibration sources and enough measurements for the connectivity requirement (see subsection 4.5).

We localize the 21 microphones of the top triangle of the Pyramic array using 90 calibration directions uniformly spaced in $[0, 180°)$. We use three loudspeakers at different heights in the far field (3.5 meters away). The measurement directions correspond to the directions of arrival of the sound played by the loudspeakers, so we have a total of $K = 3 \times 90 = 270$ vectors $\boldsymbol{\varphi_k}$.

All microphones worked properly throughout the experiment, so for every $k = 1, \dots, 270$ we can compute the pairwise differences of the sound detection times for all pairs chosen from the 21 microphones, arrange them in a measurement vector $\widetilde{\boldsymbol{d}}$, and create a corresponding matrix $\boldsymbol{B}_k$ as

$$\boldsymbol{B}_k = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 1 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \in \mathbb{R}^{210 \times 21}. \tag{28}$$

Then, we use the splitting algorithm (17) to find the locations of the microphones. As $\boldsymbol{B}_k = \boldsymbol{B}_0$ for every k, it leads to the optimal solution, but more efficiently than

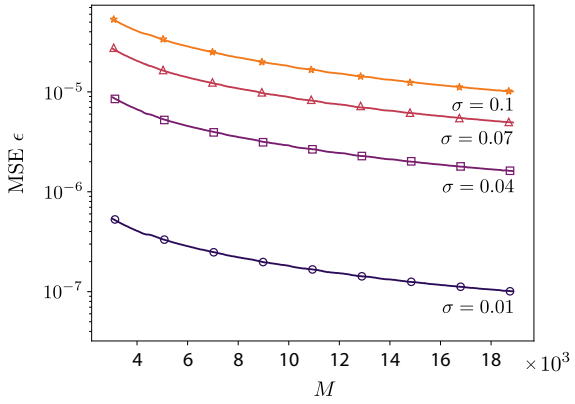


FIG. 9. *CDM estimation error ϵ. We consider $N = 21$ microphones in the plane, $K = 90$ frame vectors in the same plane, uniformly spread by $2°$, and Gaussian noise on the differences with 0-mean and a variance of σ^2. We illustrate the dependence of ϵ on number of measurements M for different values of σ. The number of measured differences varies from $M = 3000$ to the complete case, $M = 18900$.*

solving the original problem by (16). Our method localizes the microphones with an MSE of 4.49 µm when using all speakers and all directions. The error remains almost unchanged (4.45 µm) when considering only the middle speaker (placed at the same height as the top triangle of the Pyramic array), and localizing the microphones in 2D.

In addition to this real experiment, we perform numerical simulations in 2D with $K = 90$ and $N = 21$ to evaluate the impact on the number of active microphones on the robustness of the reconstruction. We start with a complete set of pairwise differences $M = \sum_{k=1}^{K} M_k = 90\binom{21}{2}$, as in the real experiment, and keep decreasing the value of M until we have a sparsely connected graph for every frame. The results are illustrated in Figure 9. These simulations confirm that the MSE decreases at a faster rate when adding a measurement in a relatively incomplete setup (see the left side of the graph). On the other hand, adding a measurement to an almost complete setup has little benefit.

7.4. 2D case for $K = D$: Multimodal sensor localization. We consider the self-localization of nodes in a sensor network, where the nodes can measure the distances and angles between each other. If we can only measure distances between the nodes, the problem is well studied and can be solved with a number of algorithms relying on the EDM theory [61, 18, 32, 22]. Similarly, if we only have access to the angles between nodes, several solutions have been proposed [12, 52, 67].

Setups leveraging both measurement modalities did not attract as much attention, despite many applications including indoor localization and self-calibration of ad hoc sensor arrays. The consistent combination of quantities of different unities into one framework is studied in [42]. For the particular case of angles and distances, trigonometric properties linking the two can be exploited, as in [7]. Macagnano and De Abreu [54] improve on this method by introducing edge-multidimensional scaling (E-MDS), which solves an MDS problem adapted for the vectors between points.

We propose constructively combining distance and angle measurements into CDMs, which allows us to localize sensors in closed form. We compare our algorithm with the state-of-the-art solution [54] and demonstrate its superior performance.

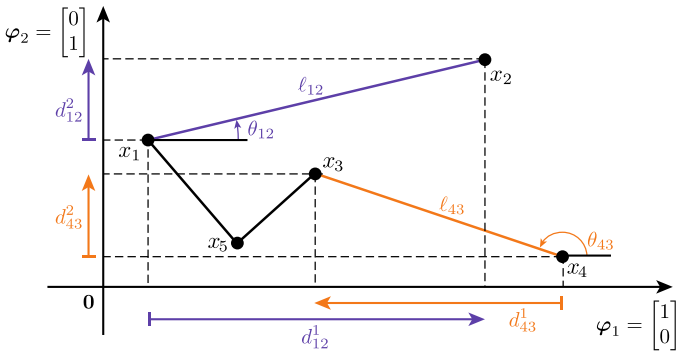


FIG. 10. *Localization setup in 2D. The difference vector $\boldsymbol{d}_{ij}$ between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined either with its distance ℓ_{ij} and directed angle θ_{ij}, or with its projections onto the x-axis and y-axis, d^1_{ij} and d^2_{ij}, respectively.*

Consider a set of N points in 2D with coordinates denoted by $\boldsymbol{x}_i \in \mathbb{R}^2$ for $i = 1, \ldots, N$. The vector of coordinate differences $\boldsymbol{d}_{ij} \in \mathbb{R}^2$ between the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is given by $\boldsymbol{d}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$. Its length is the Euclidean distance between the points, $\ell_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$, while its orientation θ_{ij} lies between 0 and 2π and is defined with respect to some common reference direction (see Figure 10). Note that the angles are often measured in a coordinate system that is local and different for each node. Distributed algorithms such as [25] can convert these relative angles to a common coordinate system.

Our goal is to estimate points $\widehat{\boldsymbol{x}}_i$ from a noisy subset of measured distances $\widetilde{\ell}_{ij}$ and angles $\widetilde{\theta}_{ij}$. Once again, we can leverage CDMs; the noisy coordinate differences are recovered from $\widetilde{\ell}_{ij}$ and $\widetilde{\theta}_{ij}$ as $\widetilde{\boldsymbol{d}}_{ij} = [\widetilde{\ell}_{ij} \cos \widetilde{\theta}_{ij} \quad \widetilde{\ell}_{ij} \sin \widetilde{\theta}_{ij}]^\top$. We observe that this is a 2D point recovery problem, which can be optimally solved with the splitting algorithm from subsection 4.4; we can decompose the problem into two independent subproblems and resolve each independently with (10). In this case, we always work in the canonical basis; hence $\boldsymbol{\Phi} = \boldsymbol{I}$.

We compare the CDM algorithm with the state of the art in multimodal localization, E-MDS [54], and the most common approach for range-only measurements, MDS [49]. We consider $N = 10$ points chosen uniformly at random from $[0, 1]^D$. As in most real-world applications, the measurements of distances and angles are obtained in an independent manner from time-of-arrival and angle-of-arrival estimates. We also generate independent additive noise for these quantities. We assume Gaussian noise with 0-mean and standard deviation σ_ℓ and σ_θ, respectively. Note that the noise exceeding $\pm\pi$ will distort the angle noise distribution, but for the range of standard deviations chosen in these experiments, this effect is negligible.

We evaluate the performance of the three algorithms using the root mean squared error (RMSE) between the original and the estimated point sets and illustrate its dependence on the noise levels in Figure 11. For a more convenient comparison, we slice the graphs at four different values of σ_ℓ and σ_θ. Figure 11a shows the dependence of the RMSE on the distance noise level for two chosen levels of angle noise: low ($\sigma_\theta = 0.11$) and high ($\sigma_\theta = 0.5$). We observe that for low σ_θ it is advantageous to include angle measurements, since both multimodal methods achieve smaller error than MDS. For higher σ_θ, our method still outperforms MDS, as long as the distance noise is not too small. The dependence of the RMSE on the angle noise level for

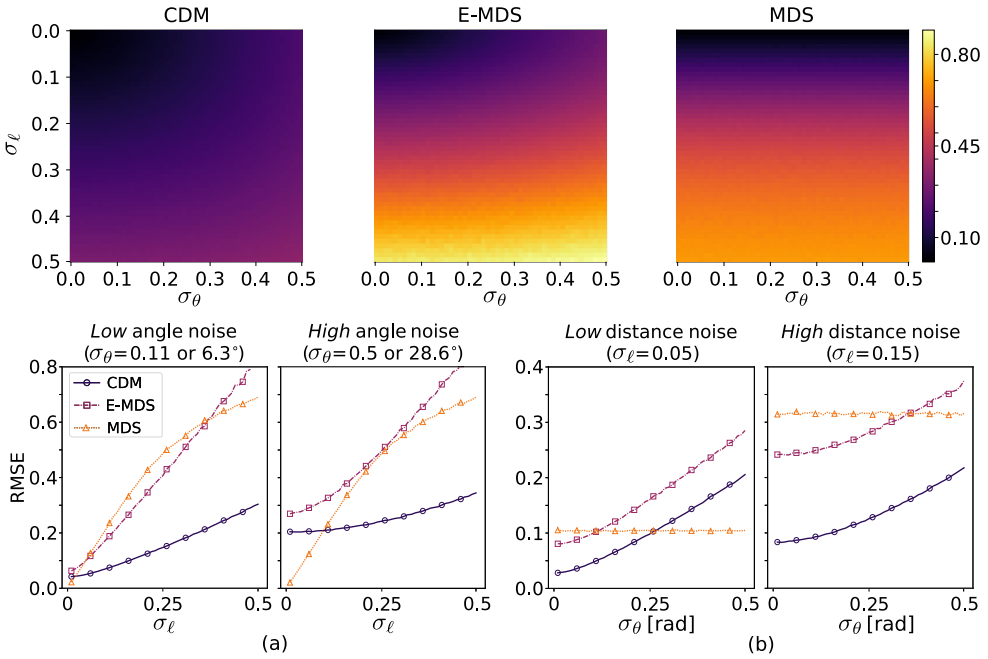


FIG. 11. *Comparison of the CDM with E-MDS and MDS for different distance and angle noise.* (a) *RMSE versus* σ_ℓ *for two fixed* σ_θ. (b) *RMSE versus* σ_α *for two fixed* σ_ℓ.

two chosen levels of distance noise is shown in Figure 11b. For low distance noise ($\sigma_\ell = 0.05$), we observe that using multimodal methods is beneficial only when angle noise is low; otherwise the angle information becomes detrimental, and one should rely on the distance-based method MDS. However, for higher distance noise ($\sigma_\ell = 0.15$), using angles significantly improves the result for all considered noise levels.

Numerical simulations show that the proposed algorithm based on CDMs surpasses the state-of-the-art multimodal localization method E-MDS for every pair of $(\sigma_\ell, \sigma_\theta)$. It also compares favorably with range-only based method MDS, except for the case of high σ_θ and low σ_ℓ.

7.5. 3D case with splitting algorithm: Molecular conformation problem. Determining the 3D structure of a protein molecule is one of the most important and at the same time most challenging problems in biology [1]. A common experimental method for finding the parameters pertaining to molecule structure is nuclear magnetic resonance (NMR) spectroscopy. The most relevant signals provided by NMR spectroscopy are distance restraints between pairs of atoms close to each other in the molecule [43]. These distances are either between the nuclei of two bonded atoms (covalent bond distances) or between the closest approach of two nonbonded atoms (van der Waals distances).

Another important piece of information extracted from NMR data is the torsion angle [27], defined as the angle between planes through two sets of three atoms with two atoms in common. Torsion angles are reported as intervals in which the true torsion angles lie [43]. In addition to distances and torsion angles, one can assume that the angles between three consecutive bonded atoms (bond angles) are known and accurate in a molecule. Figure 12a illustrates the definitions of the above quantities.

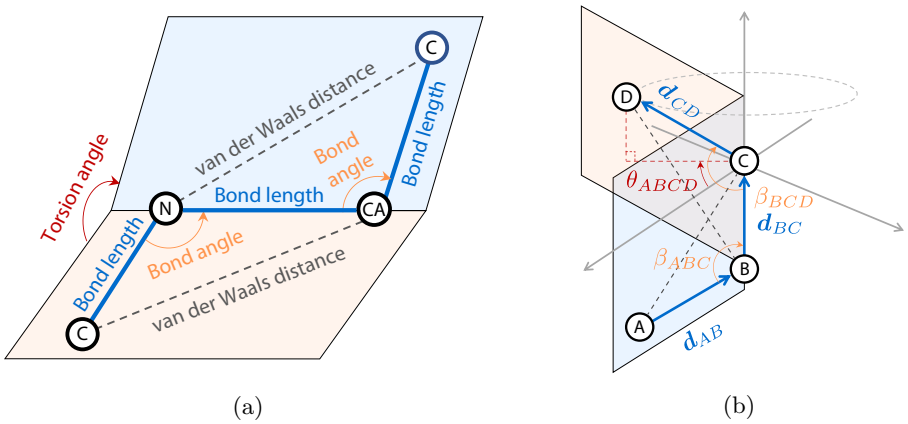


FIG. 12. (a) *A small part of the backbone of protein* 1*G*6*J, Ubiquitin. We illustrate and indicate bond angles, bond lengths, torsion angles, and van der Waals distances.* (b) *Coordinate system for four consecutive atoms used in the proposed reconstruction algorithm to independently construct three CDMs, one for each basis.*

The molecular conformation problem consists of finding the position of the atoms given measurements of torsion angles and distances, along with the information about the bond angles. A large category of the existing approaches uses Euclidean distance matrices consisting of bond lengths and van der Waals distances to estimate the molecule structure [10, 11, 35]. The authors in [53] provide a thorough summary of such approaches. A more recent overview of the methods, including the case where the distances are not assigned, is found in [5]. Other approaches combine the distance measurements together with the information about torsion angles. For example, Güntert, Mumenthaler and Wüthrich transform the problem in the so-called torsion-angle space and estimate the geometry in that space [38, 39]. Several methods then take the torsion angles and sequentially find the coordinates of the atoms using simple geometrical angular relations [9, 75, 66], often assuming a constant bond length and ignoring van der Waals distances. In a different approach, Alipanahi Ramandi constructs a semidefinite program using the distances and adds torsion angles as constraints to the SDP [1].

Here we propose a new approach that combines labeled distance and angle information to construct CDMs in 3D and solve the molecular conformation problem as a recovery of points from the noisy and incomplete measurements of their coordinate differences. In this paper, we focus on reconstructing the structure of the protein molecule backbone (which consists of repetitions of three atoms: nitrogen, alpha carbon, and carbonyl carbon). NMR spectroscopy also produces distance constraints between pairs of hydrogen atoms in the protein. However, as in this application we combine torsion angles and distances for estimating the protein structure, we only reconstruct the protein backbone, ignoring the hydrogen atoms.

Let us start by formulating the relation between bond distances, bond angles, and torsion angles with CDMs. We look at four consecutive atoms A, B, C, and D in a molecule backbone as depicted in Figure 12b. We define $\ell_{AB}, \ell_{BC}, \ell_{CD}$ as the bond distances, ℓ_{AC}, ℓ_{BD} as the van der Waals distances, β_{ABC}, β_{BCD} as the bond angles, and θ_{ABCD} as the torsion angle. We assume that the distance ℓ_{AD} does not fall in the van der Waals radius and is unknown. We set the atom C to the origin of

the coordinate system and the z-axis along bond B-C. More formally, we define the canonical basis vectors in this local system as follows:

$$\boldsymbol{z} = \frac{\boldsymbol{d}_{BC}}{\|\boldsymbol{d}_{BC}\|}, \quad \boldsymbol{y} = \frac{\boldsymbol{d}_{AB} \times \boldsymbol{d}_{BC}}{\|\boldsymbol{d}_{AB} \times \boldsymbol{d}_{BC}\|}, \quad \boldsymbol{x} = \boldsymbol{y} \times \boldsymbol{z}, \tag{29}$$

where $\boldsymbol{d}_{AB}$ denotes the vector difference between points A and B.

To shorten the notation, we introduce the following variables: $\mathcal{C}_{\beta_1} = \cos(\beta_{ABC})$, $\mathcal{S}_{\beta_1} = \sin(\beta_{ABC})$, $\mathcal{C}_{\beta_2} = \cos(\beta_{BCD})$, $\mathcal{S}_{\beta_2} = \sin(\beta_{BCD})$, $\mathcal{C}_{\theta} = \cos(\theta_{ABCD})$, and $\mathcal{S}_{\theta} = \sin(\theta_{ABCD})$. Given the measurements of the distances and angles described above, for every basis $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ defined in (29), we construct a CDM as follows:

$$\boldsymbol{C}^x = \begin{bmatrix} 0 & \ell_{AB}\mathcal{S}_{\beta_1} & \ell_{AB}\mathcal{S}_{\beta_1} & \text{NA} \\ -\ell_{AB}\mathcal{S}_{\beta_1} & 0 & 0 & -\ell_{CD}\mathcal{S}_{\beta_2}\mathcal{C}_{\theta} \\ -\ell_{AB}\mathcal{S}_{\beta_1} & 0 & 0 & -\ell_{CD}\mathcal{S}_{\beta_2}\mathcal{C}_{\theta} \\ \text{NA} & \ell_{CD}\mathcal{S}_{\beta_2}\mathcal{C}_{\theta} & \ell_{CD}\mathcal{S}_{\beta_2}\mathcal{C}_{\theta} & 0 \end{bmatrix}, \tag{30}$$

$$\boldsymbol{C}^y = \begin{bmatrix} 0 & 0 & 0 & \text{NA} \\ 0 & 0 & 0 & \ell_{CD}\mathcal{S}_{\beta_2}\mathcal{S}_{\theta} \\ 0 & 0 & 0 & \ell_{CD}\mathcal{S}_{\beta_2}\mathcal{S}_{\theta} \\ \text{NA} & -\ell_{CD}\mathcal{S}_{\beta_2}\mathcal{S}_{\theta} & -\ell_{CD}\mathcal{S}_{\beta_2}\mathcal{S}_{\theta} & 0 \end{bmatrix}, \tag{31}$$

$$\boldsymbol{C}^z = \begin{bmatrix} 0 & \ell_{AB}\mathcal{C}_{\beta_1} & -C^z_{CA} & \text{NA} \\ -\ell_{AB}\mathcal{C}_{\beta_1} & 0 & -\ell_{BC} & -C^z_{DB} \\ C^z_{CA} & \ell_{BC} & 0 & \ell_{CD}\mathcal{C}_{\beta_2} \\ \text{NA} & C^z_{DB} & -\ell_{CD}\mathcal{C}_{\beta_2} & 0 \end{bmatrix}, \tag{32}$$

where

$$C^z_{CA} = -\text{sign}(C^z_{BA} + C^z_{CB})\,\ell_{AC}\sqrt{1 - \left(\frac{\ell_{AB}}{\ell_{AC}}\right)^2 \mathcal{S}^2_{\beta_1}},$$

$$C^z_{DB} = -\text{sign}(C^z_{CB} + C^z_{DC})\,\ell_{BD}\sqrt{1 - \left(\frac{\ell_{CD}}{\ell_{BD}}\right)^2 \mathcal{S}^2_{\beta_2}}.$$

For every CDM $\boldsymbol{C}^x$, $\boldsymbol{C}^y$, and $\boldsymbol{C}^z$, we estimate the atoms' coordinates separately in each dimension using (10). This method allows us to estimate the 3D location of four consecutive atoms in the molecule backbone. To find the structure of the entire molecule backbone, we perform the above procedure sequentially on 4-tuples of consecutive atoms in the molecule. We select sequences of four consecutive atoms, use our reconstruction algorithm to estimate their positions in their own coordinate systems, and then apply orthogonal Procrustes [2] to optimally align the consecutive 4-tuples using their three common atoms. This procedure results in four different coordinates for each atom (except for atoms at the boundary of the chain), and we compute the final position as the average of these four positions. This is shown in Figure 13.

To evaluate the efficacy of using CDMs for molecular conformation, we consider the backbone of protein 1G6J, Ubiquitin,[3] which consists of 228 atoms. To mimic restraints on distances and torsion angles from NMR spectroscopy, we assume that the distances and torsion angles are corrupted by random uniform noise, $\widetilde{d}_{AB} = d_{AB} + z_d, \widetilde{\theta}_{ABCD} = \theta_{ABCD} + z_\theta$, where $z_d \in \mathcal{U}(-z_{d,\max}, z_{d,\max})$ and $z_\theta \in \mathcal{U}(-z_{\theta,\max}, z_{\theta,\max})$.

[3]Downloaded from https://files.rcsb.org/download/1G6J.pdb. We use the coordinates presented in model 1 as the ground truth.

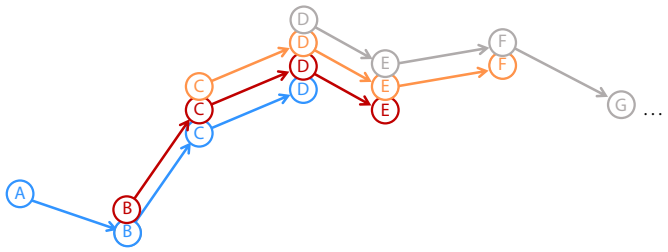


FIG. 13. *We select sequences of four consecutive atoms, use our CDM reconstruction algorithm to estimate their positions, and then align them using their three common atoms in consecutive* 4*-tuples.*

Bond angles are typically considered to be known and fixed [1], so we do not apply noise to them.

In the following experiment, we corrupt the distances and torsion angles with uniform and independent noise for different values of $z_{d,\max}$ and $z_{\theta,\max}$, apply our reconstruction algorithm to estimate the backbone structure of the protein, and compute the RMSE with respect to the original atoms. Generally, for reconstructions with RMSE values of less than 1.5 Å, the structure is considered high-resolution and useful for drug design. An RMSE value of around 4 to 6 Å can still be useful, while RMSE values above 10 Å are considered significantly inaccurate [48].

Figure 14a shows the average RMSE for the values of $z_{d,\max}$ and $z_{\theta,\max}$ ranging from 0 to 0.5. Note that torsion angle measurements are normally reported in intervals of length 10–30 degrees, and the average bond length for this protein is about 2 Å. We see that CDM reconstruction provides useful results in the presence of measurement noise.

While our assumption about the noise model is slightly different than what is normally reported as NMR measurements in practice, we mimic a similar scenario in which we can compare the results of our method to other state-of-the-art approaches, as described in the following. We again look at the structure of protein 1G6J and compare our reconstruction accuracy to that of SPROS [1], which is a typical approach to recover the atom positions. SPROS starts from a random structure that satisfies the measured upper and lower bounds on distances and finds the optimal solution through a semidefinite program. The NMR torsion angle measurements for protein 1G6J are reported with intervals of length 10 degrees; this translates to $z_{\theta,\max} = 0.175$ radians in our setting. Moreover, the maximum discrepancy between the pairwise distances of the random structure and the true molecule in SPROS is on average 0.11 for the backbone atoms; we use this value to set $z_{d,\max} = 0.11$ Å. By setting these noise bounds to our experiment, we achieve an average RMSE of 0.48 Å over 5000 trials, while SPROS results in an average RMSE of 0.68 Å [1]. Figure 14b illustrates one example of the original and reconstructed backbone from our experiments.

To summarize this section, we have demonstrated that CDMs can be incorporated in a simple build-up algorithm suitable for the molecular conformation problem. Here we only scratch the surface of the potential of CDMs for this application, but we do hope that we provide valuable insight for practitioners and researchers in the field.

8. Conclusion. The main contribution of our work is the formulation and analysis of coordinate difference matrices (CDMs), simple tools that enabled us to introduce an efficient optimization framework for reconstructing point sets from their noisy and

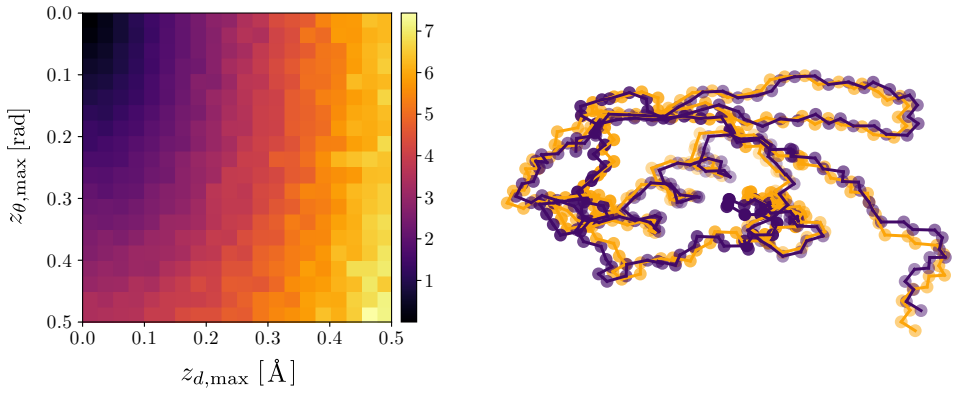


FIG. 14. (a) *RMSE for different values of the noise bounds* $z_{d,\max}$ *and* $z_{\theta,\max}$*, averaged over* 5000 *experiments.* (b) *An example of the original (orange) and reconstructed (blue) backbone of* 1*G6J with* $z_{d,\max} = 0.11$ Å *and* $z_{\theta,\max} = 0.175$ *rad, resulting in a reconstruction RMSE of* 0.48 Å. *(Color available online.)*

partial coordinate differences. The structure of CDMs also relates to graph theory, which in turn helped us provide necessary and sufficient conditions for the proposed framework to work. A significant advantage of CDMs is their easy generalization to the multidimensional setting, where we proposed several efficient and optimal methods for point reconstruction. Their potential is evident in practice: we have put the theoretical findings of this work to the test in five different practical applications in very active research fields and have shown that CDMs can be used to solve a wide range of problems with both simulated and real data.

Appendix A. Proofs of CDM properties in Table 1.

P.1. Following a basic rank inequality, $\operatorname{rank}(\boldsymbol{C}) \leq \operatorname{rank}(\boldsymbol{x}\mathbb{1}^\top) + \operatorname{rank}(\mathbb{1}\boldsymbol{x}^\top) = 2$. Observe that $\operatorname{rank}(\boldsymbol{C}) = 1$ can only happen when $N = 1$.

P.2. $C_{ij} = x_i - x_k + x_k - x_j = C_{ik} + C_{kj}$.

P.3. $C_{ij} = x_i - x_j = -(x_j - x_i) = -C_{ji}$.

P.4. $C_{ii} = x_i - x_i = 0$.

P.5. Follows from Properties P.3 and P.4.

P.6. $\sum_j C_{ij} = \sum_j (x_i - x_j) = Nx_i - \sum_j x_j$. Dividing both sides by N, $\frac{1}{N}\sum_j C_{ij} = x_i + c$, where $c = -\frac{1}{N}\sum_j x_j$.

P.7. Follows from (1): every column j of a CDM is equal to $\boldsymbol{x}$ shifted by $-x_j$.

P.8. $(\boldsymbol{x} + c)\mathbb{1}^\top - \mathbb{1}(\boldsymbol{x} + c)^\top = \boldsymbol{x}\mathbb{1}^\top - \mathbb{1}\boldsymbol{x}^\top$.

P.9. Denote λ as an eigenvalue of $\boldsymbol{C}$ and $\boldsymbol{v}$ as a corresponding eigenvector. Then, $\langle \boldsymbol{C}\boldsymbol{v}, \boldsymbol{v}\rangle = \left\langle \boldsymbol{v}, \boldsymbol{C}^\top \boldsymbol{v}\right\rangle = -\langle \boldsymbol{v}, \boldsymbol{C}\boldsymbol{v}\rangle$. Observe further that $\langle \boldsymbol{C}\boldsymbol{v}, \boldsymbol{v}\rangle = \langle \lambda\boldsymbol{v}, \boldsymbol{v}\rangle = \lambda\|\boldsymbol{v}\|^2$, and $-\langle \boldsymbol{v}, \boldsymbol{C}\boldsymbol{v}\rangle = -\langle \boldsymbol{v}, \lambda\boldsymbol{v}\rangle = -\lambda^*\|\boldsymbol{v}\|^2$. We conclude that $\lambda = -\lambda^*$; i.e., the eigenvalues of $\boldsymbol{C}$ are imaginary.

P.10. Since $\operatorname{rank}(\boldsymbol{C}) = 2$, we can denote nonzero eigenvalues as λ_1 and λ_2. Knowing that $\operatorname{eig}(\boldsymbol{C}) = \operatorname{eig}(\boldsymbol{C}^\top)$ and $\boldsymbol{C} = -\boldsymbol{C}^\top$, we can write

$$\det(\boldsymbol{I} - \lambda\boldsymbol{C}) = \det\left(\boldsymbol{I} - \lambda\boldsymbol{C}^\top\right) = \det(\boldsymbol{I} - (-\lambda)\boldsymbol{C}). \tag{33}$$

Therefore, if λ is an eigenvalue of $\boldsymbol{C}$, then $-\lambda$ is also an eigenvalue. As there

are only two nonzero eigenvalues, it follows that $\lambda_1 = -\lambda_2$.

Appendix B. Proof of convexity. The Hessian matrix $\boldsymbol{H}$ of $f(\boldsymbol{x})$ has entries

$$H_{ij} = \begin{cases} 2\sum_{k=1}^{N} W_{ik} & \text{for } i = j, \\ -2W_{ij} & \text{otherwise.} \end{cases} \tag{34}$$

Hence it is positive semidefinite for nonnegative values of the weight matrix $\boldsymbol{W}$. Indeed, for any vector $\boldsymbol{y} \in \mathbb{R}^N$, the value $\boldsymbol{y}^\top \boldsymbol{H}\boldsymbol{y}$ is nonnegative:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} H_{ij} y_i y_j = \sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}(y_i^2 - 2y_i y_j + y_j^2) \geq 0. \tag{35}$$

Appendix C. Proof of convergence. Let us define matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ as

$$\boldsymbol{P} = \sum_k \boldsymbol{P}_0^k = \sum_{k=0}^{\infty} \left(\frac{\mathbb{1}\mathbb{1}^\top - \boldsymbol{I}}{N-1} \right)^k, \quad \boldsymbol{Q} = \frac{N-11}{N}(\mathbb{1}\mathbb{1}^\top + \boldsymbol{I}). \tag{36}$$

We need to show that $\boldsymbol{P} = \boldsymbol{Q}$. To this end, we establish that the two matrices have the same eigenvalues and corresponding eigenvectors. First, observe that we can rewrite any square matrix in $\mathbb{R}^{(N-1)\times(N-1)}$ that has diagonal values equal to α and nondiagonal values equal to β as

$$\beta \mathbb{1}\mathbb{1}^\top + (\alpha - \beta)\boldsymbol{I} = (\alpha - \beta)\left(\boldsymbol{I} + \frac{\beta}{\beta - \alpha}\mathbb{1}\mathbb{1}^\top \right). \tag{37}$$

From the matrix determinant lemma, we know that

$$\det\left(\beta \mathbb{1}\mathbb{1}^\top + (\alpha - \beta)\boldsymbol{I}\right) = (\alpha - \beta)^{N-2}\left(1 + (N-1)\frac{\beta}{\alpha - \beta}\right). \tag{38}$$

Using (38), we compute the eigenvalues of $\boldsymbol{P}_0$ by solving its characteristic equation:

$$\det\left(\boldsymbol{P}_0 - \lambda \boldsymbol{I}\right) = \left(\lambda + \frac{1}{N-1}\right)^{N-2}\left(1 - \frac{N-1}{\lambda(N-1)+1}\right) = 0. \tag{39}$$

Thus, the eigenvalues of $\boldsymbol{P}_0$ are $\lambda_0 = \frac{-1}{N-1}$ (with multiplicity $N-2$) and $\lambda_1 = \frac{N-2}{N-1}$. The two eigenvalues of $\boldsymbol{P}$ are consequently equal to

$$\kappa_0 = \sum_{k=0}^{\infty}\left(\frac{-1}{N-1}\right)^k = \frac{N-1}{N}, \quad \kappa_1 = \sum_{k=0}^{\infty}\left(\frac{N-2}{N-1}\right)^k = N-1. \tag{40}$$

Furthermore, the eigenvectors of $\boldsymbol{P}_0$ (and by extension of $\boldsymbol{P}$) can be found by solving the eigenvalue equations

$$\begin{aligned} (\boldsymbol{P}_0 - \lambda_0 \boldsymbol{I})\boldsymbol{u}_0 &= \mathbb{1}\mathbb{1}^T \boldsymbol{u}_0 = \boldsymbol{0}, \\ (\boldsymbol{P}_0 - \lambda_1 \boldsymbol{I})\boldsymbol{u}_1 &= \left(\mathbb{1}\mathbb{1}^T - (N-1)\boldsymbol{I}\right)\boldsymbol{u}_1 = \boldsymbol{0}. \end{aligned} \tag{41}$$

We use the matrix determinant lemma again to compute the eigenvalues of $\boldsymbol{Q}$:

$$\det\left(\boldsymbol{Q} - \lambda \boldsymbol{I}\right) = \left(\lambda - \frac{N-1}{N}\right)^{N-2}\left(1 + \frac{(N-1)^2}{N - 1 - \lambda N}\right) = 0. \tag{42}$$

The corresponding eigenvalues are $\frac{N-1}{N}$ (with multiplicity $N-2$) and $N-1$, which are the same as $\boldsymbol{P}$. Moreover, the eigenvectors of $\boldsymbol{Q}$ can be found by solving

$$
\begin{aligned}
(\boldsymbol{Q} - \kappa_0 \boldsymbol{I})\boldsymbol{v}_0 &= \mathbb{1}\mathbb{1}^T \boldsymbol{v}_0 = \boldsymbol{0}, \\
(\boldsymbol{Q} - \kappa_1 \boldsymbol{I})\boldsymbol{v}_1 &= \left(\mathbb{1}\mathbb{1}^T - (N-1)\boldsymbol{I}\right) \boldsymbol{v}_1 = \boldsymbol{0}.
\end{aligned} \tag{43}
$$

By comparing equations (41) and (43), we see that matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ also have equal eigenvectors and thus are equal.

Acknowledgments. The authors would like to express their gratitude to Adam Scholefield, Michalina Pacholska, and the anonymous referees for their constructive comments, which helped improve the manuscript.

REFERENCES

[1] B. Alipanahi Ramandi, *New Approaches to Protein NMR Automation*, Ph.D. thesis, University of Waterloo, Waterloo, Canada, 2011.

[2] K. S. Arun, T. S. Huang, and S. D. Blostein, *Least-squares fitting of two 3-d point sets*, IEEE Trans. Pattern Anal. Machine Intell., 9 (1987), pp. 698–700.

[3] J. Azcarreta Ortiz, *Pyramic array: An FPGA Based Platform for Multi-channel Audio Acquisition*, Master's thesis, EPFL, Lausanne, Switzerland, and Universitat Politècnica de Catalunya, Barcelona, Spain, 2016.

[4] G. Baechler, M. Kreković, J. Ranieri, A. Chebira, Y. M. Lu, and M. Vetterli, *Super resolution phase retrieval for sparse signals*, IEEE Trans. Signal Process., 67 (2019), pp. 4839–4854.

[5] S. J. Billinge, P. M. Duxbury, D. S. Gonçalves, C. Lavor, and A. Mucherino, *Recent results on assigned and unassigned distance geometry with applications to protein molecules and nanostructures*, Ann. Oper. Res., 271 (2018), pp. 161–203.

[6] S. T. Birchfield and A. Subramanya, *Microphone array position calibration by basis-point classical multidimensional scaling*, IEEE Trans. Speech Audio Process., 13 (2005), pp. 1025–1034.

[7] P. Biswas, *Semidefinite Programming Approaches to Distance Geometry Problems*, Ph.D. dissertation, Stanford University, Stanford, CA, 2007.

[8] R. Biswas and S. Thrun, *A passive approach to sensor network localization*, in IEEE/RSJ International Conference Intelligent Robots and Systems (IROS), IEEE, Washington, DC, 2004, pp. 1544–1549.

[9] O. Bottema and B. Roth, *Theoretical Kinematics*, Corrected reprint of the 1979 ed., Dover, New York, 1990.

[10] W. Braun, C. Bösch, L. R. Brown, N. Gō, and K. Wüthrich, *Combined use of proton-proton Overhauser enhancements and a distance geometry algorithm for determination of polypeptide conformations. Application to micelle-bound glucagon*, Biochimica et Biophysica Acta (BBA) Protein Structure, 667 (1981), pp. 377–396.

[11] W. Braun and N. Gō, *Calculation of protein conformations by proton-proton distance constraints: A new efficient algorithm*, J. Molecular Biol., 186 (1985), pp. 611–626.

[12] J. Bruck, J. Gao, and A. A. Jiang, *Localization and routing in sensor networks by local angle information*, ACM Trans. Sensor Networks, 5 (2009), 7.

[13] F. Buekenhout and M. Parker, *The number of nets of the regular convex polytopes in dimension* ≤ 4, Discrete Math., 186 (1998), pp. 69–94.

[14] T. Callaghan, P. J. Mucha, and M. A. Porter, *Random walker ranking for NCAA division I-A football*, Amer. Math. Monthly, 114 (2007), pp. 761–777.

[15] T. P. Chartier, E. Kreutzer, A. N. Langville, and K. E. Pedings, *Sensitivity and stability of ranking vectors*, SIAM J. Sci. Comput., 33 (2011), pp. 1077–1102, https://doi.org/10.1137/090772745.

[16] M. Cieliebak, S. Eidenbenz, and P. Penna, *Partial digest is hard to solve for erroneous input data*, Theoret. Comput. Sci., 349 (2005), pp. 361–381.

[17] W. N. Colley, *Colley's Bias Free College Football Ranking Method: The Colley Matrix Explained*, Princeton University, Press, Princeton, NJ, 2002.

[18] M. Crocco, A. D. Bue, and V. Murino, *A bilinear approach to the position self-calibration of multiple sensors*, IEEE Trans. Signal Process., 60 (2012), pp. 660–673.

[19] T. Dakic, *On the Turnpike Problem*, Ph.D. thesis, Simon Fraser University, Burnaby, BC, Canada, 2000.
[20] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*, Meboo, Palo Alto, CA, 2005.
[21] J. De Leeuw, *Applications of Convex Analysis to Multidimensional Scaling*, Department of Statistics, UCLA, Los Angeles, CA, 2005.
[22] I. Dokmanić, L. Daudet, and M. Vetterli, *From acoustic room reconstruction to SLAM*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Washington, DC, 2016, pp. 6345–6349.
[23] I. Dokmanić, R. Parhizkar, J. Ranieri, and M. Vetterli, *Euclidean distance matrices: Essential theory, algorithms, and applications*, IEEE Signal Process. Mag., 32 (2015), pp. 12–30.
[24] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, *Acoustic echoes reveal room shape*, Proc. Natl. Acad. Sci. USA, 110 (2013), pp. 12186–12191.
[25] T. Du, S. Qu, Q. Guo, and L. Zhu, *A simple efficient anchor-free node localization algorithm for wireless sensor networks*, Internat. J. Distributed Sensor Networks, 13 (2017), https://doi.org/10.1177/1550147717705784.
[26] P. Duxbury, L. Granlund, S. Gujarathi, P. Juhas, and S. Billinge, *The unassigned distance geometry problem*, Discrete Appl. Math., 204 (2016), pp. 117–132.
[27] X. Feng, P. J. E. Verdegem, Y. K. Lee, D. Sandström, M. Edén, P. Bovee-Geurts, W. J. de Grip, J. Lugtenburg, H. J. M. de Groot, and M. H. Levitt, *Direct determination of a molecular torsional angle in the membrane protein rhodopsin by solid-state NMR*, J. Amer. Chem. Soc., 119 (1997), pp. 6853–6857.
[28] B. P. Flanagan and K. L. Bell, *Array self-calibration with large sensor position errors*, Signal Process., 81 (2001), pp. 2201–2214.
[29] J. Frey, *A ranking method based on minimizing the number of in-sample errors*, Amer. Statist., 59 (2005), pp. 207–216.
[30] N. Gaffke and R. Mathar, *A cyclic projection algorithm via duality*, Metrika, 36 (1989), pp. 29–54.
[31] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, *A consolidated perspective on multimicrophone speech enhancement and source separation*, IEEE/ACM Trans. Audio Speech Language Process., 25 (2017), pp. 692–730.
[32] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens, *Auto-localization in ad-hoc microphone arrays*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Washington, DC, 2013, pp. 106–110.
[33] S. Gerschgorin, *Über die Abgrenzung der Eigenwerte einer Matrix*, Izv. Akad. Nauk. USSR. Otd. Fiz-Mat. Nauk, 7 (1931), pp. 749–754.
[34] M. D. Gillette and H. F. Silverman, *A linear closed-form algorithm for source localization from time-differences of arrival*, IEEE Signal Process. Lett., 15 (2008), pp. 1–4.
[35] W. Glunt, T. Hayden, and M. Raydan, *Molecular conformations from distance matrices*, J. Comput. Chem., 14 (1993), pp. 114–120.
[36] A. Y. Govan, A. N. Langville, and C. D. Meyer, *Offense-defense approach to ranking team sports*, J. Quant. Anal. Sports, 5 (2009), 4.
[37] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, et al., *Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs*, JAMA, 316 (2016), pp. 2402–2410.
[38] P. Güntert, *Automated NMR structure calculation with CYANA*, in Protein NMR Techniques, Springer, New York, 2004, pp. 353–378.
[39] P. Güntert, C. Mumenthaler, and K. Wüthrich, *Torsion angle dynamics for NMR structure calculation with the new program Dyana*, J. Molecular Biol., 273 (1997), pp. 283–298.
[40] L. Guttman, *A general nonmetric technique for finding the smallest coordinate space for a configuration of points*, Psychometrika, 33 (1968), pp. 469–506.
[41] K. M. Hall, *An r-dimensional quadratic placement algorithm*, Management Sci., 17 (1970), pp. 219–229.
[42] G. W. Hart, *Multidimensional Analysis: Algebras and Systems for Science and Engineering*, Springer-Verlag, New York, 1995.
[43] T. F. Havel, *Distance geometry: Theory, algorithms and chemical applications*, in Encyclopedia of Computational Chemistry, John Wiley & Sons, New York, 1998, pp. 723–742.
[44] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 2012.

[45] T. Jech, *The ranking of incomplete tournaments: A mathematician's guide to popular sports*, Amer. Math. Monthly, 90 (1983), pp. 246–266.
[46] C. R. Johnson, *Inverse M-matrices*, Linear Algebra Appl., 47 (1982), pp. 195–216.
[47] J. P. Keener, *The Perron–Frobenius theorem and the ranking of football teams*, SIAM Rev., 35 (1993), pp. 80–93, https://doi.org/10.1137/1035004.
[48] D. Kihara, H. Chen, and Y. David Yang, *Quality assessment of protein structure models*, Current Protein Peptide Sci., 10 (2009), pp. 216–28.
[49] J. B. Kruskal, *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis*, Psychometrika, 29 (1964), pp. 1–27.
[50] Y. Kuang, E. Ask, S. Burgess, and K. Åström, *Understanding TOA and TDOA network calibration using far field approximation as initial estimate*, in Proceedings of the International Conference on Pattern Recognition Applications and Methods, Algarve, Portugal, 2012, pp. 590–596.
[51] Y. Kuang, S. Burgess, A. Torstensson, and K. Åström, *A complete characterization and solution to the microphone position self-calibration problem*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Washington, DC, 2013, pp. 3875–3879.
[52] P. Kułakowski, J. Vales-Alonso, E. Egea-López, W. Ludwin, and J. García-Haro, *Angle-of-arrival localization based on antenna arrays for wireless sensor networks*, Comput. Electrical Engrg., 36 (2010), pp. 1181–1186.
[53] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, *Euclidean distance geometry and applications*, SIAM Rev., 56 (2014), pp. 3–69, https://doi.org/10.1137/120875909.
[54] D. Macagnano and G. T. F. De Abreu, *Algebraic approach for robust localization with heterogeneous information*, IEEE Trans. Wireless Commun., 12 (2013), pp. 5334–5345.
[55] D. Malioutov, M. Cetin, and A. Willsky, *A sparse signal reconstruction perspective for source localization with sensor arrays*, IEEE Trans. Signal Process., 53 (2005), pp. 3010–3022.
[56] K. Massey, *Statistical Models Applied to the Rating of Sports Teams*, Bluefield College, Bluefield, VA, 1997.
[57] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, *Fisher discriminant analysis with kernels*, in Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop, IEEE, Washington, DC, 1999, pp. 41–48.
[58] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, in Advances in Neural Information Processing Systems, NeurIPS, San Diego, CA, 2013, pp. 3111–3119.
[59] A. Mucherino, C. Lavor, and L. Liberti, *The discretizable distance geometry problem*, Optim. Lett., 6 (2012), pp. 1671–1686.
[60] P. Nicolas and G. Vezzosi, *Localization of Far-Field Sources with an Array of Unknown Geometry*, Springer, Dordrecht, The Netherlands, 1989, pp. 503–509.
[61] N. Ono, H. Kohno, N. Ito, and S. Sagayama, *Blind alignment of asynchronously recorded signals for distributed microphone array*, in Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), IEEE, Washington, DC, 2009, pp. 161–164.
[62] B. Osting, C. Brune, and S. J. Osher, *Optimal data collection for informative rankings expose well-connected graphs*, J. Mach. Learn. Res., 15 (2014), pp. 2981–3012.
[63] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Softw., 8 (1982), pp. 43–71.
[64] H. Pan, R. Scheibler, E. Bezzam, I. Dokmanic, and M. Vetterli, *FRIDA: FRI-based DOA estimation for arbitrary array layouts*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Washington, DC, 2017, pp. 3186–3190.
[65] R. Parhizkar, *Euclidean Distance Matrices: Properties, Algorithms and Applications*, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2013.
[66] J. Parsons, J. B. Holmes, J. M. Rojas, J. Tsai, and C. E. M. Strauss, *Practical conversion from torsion space to Cartesian space for in silico protein synthesis*, J. Comput. Chem., 26 (2005), pp. 1063–1068.
[67] R. Peng and M. L. Sichitiu, *Angle of arrival localization for wireless sensor networks*, in Proceedings of the 3rd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE, Washington, DC, 2006, pp. 374–382.
[68] G. Poole and T. Boullion, *A survey on M-matrices*, SIAM Rev., 16 (1974), pp. 419–427, https://doi.org/10.1137/1016079.

[69] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003, https://doi.org/10.1137/1.9780898718003.

[70] R. Scheibler, J. Azcarreta, R. Beuchat, and C. Ferry, *Pyramic: Full stack open microphone array architecture and dataset*, in Proceedings of the 16th International Workshop on Acoustic Signal Enhancement (IWAENC), Tokyo, Japan, 2018, pp. 226–230.

[71] R. O. Schmidt, *Multiple emitter location and signal parameter estimation*, IEEE Trans. Antennas and Propagation., 34 (1986), pp. 276–280.

[72] B. Schölkopf, A. Smola, and K.-R. Müller, *Nonlinear component analysis as a kernel eigenvalue problem*, Neural Comput., 10 (1998), pp. 1299–1319.

[73] S. Shiraishi, T. Obata, and M. Daigo, *Properties of a positive reciprocal matrix and their application to AHP*, J. Oper. Res. Soc. Japan, 41 (1998), pp. 404–414.

[74] D. A. Spielman and S.-H. Teng, *Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 835–885, https://doi.org/10.1137/090771430.

[75] C. H. Suh and C. W. Radcliffe, *Kinematics and Mechanisms Design*, Wiley, New York, 1987.

[76] S. Thrun, *Affine structure from sound*, in Advances in Neural Information Processing Systems 18, NeurIPS, San Diego, CA, 2006, p. 1353.

[77] D. S. W. Ting, C. Y.-L. Cheung, G. Lim, G. S. W. Tan, N. D. Quang, A. Gan, H. Hamzah, R. Garcia-Franco, I. Y. San Yeo, S. Y. Lee, et al., *Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes*, JAMA, 318 (2017), pp. 2211–2223.

[78] P. D. Turney and P. Pantel, *From frequency to meaning: Vector space models of semantics*, J. Artificial Intell. Res., 37 (2010), pp. 141–188.

[79] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of Signal Processing*, Cambridge University Press, Cambridge, UK, 2014.

[80] J. Wendeberg, F. Höflinger, C. Schindelhauer, and L. Reindl, *Calibration-free TDOA self-localisation*, J. Location Based Services, 7 (2013), pp. 121–144.