Chapter 10
ASAP: An Eigenvector Synchronization Algorithm for the Graph Realization Problem

Mihai Cucuringu

Abstract We review a recent algorithm for localization of points in Euclidean space from a sparse and noisy subset of their pairwise distances. Our approach starts by extracting and embedding uniquely realizable subsets of neighboring sensors called patches. In the noise-free case, each patch agrees with its global positioning up to an unknown rigid motion of translation, rotation, and possibly reflection. The reflections and rotations are estimated using the recently developed eigenvector synchronization algorithm, while the translations are estimated by solving an overdetermined linear system. In other words, to every patch, there corresponds an element of the Euclidean group Euc(3) of rigid transformations in $\mathbb{R}^3$, and the goal is to estimate the group elements that will properly align all the patches in a globally consistent way. The algorithm is scalable as the number of nodes increases, and can be implemented in a distributed fashion. Extensive numerical experiments show that it compares favorably to other existing algorithms in terms of robustness to noise, sparse connectivity and running time.

Keywords Graph realization problem • Sensor networks • Molecule problem • Distance geometry • Eigenvectors • Synchronization • Rigidity theory • Spectral graph theory

10.1 Introduction

The graph realization problem has attracted significant attention in recent years, especially in the context of localization of sensor networks and three-dimensional structuring of molecules [30, 31]. The problem falls naturally under the large

M. Cucuringu (✉)
PACM, Princeton University, Fine Hall, Washington Road, Princeton, NJ 08544-1000, USA
e-mail: mcucurin@math.princeton.edu

umbrella of distance geometry problems and has received growing interest from researchers across a variety of fields, including computer science, engineering, and mathematics. In this chapter we review a recently proposed two dimensional sensor network localization (SNL) algorithm introduced in [16]. Very recently, we have extended our approach to three dimensions and have added several improvements to the algorithm specific to the molecule problem from structural biology [17].

Given a set of $|V| = n$ nodes and $|E| = m$ edges, defining the graph $G = (V,E)$, together with a distance measurement associated with each edge, the graph realization problem is to assign to each vertex coordinates in $\mathbb{R}^d$ such that the Euclidean distance between any two adjacent nodes matches the prescribed distance associated to that edge. In other words, for any edge $(i, j) \in E$ of the measurement graph G, one is given the distance $d_{ij} = d_{ji}$, with the goal of finding a d-dimensional embedding $p_1, p_2, \ldots, p_n \in \mathbb{R}^d$ such that $\|p_i - p_j\| = d_{ij}$, for all $(i, j) \in E$. In this chapter, we focus on the two-dimensional case, although the approach is applicable to higher dimensions $d > 2$ as well. The graph realization problem comes up naturally in a variety of settings such as wireless sensor networks [9, 45], structural biology [24], environmental monitoring [1], and multidimensional scaling (MDS) [15]. In such real-world applications, it is typically the case that the available distances d_{ij} between the nodes are very noisy, with $d_{ij} = \|p_i - p_j\| + \varepsilon_{ij}$ where ε_{ij} represents the added noise, and the goal is to find an embedding that matches all available distances d_{ij} as best as possible. Classical multidimensional scaling successfully solves the localization problem as long as all $n(n-1)/2$ pairwise distances are available, which unfortunately is rarely the case in practical applications.

We assume that the graph realization problem has a unique solution in other words, the underlying graph is globally rigid, and note that applying a rigid transformation (composition of rotation, translation, and possibly reflection) to a graph realization results in another graph realization, as rigid transformations preserve distances. Whenever an embedding is possible, it is unique (up to rigid transformations) only if there are enough distance constraints, in which case the graph is said to be globally rigid (see, e.g., [23]). From a computational perspective, the graph realization problem has been shown to be very difficult. Saxe has shown it is strongly NP-complete in one dimension and strongly NP-hard for higher dimensions [35, 47]. A popular model for the SNL problem is that of a disc graph model, where two sensors communicate with each other if and only if they are within sensing radius ρ of each other, i.e., $(i, j) \in E \iff d_{ij} \leq \rho$. The SNL problem is NP-hard also under the disc graph model [4]. Despite its difficulty, the problem has received a great deal of attention in the networking and distributed computation communities, and numerous heuristic algorithms exist that approximate its solution. In the context of sensor networks [2, 4, 5, 27], there are many algorithms that solve the graph realization problem, and they include methods such as global optimization [12], semidefinite programming (SDP) [9–11, 42, 43, 49], and local to global approaches [29, 32, 36, 37, 48], some of which we briefly review in Sect. 10.2.

The algorithm we review in this chapter follows a local to global divide-and-conquer approach, integrating local distance information into a global structure determination. Locally, we identify for every sensor, the globally rigid subgraphs of its 1-hop neighborhood, which we call patches. Once the 1-hop neighborhood has been decomposed into patches, we separately localize each such patch in a coordinate system of its own using either the stress minimization method of [21] or SDP. When all available distances are noiseless, the computed coordinates of the sensors in each patch will agree with the ground truth solution up to some unknown rigid motion, i.e., a combination of a translation, rotation, and possibly reflection, which is precisely what out proposed algorithm is estimating. In other words, to every existing patch, there corresponds an element of the Euclidean group Euc(2) of rigid transformations in the plane, and the goal is to estimate these unknown group elements that will properly align all the patches in a globally consistent framework. In this process, the only available information we make use of is the set of pairwise alignments between any two patches that overlap in sufficiently many nodes. In other words, by finding the optimal alignment between pairs of patches whose intersection is large enough, we obtain measurements for the ratios of the unknown corresponding group elements. Finding group elements from noisy measurements of their ratios is also known as the synchronization problem [20, 28], which we discuss in Sect. 10.6. Intuitively, we use the eigenvector method for the compact part of the group, when we synchronize over the groups $\mathbb{Z}_2$ and SO(2) to recover the reflections and rotations, and solve by least squares an overdetermined linear system in $\mathbb{R}^2$ for the translations. We consider the performance of our algorithm with respect to several criteria, including robustness to noise and sparsity of the distance measurements, and scalability to large networks with tens or hundreds of thousands of nodes.

This chapter is organized as follows: Section 10.2 is a survey of existing methods for solving the two-dimensional graph realization problem, with a focus on localization of planar sensor networks. Section 10.3 gives an overview of the 2D-as-synchronized-as-possible (ASAP) algorithm described in this chapter. In Sect. 10.4, we motivate our divide-and-conquer approach and explain how to break up the initial measurement graph and how to localize the resulting patches. Section 10.5 explains the synchronization algorithm that aligns all patches in a globally consistent structure. In Sect. 10.6 we give a brief self-contained introduction to the group synchronization problem and the references therein. In Sect. 10.7, we report on numerical simulations where we tested the performance of 2D-ASAP in comparison to existing state-of-the-art algorithms. Finally, Sect. 10.8 is a summary and discussion of the extension of our algorithm to the molecule problem in structural biology.

10.2 Related Work

Numerous algorithms across different communities have been proposed for the SNL problem, with the goal of finding an approximate embedding $p_1, \dots, p_n \in \mathbb{R}^2$ that preserves the measured (noisy) distances $d_{ij}, (i, j) \in E$ as best as possible.

Approaches coming from the SDP community [8–11, 49] propose minimizing a variety of error functions, such as

$$f(p_1,\ldots,p_n) = \sum_{(i,j)\in E} \left(\|p_i - p_j\|^2 - d_{ij}^2\right)^2 \tag{10.1}$$

$$g(p_1,\ldots,p_n) = \sum_{(i,j)\in E} \left|\| p_i - p_j \|^2 - d_{ij}^2\right| \tag{10.2}$$

$$Stress(p_1,\ldots p_n) = \sum_{(i,j)\in E} (\| p_i - p_j \| - d_{ij})^2 . \tag{10.3}$$

Unfortunately, all the above functions are not convex over the constraint set, and the search for the global minimum is prone to getting stuck at a local minima. Their relaxations to an SDP [9] are computationally expensive and not very robust to noise, as the solutions belong to a higher dimensional Euclidean space, and the projection to the plane often results in large errors for the estimation of the coordinates. The stress majorization algorithm (also known as SMACOF [12]), was originally introduced by Leeuw [18] as a variant of the gradient descent approach for minimizing the stress function in Eq. (10.3).

Maximum variance unfolding (MVU) is a non-linear dimensionality reduction algorithm proposed by Weinberger et al.[46], which became very popular within the machine-learning community. The algorithm produces a low-dimensional representation of the data by maximizing the variance of its embedding while preserving the original local distance constraints. MVU builds on the SDP approach and addresses the issue of the possibly high-dimensional solution to the SDP problem by maximizing the variance of the embedding (also known as the maximum trace heuristic). The main contribution of the FAST-MVU algorithm in [46] is the approximation of the x and y coordinate vectors of the sensors by just the first few (e.g., 10) low-oscillatory eigenvectors of the graph Laplacian. This allows one to replace the original and possibly large-scale SDP by a much smaller SDP, which leads to a significant reduction in running time. The locally rigid embedding (LRE) algorithm [37] is reminiscent of the locally linear embedding (LLE) [34] technique used in machine learning for dimensionality reduction. LRE tries to preserve, in a global coordinate system, the local affine relationships present within patches. Each sensor contributes with a linear equation relating its location to those of its neighboring nodes, thus altogether setting up a global linear system. LRE builds up a specially designed sparse matrix whose eigenvectors give an embedding of all sensors, from which a global affine transformation must be removed.

The recent as-rigid-as-possible (ARAP) algorithm in [48] is along the lines of PATCHWORK [29] and LRE, and starts off by localizing small patches in a similar manner, but instead of finding a global embedding via affine mappings, they use rigid mappings. Again, the patch overlaps impose constraints on the mappings however, the usage of rigid mappings has the advantage of better preserving the local relationships between patches. This comes at the price of resulting in a non linear optimization problem, which is solved efficiently using a two-phase alternating least-squares method. The initial guess required by the nonlinear optimization is obtained by as-affine-as-possible (AAAP), an improved version of the LRE and PATCHWORK algorithms.

Very recently, Javanmard and Montanari [26] proposed a localization algorithm based on SDP, for which they provide a theoretical analysis in terms of robustness to noise for the random geometric graph model and uniformly bounded adversarial noise. For noiseless data, they provide a lower bound for the radius beyond which the algorithm is guaranteed to recover the original solution, up to a rigid transformation. On a related note, we also report on recent and ongoing work of Ozyesil and Singer on SyncContract [33], an optimization algorithm able to synchronize over the non-compact special Euclidean group $\mathrm{SE}(k)$ by solving an analogous problem on a compact group, of which $\mathrm{SE}(k)$ is a Lie group contraction. They provide experimental results for synthetic data and for the SNL problem and a robustness analysis of their algorithm for the complete and the sparse Erdös-Rényi graph models.

10.3 Overview of the 2D-ASAP Algorithm

This section provides an overview of the 2D-ASAP algorithm reviewed in this chapter. We follow a divide-and-conquer approach that breaks up the large graph into many smaller overlapping subgraphs, that we call patches, and "stitches" them together concurrently and consistently in a global coordinate system with the purpose of localizing the entire initial measurement graph. To avoid foldovers in the final solution, each such patch needs to be globally rigid and the entire measurement graphs needs to be globally rigid as well.

We break up the initial graph into patches in the following way. For every node i we let $V(i) = \{j : (i,j) \in E\} \cup \{i\}$ the set of its 1-hop neighbors together with the node itself, and $E(i) = \{(i,j) \in E | \{i,j\} \in V(i)\}$, and denote by $G(i) = (V(i), E(i))$ its subgraph of 1-hop neighbors. If $G(i)$ is a globally rigid graph, we embed it in $\mathbb{R}^2$, otherwise we break it into maximally globally rigid subgraphs that we call patches, and embed each patch in $\mathbb{R}^2$. The embedding of every patch in $\mathbb{R}^2$ is given in its own local frame. We defer to Sect. 10.4 the specific details of breaking up the measured graph into smaller maximally globally rigid subgraphs. Let N denote the number of patches obtained in the above decomposition of the measurement graph, and note that it may be different from n, the number of nodes in G, since the neighborhood graph of a node may contribute several patches or none.

For the embedding of local patches we usually use the stress majorization algorithm as described in [21]. Once each patch is embedded in its own coordinate system, one must find the reflections, rotations, and translations that will stitch all patches together in a consistent manner, a process to which we refer as *synchronization*. To every embedded patch P_i there corresponds an element $e_i \in$ Euc(2), where Euc(2) is the Euclidean group of rigid motions in the plane, i.e., reflections, rotations, and translations. The rigid motion e_i moves patch P_i to its correct position with respect to the global coordinate system. Our goal is to estimate simultaneously the rigid motions $e_1,\dots,e_N$ (up to a global rigid motion) that will properly align all the patches in a globally consistent way. To achieve this goal, we first estimate the alignment between any pair of patches P_i and P_j that have enough nodes in common. We describe one such alignment method in Sect. 10.4 and refer the reader to Sect. 6 of [16] for additional robust alignment methods. The alignment of patches P_i and P_j provides a (usually noisy) measurement for the ratio $e_i e_j^{-1}$ in Euc(2). We solve the resulting synchronization problem in a globally consistent manner, such that information from local alignments propagates to pairs of nonoverlapping patches. Ideally, we would like to be able to solve the synchronization problem over Euc(2); however, the non-compactness of the group makes the problem significantly harder. Only very recently, the authors of [33] introduced several optimization-based algorithms that are able to synchronize over the non-compact special Euclidean group $\mathrm{SE}(k)$ by solving an analogous problem on a compact group.

As an alternative, we replace the synchronization problem over Euc(2) by three different consecutive synchronization problems. In the first one, we find the reflections of all the patches using the eigenvector synchronization algorithm over the group $\mathbb{Z}_2$. After we have estimated the reflections, we use the same eigenvector synchronization method, but this time over the group SO(2) to estimate the rotations of all patches. Once both reflections and rotations have been computed, we estimate the translations by solving an overdetermined linear system. To summarize, we simultaneously integrate all the available local information into a global coordinate system over three steps, using the eigenvector synchronization algorithm and the least-squares method over the isometries of the Euclidean plane. As we shall see, the main advantage of the eigenvector method is that it can recover the reflections and rotations even when many of the pairwise alignments are incorrect. A complete summary of the 2D-ASAP algorithm is given in Table 10.1.

10.4 Finding and Localizing Globally Rigid Patches

Next we describe how to identify and localize patches, a crucial step of our divide-and-conquer algorithm. Unlike most other localization algorithms, we choose to build patches that are globally rigid, and provide an efficient and theoretically motivated method for doing so. Previous localization algorithms that use a local to global approach, such as PATCHWORK, LRE, and ARAP, simply define patches by associating with every node i its entire 1-hop neighborhood $G(i)$, which usually leads to patches which are not globally rigid and have more than one possible realization in the plane. Therefore, whenever $G(i)$ is not globally rigid, we find its maximally globally rigid components, which we call patches. Note that the number of resulting patches can be 0, 1, or greater than 1. The novelty of our approach is that breaking up the 1-hop neighborhood subgraph $G(i)$ is much easier than breaking up a general graph, by utilizing recent results of [14] about the global rigidity property of cone graphs.

Table 10.1 Overview of the 2D-ASAP algorithm

Input	$G=(V,E)$, $\lvert V\rvert=n$, $\lvert E\rvert=m$, d_{ij} for $(i,j)\in E$
Pre-processing step	1. Break the measurement graph G into N globally rigid patches $P_1,\ldots,P_N$ 2. Embed each patch P_i separately using the embedding method of choice (e.g., stress majorization or SDP)
Step 1 Estimating reflections	1. Align all pairs of patches (P_i,P_j) that have enough nodes in common 2. Estimate their relative reflection $z_{ij}\in\{-1,+1\}$ 3. Build a sparse $N\times N$ symmetric matrix $Z=(z_{ij})$ as defined in Eq. (10.4) 4. Define $\mathscr{Z}=D^{-1}Z$, where D is a diagonal matrix with $D_{ii}=deg(i)$ 5. Compute the top eigenvector $v_1^{\mathscr{Z}}$ of $\mathscr{Z}$ which satisfies $\mathscr{Z}v_1^{\mathscr{Z}}=\lambda_1^{\mathscr{Z}}v_1^{\mathscr{Z}}$ 6. Estimate the global reflection of patch P_i by $\hat{z}_i=\operatorname{sign}(v_1^{\mathscr{Z}}(i))=\frac{v_1^{\mathscr{Z}}(i)}{\lvert v_1^{\mathscr{Z}}(i)\rvert}$ 7. Replace the embedding patch P_i with its mirrored image whenever $\hat{z}_i=-1$
Step 2 Estimating rotations	1. Align all pairs of patches (P_i,P_j) that have enough nodes in common 2. Estimate their relative rotation angle $\theta_{ij}\in[0,2\pi)$ and set $r_{ij}=e^{\iota\theta_{ij}}$ 3. Build a sparse $N\times N$ Hermitian matrix $R=(r_{ij})$ as defined in Eq. (10.5) 4. Define $\mathscr{R}=D^{-1}R$ 5. Compute the top eigenvector $v_1^{\mathscr{R}}$ of $\mathscr{R}$ corresponding to $\mathscr{R}v_1^{\mathscr{R}}=\lambda_1^{\mathscr{R}}v_1^{\mathscr{R}}$ 6. Estimate the global rotation angle $\hat{\theta}_i$ of patch P_i using $e^{\iota\hat{\theta}_i}=\frac{v_1^{\mathscr{R}}(i)}{\lvert v_1^{\mathscr{R}}(i)\rvert}$ 7. Rotate the embedding of patch P_i by the angle θ_i
Step 3 Estimating translations	1. Build an $m\times n$ overdetermined system of linear equations 2. Include the anchors information (if available) into the linear system 3. Compute the least squares solution for the x-axis and y-axis coordinates
Output	Estimated coordinates $\hat{p}_1,\ldots,\hat{p}_n$

We call a *star graph* a graph which contains at least one vertex that is connected to all remaining nodes. Note that for each node i, the local graph $G(i)$ composed of the central node i and all its neighbors takes the form of a star graph. We make use of this structural property of the graph, and propose a simple efficient algorithm that break up non-globally rigid star graph into smaller globally rigid star subgraphs, each of which is of maximal size.

Proposition 10.1. *A star graph is generically globally rigid in $\mathbb{R}^2$ iff it is three-vertex-connected.*

In light of Proposition 10.1 [16], we propose the following simple algorithm for breaking up a star graph into maximally globally rigid components. We start by removing all vertices of degree one, since no globally rigid subgraph can contain such a vertex. Note that a vertex of degree two can be only be contained in a triangle, provided its two neighbors are connected. Next, we search for the (maximal) three-connected components in the graph, taking advantage of its structure as a star graph.

Once we have divided the original graph into many overlapping globally rigid patches, the next step is to find a two-dimensional embedding of each one of them. Localizing a small globally rigid subgraph is significantly easier in terms of speed and accuracy than localizing the whole measurement graph. First, the size of a patch is significantly smaller than the size of the whole network. Also, another advantage of embedding locally is that we are no longer constrained to a distributed computation that can impose additional challenges due to intersensor communication. Since each node in the patch is connected to a central node, all the information can be passed on to this node which will perform the computation in a centralized manner. Finally, under the assumptions of the disc graph model, it is likely that 1-hop neighbors of the central node will also be interconnected, rendering a relatively high density of edges for the patches.

After extensively experimenting with different localization algorithms, our method of choice for embedding the patches was the three-stage procedure described in [21], due to its relatively low running time and its robustness to noise for small patches. When used for small patches (e.g., of size 20–30) rather than the entire network, the stress minimization is more reliable and less sensitive to local minima. Compared to an anchor-free SDP localization algorithm like SNL-SDP,[1] it produces similar results in terms of the localization error, but with lower running times. To the best of our knowledge, the SDP-based approaches (in particular those of [9–11, 42, 43, 49]) have not been analyzed in the context of the disc graph model, and the SDP localization theory is built only on the known distances, without any additional lower and upper bounds that can be inferred from the disc graph assumption. Note that we restrict the size of the patches to some maximal prescribed size to avoid inaccurate patch embeddings.

A crucial step in the synchronization algorithm is the accurate alignment of pairs of patches for building the matrices Z and R of pairwise reflections and rotations. For any two patches P_i and P_j embedded in their own coordinate system, we are interested in estimating their relative reflection rotation. Clearly, any two patches that are far apart and have no common nodes cannot be aligned thus, there must be enough overlapping nodes to make the alignment possible. The problem of aligning two labeled sets of nodes is known as the registration problem, for which a closed form solution for any dimension was proposed by [25], where the best rigid transformation between two sets of points is obtained by various matrix manipulations and eigenvalue/eigenvector decomposition. We refer the reader to Sect. 6 of [16] for a thorough discussion of several methods for aligning patches.

10.5 Synchronization over $\mathbb{Z}_2$, SO(2), and $\mathbb{R}^2$

This section details Steps 1, 2, and 3 of the 2D-ASAP algorithm. We use the eigenvector method for the group synchronization problems over the groups $\mathbb{Z}_2$ and SO(2), to estimate the reflections and rotations of the N patches. In the last step, we synchronize over $\mathbb{R}^2$ by solving an overdetermined system of linear equations to recover the translations of the patches and provide a final estimate for the sensor coordinates.

[1]We used the SNL-SDP code of [44].

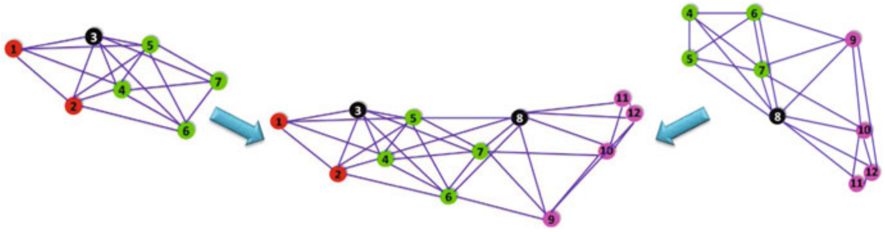

Fig. 10.1 Optimal alignment of two patches that overlap in four nodes. The alignment provides a measurement for the ratio of the two group elements in Euc(2). In this example we see that a reflection was required to properly align the patches

10.5.1 Step 1: Synchronization over $\mathbb{Z}_2$ to Estimate Reflections

In the first step of the algorithm, we identify which patches need to be reflected with respect to the global coordinate system. We denote the reflection of patch P_i by $z_i \in \{-1, 1\}$, a -1, indicating that the patch requires a reflection, and $+1$ otherwise. Note that all reflections are defined up to a global reflection (global sign). The alignment of every pair of patches P_i and P_j whose intersection is sufficiently large, provides a measurement z_{ij} for the ratio $z_i z_j^{-1}$ (in the case of $\mathbb{Z}_2$ this is simply the product $z_i z_j$, since an element is its own inverse). When the initial distance measurements are noisy (and hence the patch embeddings) many ratio measurements can be corrupted, i.e., have their sign flipped. We denote by $G^P = (V^P, E^P)$ the patch graph whose vertices V^P are the patches $P_1, \dots, P_N$, and two patches P_i and P_j are adjacent, $(P_i, P_j) \in E^P$, iff they have enough[2] vertices in common to be aligned such that the ratio $z_i z_j^{-1}$ can be estimated (Fig. 10.1).

We solve this synchronization problem over $\mathbb{Z}_2$ using the eigenvector method, which starts off by building the following $N \times N$ sparse symmetric matrix $Z = (z_{ij})$:

$$z_{ij} = \begin{cases} 1 & \text{aligning } P_i \text{ with } P_j \text{ did not require reflection} \\ -1 & \text{aligning } P_i \text{ with } P_j \text{ required reflection of one of them} \\ 0 & (i,j) \notin E^P \; (P_i \text{ and } P_j \text{ cannot be aligned}) \end{cases} \tag{10.4}$$

Prior to computing the top eigenvector of the matrix Z, as done in [38], we choose the following normalization that increases the robustness to noise and numerical stability. Let D be an $N \times N$ diagonal matrix,[3] whose entries are given by $D_{ii} = \sum_{j=1}^{N} |z_{ij}|$. In other words, $D_{ii} = deg(i)$, where $deg(i)$ is the node degree of patch P_i in G^P, i.e., the number of other patches that can be aligned with it. We define the

[2]For example three common vertices, although the precise definition of "enough" will be given later.

[3]The diagonal matrix D should not be confused with the partial distance matrix.

matrix $\mathscr{Z}$ as $\mathscr{Z} = D^{-1}Z$, and note that, although not necessarily symmetric, it is similar to the symmetric matrix $D^{-1/2}ZD^{-1/2}$ through

$$\mathscr{Z} = D^{-1/2}(D^{-1/2}ZD^{-1/2})D^{1/2}.$$

Therefore, the matrix $\mathscr{Z}$ has N real eigenvalues $\lambda_1^{\mathscr{Z}} > \lambda_2^{\mathscr{Z}} \geq \cdots \geq \lambda_N^{\mathscr{Z}}$ and N orthonormal eigenvectors $v_1^{\mathscr{Z}}, \ldots, v_N^{\mathscr{Z}}$, satisfying $\mathscr{Z}v_i^{\mathscr{Z}} = \lambda_i^{\mathscr{Z}} v_i^{\mathscr{Z}}$. In the eigenvector method, we compute the top eigenvector $v_1^{\mathscr{Z}} \in \mathbb{R}^N$ of $\mathscr{Z}$ and use it to obtain estimators $\hat{z}_1, \ldots, \hat{z}_N$ for the reflections of the patches, in the following way: $\hat{z}_i = \operatorname{sign}(v_1^{\mathscr{Z}}(i)) = \frac{v_1^{\mathscr{Z}}(i)}{|v_1^{\mathscr{Z}}(i)|}$, $i = 1, 2, \ldots, N$. After estimating the reflection of all patches (up to a global sign), we replace the embedding of patch P_i by its mirrored image whenever $\hat{z}_i = -1$.

10.5.2 Step 2: Synchronization over SO(2) to Estimate Rotations

After having estimated the appropriate reflections, next we estimate the rotations of all patches. To each patch we associate an element $r_i \in \text{SO}(2)$, $i = 1, \ldots, N$ that we represent as a point on the unit circle in the complex plane $r_i = e^{\imath\theta_i} = \cos\theta_i + \imath\sin\theta_i$. We repeat the alignment process from Step 1 to estimate the angle θ_{ij} between two overlapping patches, i.e., the angle by which one needs to rotate patch P_i to align it with patch P_j. When the aligned patches contain corrupted distance measurements, θ_{ij} is a noisy measurement of their offset $\theta_i - \theta_j \mod 2\pi$. Following a similar approach to Step 1, we build the $N \times N$ sparse symmetric matrix $R = (r_{ij})$ whose elements are either 0 or points on the unit circle in the complex plane:

$$r_{ij} = \begin{cases} e^{\imath\theta_{ij}} & \text{if } (i,j) \in E^P \\ 0 & \text{if } (i,j) \notin E^P \end{cases}. \tag{10.5}$$

Since $\theta_{ij} = -\theta_{ji} \mod 2\pi$, it follows that R is a Hermitian matrix, i.e., $R_{ij} = \bar{R}_{ji}$, where for any complex number $w = a + \imath b$ we denote by $\bar{w} = a - \imath b$ its complex conjugate. As in Step 1, we choose to normalize R using the diagonal matrix D, whose diagonal elements are also given by $D_{ii} = \sum_{j=1}^{N} |r_{ij}|$. Next, we define the matrix $\mathscr{R} = D^{-1}R$, which is similar to the Hermitian matrix $D^{-1/2}RD^{-1/2}$ through

$$\mathscr{R} = D^{-1/2}(D^{-1/2}RD^{-1/2})D^{1/2}.$$

We define the estimated rotation angles (up to an additive phase) $\hat{\theta}_1, \ldots, \hat{\theta}_N$ and their corresponding elements in SO(2), $\hat{r}_1, \ldots, \hat{r}_N$ using the top eigenvector $v_1^{\mathscr{R}}$ as

$$\hat{r}_i = e^{\imath\hat{\theta}_i} = \frac{v_1^R(i)}{|v_1^R(i)|}, \quad i = 1, 2, \ldots, N. \tag{10.6}$$

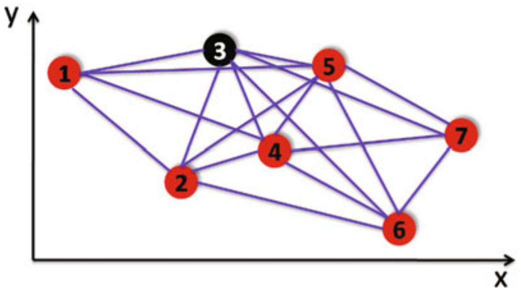

Fig. 10.2 Embedding patch P_k in its local coordinate frame after it was appropriately reflected and rotated. In the noise-free case, the coordinates $p_i^{(k)} = (x_i^{(k)}, y_i^{(k)})^T$ agree with the global positioning $p_i = (x_i, y_i)^T$ up to some translation $t^{(k)}$ (unique to all i in V_k)

10.5.3 Step 3: Synchronization over $\mathbb{R}^d$ to Estimate Translations

In the last step of the 2D-ASAP algorithm we compute the global translations of all patches and obtain the final estimates for the coordinates. For each patch P_k, we denote by $G_k = (V_k, E_k)$[4] the graph associated to patch P_k, where V_k is the set of nodes in P_k, and E_k is the set of edges induced by V_k in the measurement graph $G = (V, E)$. We denote by $p_i^{(k)} = (x_i^{(k)}, y_i^{(k)})^T$ the known local frame coordinates of node $i \in V_k$ in the embedding of patch P_k (see Fig. 10.2). Since each patch P_k has been properly reflected and rotated so that the local frame coordinates are consistent with the global coordinates , up to a translation $t^{(k)} \in \mathbb{R}^2$, in the noise-free case, it holds that

$$p_i = p_i^{(k)} + t^{(k)}, \quad i \in V_k, \quad k = 1, \ldots, N. \tag{10.7}$$

We estimate the global coordinates $p_1, .., p_n$ as the least-squares solution to the overdetermined system of linear equation (10.7), while ignoring the by-product translations $t^{(1)}, \ldots, t^{(N)}$. In practice, we write a linear system for the displacement vectors $p_i - p_j$ for which the translations have been eliminated. From Eq. (10.7) it follows that each edge $(i, j) \in E_k$ contributes a linear equation of the form

$$p_i - p_j = p_i^{(k)} - p_j^{(k)}, \quad (i, j) \in E_k, \quad k = 1, \ldots, N. \tag{10.8}$$

We separate these constraints along the x and y global coordinates of nodes i and j, and solve (independently) each of the two resulting linear systems using the ordinary linear regression.

[4]Not to be confused with $G(i) = (V(i), E(i))$ defined in the beginning of this section.

10.6 The Eigenvector Method for the Group Synchronization Problem

In general, the synchronization problem can be applied in such settings where the underlying problem exhibits a group structure and one has available noisy measurements of ratios of group elements. We have already seen in the previous sections two such instances of the group synchronization problem. The eigenvector and SDP-based methods for solving angular synchronization problem for the group SO(2) were originally introduced by Singer in [38]. There, one is asked to estimate N unknown angles $\theta_1, \dots, \theta_N \in [0, 2\pi)$ given M noisy measurements δ_{ij} of their offsets $\theta_i - \theta_j \mod 2\pi$. The difficulty of the problem is amplified on one hand by the amount of noise in the offset measurements, and on the other hand by the fact that $M << \binom{N}{2}$, i.e., only a small subset of all possible offsets are measured. In general, one may consider any group $\mathscr{G}$ other than SO(2), for which there are available noisy measurements g_{ij} of ratios between group elements

$$g_{ij} = g_i g_j^{-1}, g_i, g_j \in \mathscr{G}.$$

As long as the group $\mathscr{G}$ is compact and has a real or complex representation, one may construct a real or Hermitian matrix (which may be a matrix of matrices) where the element in the position $\{ij\}$ is the matrix representation of the measurement g_{ij} (possibly a matrix of size 1×1) or the zero matrix if there is no direct measurement for the ratio of g_i and g_j. For example, the rotation group SO(3) has a real representation using 3×3 rotation matrices, and the rotation group SO(2) has a complex representation as points on unit circle $e^{\imath\theta_i} = \cos\theta_i + \imath \sin\theta_i$. One may now make use of the top eigenvectors of this matrix to estimate the unknown group elements. Alternatively, one may use this matrix to formulate an SDP program and extract the unknown group elements. The set E of pairs $\{ij\}$ for which a ratio of group elements is available can be realized as the edge set of a graph $G^P = (V^P, E^P)$, $|V^P| = N, |E^P| = M$ with vertices corresponding to the group elements $g_1, \dots, g_N$ and edges corresponding to the available measurements $g_{ij} = g_i g_j^{-1}$. Note that we use the superscript to denote the patch graph, as introduced in the previous section. Two vertices i and j of the graph G^P are connected, i.e., $\{ij\} \in E^P$, if and only if their corresponding patches P_i and P_j have enough points in common and can be pairwise aligned.

In Sect. 4 of [16] we give an analysis of the eigenvector method for the group synchronization problem in the noiseless case, using the fact that the eigenvalues of the normalized matrices $\mathscr{Z}$ and $\mathscr{R}$ are related to the those of the discrete normalized graph Laplacian of the underlying patch graph. An analysis of the eigenvector synchronization method in the presence of noise was first explored by Singer [38], in the case of the group SO(2), and uses tools from random matrix theory that allow for a precise matrix perturbation analysis that quantifies the robustness to noise of the method under a certain random noise model. Furthermore, it provides an information theoretic analysis showing that the eigenvector method is asymptotically nearly optimal and achieves the information theoretic Shannon bound up to a multiplicative factor that depends only on the discretization error of the measurements. In very recent work, Bandeira, Singer, and Spielman [6] proved a Cheeger-type inequality via the graph connection Laplacian operator, providing a deterministic worst case performance guarantee for the synchronization problem over the group O(d) of orthogonal transformations.

The eigenvector synchronization method has proven extremely useful in a variety of applications other than the SNL problem. In particular, Hadani et al.[22, 39, 41] demonstrated its usefulness in solving the "class averaging" problem in cryo-electron microscopy [19] and showed its mathematical connection to the parallel transport and the connection Laplacian operators from differential geometry. Other applications include the 3D structure from motion problem in computer vision [3] and the analysis of high-dimensional data point clouds [40], specifically, to robustly compute Laplacian eigenmaps and diffusion maps [7, 13] that are popular methods for dimensionality reduction and spectral clustering.

10.7 Experimental Results

We have implemented our 2D-ASAP algorithm and performed numerical simulations, comparing its performance with other methods across a variety of measurement graphs. In this section we report such results for three data sets, and refer the reader to Sect. 8 of [16] for additional numerical experiments. We use multiplicative and uniform noise, meaning that to each true distance measurement $l_{ij} = \| p_i - p_j \|$, we add random independent noise ε_{ij} in the range $[-\eta l_{ij}, \eta l_{ij}]$, i.e., $d_{ij} = l_{ij} + \varepsilon_{ij}$ where $\varepsilon_{ij} \sim Uniform([-\eta l_{ij}, \eta l_{ij}])$. The percentage noise added is 100η (e.g., $\eta = 0.1$ corresponds to 10% noise).

In terms of time complexity, 2D-ASAP scales almost linearly in the size of the network, number of nodes n, and edges m. We refer the reader to Sect. 7 of [16] for a detailed complexity analysis of each step of the algorithm. We augment this theoretical analysis with the running times of numerical simulations for the localization of networks of increasing sizes $n = 10^3, 10^4, 10^5$, as detailed in Table 10.2.

Table 10.2 Running times (in seconds) of the ASAP algorithm on the SQUARE graph with $n = \{10^3, 10^4, 10^5\}$ nodes inside the unit square, $\eta = 0\%$ and $deg \approx 12, 13$

Stage \# of nodes n	1,000	10,000	100,000
Break G into patches	41	901	52,180
Embedding patches	414	4,325	37,140
Patch intersections	2	132	58,134
Build $\mathscr{Z}$	8.7	90	2,237
Compute $v_1^{\mathscr{Z}}$	0.8	13	926
Build $\mathscr{R}$	4.6	49	3,414
Compute $v_1^{\mathscr{R}}$	0.2	7	522
Step 3	6	88	4,772
Total time (s)	477	5,605	159,325

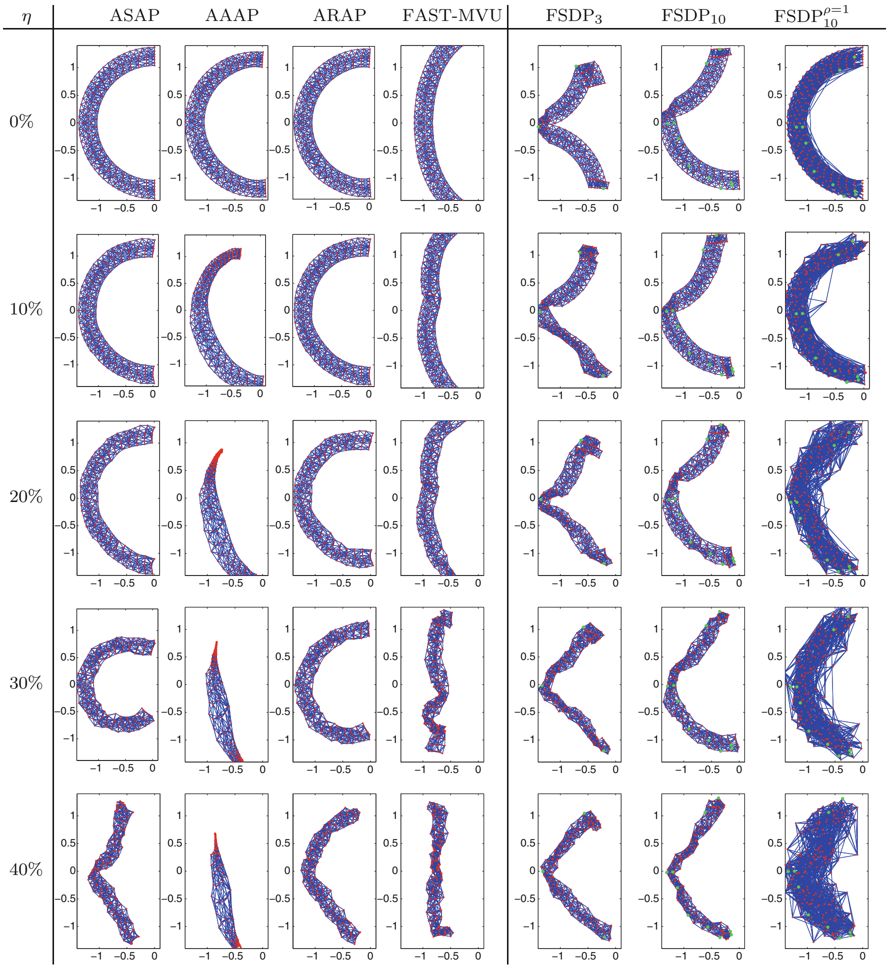

Fig. 10.3 Reconstructions of the dense C graph with $n = 200$ nodes, $\rho = 0.28$, and $\eta = 35\%, 40\%, 50\%, 60\%$, and 70%

The C-shape, graphs in Fig. 10.3 have $n = 200$ nodes, sensing radius $\rho = 0.28$, the average degrees are between 20 and 28, and the noise levels are $\eta = 35\%, 40\%, 50\%, 60\%$, and 70%. In addition to ARAP and FAST-MVU, we compare our results against the FULL-SDP algorithm [11] in three different scenarios. In the first two, we run FULL-SDP on the same measurement graph used by the other algorithms, but provide FULL-SDP with additional 3 and 10 anchors placed at random, that are not provided to the other algorithms. We choose the anchors at random from the set of all sensors. In the third scenario, we use a measurement graph of (approximately) the same average degree *deg* as the one used by the other

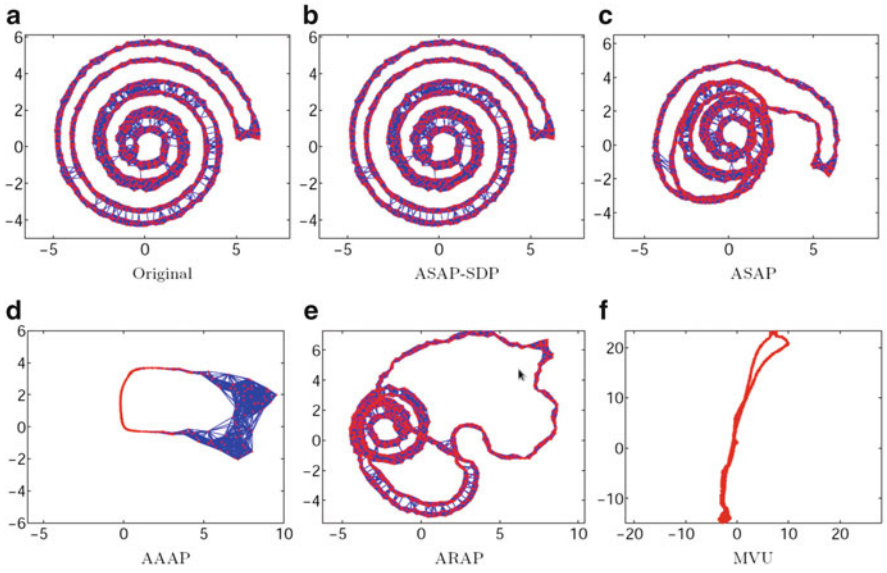

Fig. 10.4 Reconstructions of the SPIRAL graph with $n = 2259$ nodes, $\rho = 0.47$ and $\eta = 0\%$. ASAP-SDP is a version of ASAP where we used SDP for the localization of the patches, instead of SMACOF

algorithms, but allow FULL-SDP to use a much larger sensing radius $\rho = 1$, while keeping the average degree constant. These experiments show that FULL-SDP is somewhat sensitive to the sensing radius and the number of anchors used.

A second graph we report on is the SPIRAL graph shown in Fig. 10.4a. This graph contains $n = 2259$ nodes that are spread near a spiral curve that starts at the origin, and once it gets to its outermost loop it traces back towards the origin. The perturbation of the sensors from the curve ensures that the 1-hop neighborhoods are not too close to being collinear. The sensing radius for this graph is $\rho = 0.47$. Despite the fact that the measured distances are noise-free, the localizations obtained by both ASAP, AAAP, ARAP, and MVU (Fig. 10.4c, e, f) deviate from the true positioning. The failure of ASAP to find the original embedding in this noise-free case is due to a failure of the SMACOF procedure to localize a small number of patches. Although there is no noise in the distance measurements, the stress minimization algorithm sometimes converges to a local minimum, resulting in patches that are incorrectly localized. Since the topology of this graph is that of a closed curve, such bad patches lead to incorrect twists and turns in our computed embedding. Although ASAP and ARAP are using the same algorithm to localize the patches, it is clear that the incorrectly localized patches are less harmful to ASAP as they are to ARAP. Figure 10.4b shows the accurate embedding obtained by ASAP when SNL-SDP was used to localize the patches, denoted ASAP-SDP.

Finally, in order to illustrate the scaling behavior of ASAP and compare its running time to that of the other algorithms, we experimented with random graphs

Table 10.3 Comparison of the running times (in seconds) of different algorithms for the SQUARE graph with $n = \{10^3, 10^4\}$ nodes and $\eta = 0\%$

Algorithm	$n = 1{,}000$	$n = 10{,}000$
ASAP	477	5,605
AAAP	1,170	> 48 h
ARAP	1,201	> 48 h
FAST-MVU	2.7	10.8
FULLSDP$_{20}$	5,250	–

with $n = \{10^3, 10^4, 10^5\}$ nodes distributed uniformly at random in the unit square, with average degree close to 13. Table 10.2 details the running times of the various steps of the ASAP algorithm for three graphs, and Table 10.3 compares the running times of ASAP, AAAP, ARAP, FAST-MVU, and FULLSDP$_{20}$ for a random graph on $n = 10^3$ nodes.

10.8 Summary and Discussion

In this chapter we reviewed 2D-ASAP, a novel non-incremental non-iterative anchor-free algorithm for solving ab initio the SNL problem. Our extensive numerical simulations show that 2D-ASAP is very robust to high levels of noise in the measured distances and to sparse connectivity in the measurement graph. It compares favorably to some of the current state-of-the-art graph localization algorithms both in terms of robustness to noise and running time. Following a "divide and conquer" philosophy, we start with local coordinate computations based only on the 1-hop neighborhood information of individual nodes, but unlike previous incremental methods, we synchronize all such local information in a noise robust global optimization process using an efficient eigenvector computation.

In very recent work [17], we consider the three-dimensional version of the localization problem, and formulate it as a synchronization problem over Euc(3). The problem can be similarly solved in three steps: an eigenvector synchronization for the reflections over $\mathbb{Z}_2$, an eigenvector synchronization for the rotations over SO(3), and a least-squares solution for the translations in $\mathbb{R}^3$. In the second step of the algorithm, the optimal rotations between pairs of patches will be represented by 3×3 rotation matrices, and the elements of SO(3) will be obtained from the top three eigenvectors. Furthermore, we build on the approach used in 2D-ASAP to accommodate for the additional challenges posed by rigidity theory in $\mathbb{R}^3$ as opposed to $\mathbb{R}^2$. In particular, we extract patches that are not only globally rigid, but also weakly uniquely localizable, a notion that is based on the recent unique localizability of So and Ye [43]. In addition, we also increase the robustness to noise of the algorithm by using a median-based denoising algorithm in the preprocessing step by combining into one step the methods for computing the reflections and rotations and thus doing synchronization over $O(3) = \mathbb{Z}_2\times$ SO(3) rather than individually over $\mathbb{Z}_2$ followed by SO(3). Of equal importance is the possibility to integrate prior available information. As it is often the case in real applications (such as NMR), one has readily available structural information on various parts of the network that we are trying to localize. For example, in the NMR application, there are often subsets of atoms whose relative coordinates are known a priori, and thus it is desirable to be able to incorporate such information in the reconstruction process.

References

1. Akyildiz, I.F., Su, W., Cayirci, Y.S.E.: Wireless sensor networks: A survey. Comput. Network. **38**, 393–422 (2002)
2. Anderson, B.D.O., Belhumeur, P.N., Eren, T., Goldenberg, D.K., Morse, A.S., Whiteley, W.: Graphical properties of easily localizable networks. Wireless Network. **15**, 177–191 (2009)
3. Arie-Nachimson, M., Basri, R., Singer, A.: in preparation
4. Aspnes, J., Eren, T., Goldenberg, D.K., Morse, A.S., Whiteley, W., Yang, Y.R., Anderson, B.D.O., Belhumeur, P.N.: A theory of network localization. IEEE Trans. Mobile. Comput. **5**, 1663–1678 (2006)
5. Aspnes, J., Goldenberg, D.K., Yang, Y.R.: On the computational complexity of sensor network localization. In: Proceedings of Algorithmic Aspects of Wireless Sensor Networks: First International Workshop (ALGOSENSORS), Lecture Notes in Computer Science, pp. 32–44. Springer (2004)
6. Bandeira, A.S., Singer, A., Spielman, D.A.: A cheeger inequality for the graph connection laplacian, arXiv.12043873
7. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**, 1373–1396 (2003)
8. Biswas, P., Aghajan, H., Ye, Y.: Semidefinite programming algorithms for sensor network localization using angle of arrival information. In: Proceedings of the 39th Annual Asilomar Conference on Signals, Systems, and Computers, pp. 220–224 (2005)
9. Biswas, P., Lian, T.C., Wang, T.C., Ye, Y.: Semidefinite programming based algorithms for sensor network localization. ACM Transactions on Sensor Networks **2**, 188–220 (2006) a
10. Biswas, P., Liang, T., Toh, K., Ye, Y., Wang, T.: Semidefinite programming approaches for sensor network localization with noisy distance measurements. IEEE Transactions on Automation Science and Engineering **3**, 360–371 (2006) b
11. Biswas, P., Ye, Y.: Semidefinite programming for ad hoc wireless sensor network localization. In: ACM Conference Proceedings, Third International Symposium on Information Processing in Sensor Networks, pp. 46–54. New York (2004)
12. Borg, I., Groenen, P.J.F.: Modern Multidimensional Scaling: Theory and Applications. Springer, New York (2005)
13. Coifman, R.R., Lafon, S.: Diffusion maps. Applied and Computational Harmonic Analysis **21**, 5–30 (2006)
14. Connelly, R., Whiteley, W.J.: Global rigidity: The effect of coning. Discrete Comput. Geom. **43**, 717–735 (2010)
15. Cox, T.F., Cox, M.A.A.: Multidimensional scaling. Monographs on Statistics and Applied Probability 88. Chapman & Hall/CRC, Boca Raton (2001)
16. Cucuringu, M., Lipman, Y., Singer, A.: Sensor network localization by eigenvector synchronization over the Euclidean group. ACM Tran. Sen. Net. **8**(3), 1–42 (2011)
17. Cucuringu, M., Singer, A., Cowburn, D.: Synchronization, graph rigidity and the molecule problem. submitted (2011)

18. De Leeuw, J.: Applications of convex analysis to multidimensional scaling. In: Barra, J.R., Brodeau, F., Romierand, G., Cutsem, B.V. (eds.) Recent Developments in Statistics, pp. 133–146. North Holland Publishing Company, Amsterdam (1977)
19. Frank, J.: Three-dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State, 2nd edn. Oxford University Press, USA (2006)
20. Giridhar, A., Kumar, P.R.: Distributed clock synchronization over wireless networks: algorithms and analysis. In: IEEE Conference Proceedings, 45th IEEE Conference on Decision and Control, pp. 4915–4920 (2006)
21. Gotsman, C., Koren, Y.: Distributed graph layout for sensor networks. In: Proceedings of the International Symposium on Graph Drawing, pp. 273–284 (2004)
22. Hadani, R., Singer, Representation theoretic patterns in three dimensional Cryo-Electron Microscopy I – the intrinsic reconstitution algorithm, Ann. Math. **174**(2), pp. 1219–1241 (2011).
23. Hendrickson, B.: Conditions for unique graph realizations. SIAM J. Comput. **21**, 65–84 (1992)
24. Hendrickson, B.: The molecule problem: exploiting structure in global optimization. SIAM J. Optim. **5**, 835–857 (1995)
25. Horn, B., Hilden, H., Negahdaripour, S.: Closed-form solution of absolute orientation using orthonormal matrices. J. Opt. Soc. Am. A **5**, 1127–1135 (1988)
26. Javanmard, A., Montanari, A.: Localization from incomplete noisy distance measurements, submitted
27. Ji, X., Zha, H.: Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In: Proceedings of INFOCOM, vol. 4, pp. 2652–2661 (2004)
28. Karp, R., Elson, J., Estrin, D., Shenker, S.: Optimal and global time synchronization in sensornets. Tech. Report, Center for Embedded Networked Sensing, University of California, Los Angeles (2003)
29. Koren, Y., Gotsman, C., Ben-Chen, M.: PATCHWORK: Efficient localization for sensor networks by distributed global optimization. Tech. Report (2005)
30. Liberti, L., Lavor, C., Maculan, N., Mucherino, A.: Euclidean distance geometry and applications. Tech. Report, arXiv.12050349 (2012)
31. Liberti, L., Lavor, C., Mucherino, A., Maculan, N.: Molecular distance geometry methods: From continuous to discrete. Int. Trans. Oper. Res. **18**, 33–51 (2011)
32. Moore, D., Leonard, J., Rus, D., Teller, S.: Robust distributed network localization with noisy range measurements. In: Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems, pp. 50–61 (2004)
33. Ozyesil, O., Singer, A.: Synchronization in non-compact groups: Special euclidean group case, submitted
34. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290** 2323–2326 (2000)
35. Saxe, J.B.: Embeddability of weighted graphs in k-space is strongly NP-hard. In: Proceedings of 17th Allerton Conference in Communications, Control and Computing, pp. 480–489 (1979)
36. Shang, Y., Ruml, W.: Improved MDS-based localization. In: Proceedings of IEEE INFOCOM, vol. 23, pp. 2640–2651. Hong Kong, China (2004)
37. Singer, A.: A remark on global positioning from local distances. Proc. Natl. Acad. Sci. **105**, 9507–9511 (2008)
38. Singer, A.: Angular synchronization by eigenvectors and semidefinite programming. Applied and Computational Harmonic Analysis **30**, 20–36 (2011)
39. Singer, A., Shkolnisky, Three-Dimensional Structure Determination from Common Lines in Cryo-EM by Eigenvectors and Semidefinite Programming. *SIAM* J. Imag. Sci. **4**(2), pp. 543–572 (2011).
40. Singer, A., Wu, H.-T.: Vector diffusion maps and the connection Laplacian, Commun. Pur. Appl. Math. *(CPAM)*, **65**(8), pp. 1067–1144 (2012)
41. Singer, A., Zhao, Z., Shkolnisky, Y., Hadani, R.: Viewing angle classification of cryo-electron microscopy images using eigenvectors. SIAM J. Imag. Sci. **4**, 543–572 (2011)

42. So, A.M.C.: A semidefinite programming approach to the graph realization problem: theory, applications and extensions. Ph.D. Thesis (2007)
43. So, A.M.C., Ye, Y.: Theory of semidefinite programming for sensor network localization. In: Proceedings of the 17th Annual ACM–SIAM Symposium on Discrete Algorithm (SODA), pp. 405–414 (2005)
44. Toh, K., Biswas, P., Ye, Y.: SNLSDP version 0 – a MATLAB software for sensor network localization, October 2008 `http://www.math.nus.edu.sg/ mattohkc/SNLSDP.html`
45. Tubaishat, M., Madria, S.: Sensor networks: an overview. IEEE Potentials **22**, 20–23 (2002)
46. Weinberger, K.Q., Sha, F., Zhu, Q., Saul, L.K.: Graph laplacian regularization for large-scale semidefinite programming. In: Schoolkopf, B., Platt, J., Hofmann, T. (eds.) Advances in neural information processing systems (NIPS), MIT Press, Cambridge (2007)
47. Yemini, Y.: Some theoretical aspects of location-location problems. In: Proceedings of the IEEE Annual Symposium on Foundations of Computer Science, pp. 1–8 (1979)
48. Zhang, L., Liu, L., Gotsman, C., Gortler, S.J.: An As-Rigid-As-Possible approach to sensor network localization. ACM Tran. Sen. Net. **6**(4), 1–21 (2010)
49. Zhu, Z., So, A.M.C., Ye, Y.: Universal rigidity: towards accurate and efficient localization of wireless networks. In: Proceedings of the IEEE INFOCOM, pp. 1–9 (2010)