Published in final edited form as:

IEEE Trans Comput Imaging. 2016 September ; 2(3): 323–334. doi:10.1109/TCI.2016.2557076.

A Graph Partitioning Approach to Simultaneous Angular Reconstitution

Gabi Pragier,

Department of Applied Mathematics, School of Mathematical Sciences, Tel-Aviv University, Tel Aviv 6997801, Israel

Ido Greenberg,

Department of Applied Mathematics, School of Mathematical Sciences, Tel-Aviv University, Tel Aviv 6997801, Israel

Xiuyuan Cheng, and

Applied Mathematics Program, Yale University, New Haven, CT 06520 USA

Yoel Shkolnisky

Department of Applied Mathematics, School of Mathematical Sciences, Tel-Aviv University, Tel Aviv 6997801, Israel

Abstract

One of the primary challenges in single particle reconstruction with cryo-electron microscopy is to find a three-dimensional model of a molecule using its noisy two-dimensional projection-images. As the imaging orientations of the projection-images are unknown, we suggest a common-lines-based method to simultaneously estimate the imaging orientations of all images that is independent of the distribution of the orientations. Since the relative orientation of each pair of images may only be estimated up to a two-way handedness ambiguity, we suggest an efficient procedure to consistently assign the same handedness to all relative orientations. This is achieved by casting the handedness assignment problem as a graph-partitioning problem. Once a consistent handedness of all relative orientations is determined, the orientations corresponding to all projection-images are determined simultaneously, thus rendering the method robust to noise. Our proposed method has also the advantage of allowing one to incorporate confidence information regarding the trustworthiness of each relative orientation in a natural manner. We demonstrate the efficacy of our approach using simulated clean and noisy data.

Index Terms

Angular reconstitution; cryo-electron microscopy; graph partitioning; line graph; synchronization; tomography

I. Introduction

A first step towards investigating the biological functions of a molecule is to determine its three-dimensional structure [13]. A classical technique for achieving this goal is X-ray crystallography, where a purified sample is first crystallized and then exposed to an X-ray beam. This results in a diffraction pattern that may be further analyzed to give valuable

information about the molecule's structure. Cryo-electron microscopy (cryo-EM) and in particular single particle reconstruction is an alternative method for structure determination. In recent years it has gained a lot of popularity [3], [5], [7], as it does not require crystallization, and with recent technological advances, in many cases, achieves near atomic resolution [4], [6]. In cryo-EM, a sample containing copies of the molecule under investigation is first rapidly frozen in a thin layer of ice and is then imaged using an electron microscope. Since each copy is free to move before its orientation is fixed upon freezing, each image is generated from a copy of the molecule rotated by some random unknown rotation.

Formally, if we denote by $\phi: \mathbb{R}^3 \rightarrow \mathbb{R}$ the electric potential of the molecule, and consider a rotation matrix

$$SO(3) \ni R_k = \begin{pmatrix} | & | & | \\ R_k^{(1)} & R_k^{(2)} & R_k^{(3)} \\ | & | & | \end{pmatrix},$$

then the projection-image P_{R_k} is given by

$$\begin{aligned} P_{R_k}(x, y) &= \int_{-\infty}^{\infty} \phi(R_k r) dz \\ &= \int_{-\infty}^{\infty} \phi(x R_k^{(1)} + y R_k^{(2)} + z R_k^{(3)}) dz, \end{aligned} \quad (1)$$

where $r = (x, y, z)^T$. The image P_{R_k} is generated by imaging a copy of the molecule ϕ rotated by R_k^T . The goal is to recover the unknown function ϕ given only a set of its projection-images $\{P_{R_k}\}_{k=1}^n$, where the corresponding rotations R_k are unknown. This may be achieved by first estimating the rotations R_k from the images, followed by standard tomographic inversion algorithms [16], [18]. Thus, a key step in recovering ϕ is estimating the rotations $\{R_k\}_{k=1}^n$.

Since each image P_{R_k} in (1) is the summation of ϕ along some beaming direction (given by $R_k^{(3)}$ in (1)), molecules that are mirror images of one another (enantiomers) result in identical projection-images. Therefore, in cryo-EM imaging, the inherent handedness of the underlying molecule is lost. In [20] it has been shown that given a set of n projection-images, one can recover either the set $\{R_k\}_{k=1}^n$ or the set $\{J R_k J\}_{k=1}^n$, where $J = \text{diag}(1, 1, -1)$ is the reflection through the xy -plane, and there is no way to distinguish between the two sets. Thus, one can only expect to recover either one of these sets of rotations from a given set of projection-images.

Algorithms for recovering the rotations R_k above given only the projection-images are often based on the projection-slice theorem [17], which states that the two-dimensional Fourier transform of any projection-image is equal to the restriction of the three-dimensional Fourier transform of ϕ to the central plane whose normal coincides with the beaming direction of the

image. Thus, any two Fourier transformed projection-images share a common-line in Fourier space.

It has long been known [15], [23] that common-lines between every three projection-images determine the relative orientations between every two of them up to handedness. Thus, any three images establish a coordinate system, and the orientations associated with the remaining projection-images may then be deduced in a sequential manner using the common-lines between any given projection-image and those three images. This sequential approach results in a consistent handedness of all images. Specifically, the two possible sets of orientations of the images (corresponding to the two possible hands of the underlying molecule) are related by an in-plane rotation of 180 degrees of every image [9], [10]. Therefore, to achieve consistent handedness, one of the two possible orientations of the first image is arbitrarily chosen. This effectively fixes the hand of the reconstruction. In particular, there is only one set of possible orientations for the next two images, determined using common-lines among the first three images. Similarly, the orientation of any other image is uniquely determined using common-lines with those three images. The problem, however, with this approach is that it critically depends on the correct positioning of the first three images, which is often inaccurate due to noise, and thus results in errors when assigning the orientations to the remaining images. Various approaches have been proposed to improve the robustness of this procedure. In [8], the common-lines are organized on a sphere in a global manner, and the orientations are recovered using an averaging operator that is applied to the sphere. To better identify the correct common-lines in noisy images, [21] suggested a self-correcting voting procedure in which the common-line between every pair of images is determined by all images in the data set. In [22], a global self-consistency functional for the errors between common-lines is proposed, which is minimized by a convex relaxation and semidefinite programming. In order to handle the plausible case in which there exist large deviations between the detected common-lines and the correct ones, [25] suggested an l_1 relaxation of the self-consistency functional. Other approaches [11], [19] define a global non-convex functional for common-lines discrepancy, and search for its minimum using some optimization scheme.

In this paper we propose a common-lines-based algorithm which finds the orientations of all projection-images at once, thus avoiding the error accumulation that is inherent to sequential-based methods. Moreover, in contrast to optimization-based approaches, it is not susceptible to local minima, and it requires no initial estimate for the orientations. Furthermore, we demonstrate by numerical examples that the algorithm is applicable to a large number of images at once, and is robust to high levels of noise.

A key observation, also proposed in [20], is that one could first establish the relative rotations $R_i^T R_j$ (estimates thereof) between every pair of projection-images $i < j$, $i, j = 1, \dots, n$, then construct a matrix S of size $3n \times 3n$, whose (i, j) th block of size 3×3 is given by $R_i^T R_j$, that is

$$S_{ij} = R_i^T R_j, \quad i, j = 1, \dots, n, \quad (2)$$

and then obtain all rotations at once by the factorization

$$S = H^T H, \quad H = (R_1, \dots, R_n) \in \mathbb{R}^{3 \times 3n}, \quad (3)$$

using SVD. In particular, due to the orthogonality of the rotations, we immediately get that $SH^T = nH^T$. Hence, n is an eigenvalue of S with multiplicity 3, and as S is of rank 3, all its other eigenvalues are 0. Moreover, the rotations are given by the eigenspace of S that corresponds to the eigenvalue n . These properties of S are independent of the distribution of the imaging orientations of the projection-images.

However, once we abandon sequential-based approaches such as [9], [10], [23], a handedness ambiguity might be present in each and every block S_{ij} of S independently of other blocks. That is, for each block S_{ij} it is impossible to tell if it satisfies $S_{ij} = R_i^T R_j$ or rather $S_{ij} = J R_i^T R_j J$. Thus, constructing the matrix S of (2) directly is not possible and an additional, separate procedure for achieving a consistent hand of all blocks S_{ij} is required. Note that checking the two possibilities for each block S_{ij} until we get a rank-3 matrix is impractical, as the complexity of this test is exponential in n^2 . In subsequent derivations we denote by R_{ij} the estimate for the relative rotation between the images P_{R_i} and P_{R_j} which in light of the above discussion satisfies

$$R_{ij} \in \{R_i^T R_j, J R_i^T R_j J\}, \quad J = \text{diag}(1, 1, -1). \quad (4)$$

To circumvent the abovementioned handedness ambiguity in the construction of S , [20] suggested to consider $R_{ij} + J R_{ij} J$ for each pair of images, which is invariant under a possible spurious J -conjugation in the estimate R_{ij} . However, this comes at a cost as the resulting matrix is a $2n \times 2n$ matrix instead of the $3n \times 3n$ matrix S . The factorization of this $2n \times 2n$ matrix does not give the rotations directly as in (3), and an additional, over-constrained system of linear equations emerges. This system is naturally solved by using least squares. Due to noise inherent in the projection-images, the resulting system of equations inevitably contains errors. However, there is no reason to expect that the errors would be normally distributed to justify a least-squares approach, nor does any other meaningful distribution seem appropriate.

In this paper we present an efficient approach which allows the recovery of the matrix S of (2) whose blocks are estimates of $R_i^T R_j$ (up to a possible J -conjugation of all blocks simultaneously). To construct the matrix S , we first need to synchronize the handedness of all estimates R_{ij} . As described above, for each pair of images P_{R_i} P_{R_j} we can estimate either $R_i^T R_j$ or $J R_i^T R_j J$, not being able to distinguish between the two. We would like to divide

the set of estimates $\{R_{ij}: i < j, i, j \in [n]\}$ (we subsequently denote by $[n]$ the set $\{1, 2, 3, \dots, n\}$) into two classes, given by

$$\begin{aligned} G &= \{R_{ij} | R_{ij} = R_i^T R_j\}, \\ B &= \{R_{ij} | R_{ij} = J R_i^T R_j J\}. \end{aligned} \quad (5)$$

Class B (“bad” class) contains all estimates with a spurious J -conjugation, while class G (“good” class) contains all of the remaining estimates, those without a spurious J -conjugation. Obviously, there is nothing bad about the “bad” class, and the naming B and G is therefore given only for convenience. Once the estimates R_{ij} are divided into these two classes, we choose either one of the classes, and replace each estimate R_{ij} in it with $J R_{ij} J$. As a result, since $J^2 = I$, either all estimates R_{ij} , $i, j = 1, \dots, n$, have a spurious J or none do at all. In what follows, we refer to this task as the “ J -synchronization” of all estimates. Once this task is completed, we can form the $3n \times 3n$ matrix S of (2) and estimate all rotations $\{R_k\}_{k=1}^n$ using (3).

The rest of this paper is organized as follows. In Section II we present an algorithm for the J -synchronization of all estimates R_{ij} . In Section III we prove its correctness by casting the task of J -synchronization as a graph-partitioning problem. In Section IV we show that estimating the rotations using the matrix S of (3) allows to integrate confidence information regarding the trustworthiness of each and every relative rotation. Then, in Section V we report some numerical experiments we conducted on simulated clean data, as well as on noisy data. Finally, in Section VI we present some conclusions and possible extensions of this work.

II. J -Synchronization

In this section we show how to partition all estimates R_{ij} , $i, j = 1, \dots, n$, of (4) into the two classes defined by (5). The algorithm consists of two steps: constructing a graph using the estimates R_{ij} , and computing the leading eigenvector of the adjacency matrix of the graph. We then show in Section III that this eigenvector encodes the partitioning of all estimates into the two classes of (5).

Let us define the undirected graph $\Sigma = (V, E)$, where each node in V corresponds to one of

the estimates R_{ij} , $i < j \in [n]$ (and thus $|V| = \binom{n}{2}$), and the set of edges E is given by

$$E = \bigcup_{i < j < k \in [n]} \{\{R_{ij}, R_{jk}\}, \{R_{jk}, R_{ki}\}, \{R_{ki}, R_{ij}\}\}, \quad (6)$$

where we have used curly-brackets to denote that the edges are undirected. In other words, E consists of the edges between all triplets of nodes R_{ij} , R_{jk} , and R_{ki} . The weight of each edge is set to either $+1$ or -1 as explained below. The weight of all other edges, that is edges of

the form (R_{ij}, R_{kl}) for $|\{i, j\} \cap \{k, l\}| = 0$, is set to zero (namely, they do not exist in the

graph). The number of edges in E is thus $|E| = 3 \binom{n}{3}$.

Now, consider any triplet of estimates R_{ij} , R_{jk} , and R_{ki} . If all three belong to the same class in (5) (either all in B or all in G), then $R_{ij} R_{jk} R_{ki} = I$ (since $\mathcal{J}^2 = I$). More generally, we examine which of the following products, each obtained by $\mathcal{J}$ -conjugating a different subset of the original estimates, equals the identity matrix:

$$\begin{array}{ll} 1. R_{ij} R_{jk} R_{ki} & 5. \mathcal{J} R_{ij} \mathcal{J} \mathcal{J} R_{jk} \mathcal{J} \mathcal{J} R_{ki} \mathcal{J} \\ 2. \mathcal{J} R_{ij} \mathcal{J} R_{jk} R_{ki} & 6. R_{ij} \mathcal{J} R_{jk} \mathcal{J} \mathcal{J} R_{ki} \mathcal{J} \\ 3. R_{ij} \mathcal{J} R_{jk} \mathcal{J} R_{ki} & 7. \mathcal{J} R_{ij} \mathcal{J} R_{jk} \mathcal{J} R_{ki} \mathcal{J} \\ 4. R_{ij} R_{jk} \mathcal{J} R_{ki} \mathcal{J} & 8. \mathcal{J} R_{ij} \mathcal{J} \mathcal{J} R_{jk} \mathcal{J} R_{ki} \end{array} \quad (7)$$

For example, when all three estimates belong to the same class, the first and fifth products are equal to the identity matrix. The second and sixth products, for example, correspond to the case where R_{ij} is in one class and R_{jk}, R_{ki} are in the other class. Now, given a triplet of nodes R_{ij}, R_{jk}, R_{ki} , it follows from (6) that there is an edge in Σ between any two of them (that is, they form a “triangle”). We set the weight of each edge out of these three edges to +1 in case that the estimates that correspond to the edge’s incident nodes belong to the same class, and we set it to −1 otherwise. As an example, Fig. 1 depicts the triangles that correspond to the first two cases listed in (7).

Recall that the adjacency matrix of an undirected, unweighted graph with m nodes, is an $m \times m$ symmetric matrix, whose (i, j) th entry equals 1 if there is an edge between nodes v_i and v_j , and equals 0 otherwise. Similarly, we encode the graph Σ using its adjacency matrix, while also taking into account the weights of the edges. Specifically, the weighted adjacency

matrix of the graph Σ , also denoted by Σ , is a $\binom{n}{2} \times \binom{n}{2}$ symmetric matrix whose entries are given by

$$\sum_{(i,j),(k,l)} = \begin{cases} 1, & \text{if } |\{i, j\} \cap \{k, l\}| = 1, \text{ and } R_{ij}, R_{kl} \text{ are in the same class of (5),} \\ -1, & \text{if } |\{i, j\} \cap \{k, l\}| = 1, \text{ and } R_{ij}, R_{kl} \text{ are in different classes of (5),} \\ 0, & \text{if } |\{i, j\} \cap \{k, l\}| = 0. \end{cases} \quad (8)$$

By construction, there exists an edge in Σ between the nodes that correspond to R_{ij} and R_{kl} if and only if $|\{i, j\} \cap \{k, l\}| = 1$. In addition, their mutual edge has the weight +1 when R_{ij} and R_{kl} belong to the same class (i.e., either both belong to class G , or both belong to class B), and it has the weight −1 in case they belong to different classes.

Once the matrix Σ of (8) is constructed, we calculate the eigenvector u_s that corresponds to its leading eigenvalue. In Claim 3.11 below we prove that (i) the leading eigenvalue of Σ is

simple, (ii) $u_s \in \{-\alpha, \alpha\}$ $\binom{n}{2}$ (the value of α depends on the normalization of u_s), and (iii) u_s encodes the class membership of all nodes. Specifically, entries in u_s with value $-\alpha$ correspond to one class, whereas entries with value α correspond to the other class. We then select the set of all nodes that correspond to one of the classes we found (it does not matter which one), and J -conjugate each estimate R_{ij} in it, that is, we replace it by $J R_{ij} J$. By doing so, we can be certain that all estimates R_{ij} belong to the same class, as needed.

The algorithm is summarized in Algorithm 1 and its correctness is proved in Section III. Note that, in practice, the estimates R_{ij} are computed (as explained in [20], [23]) from noisy images, and therefore none of the products in (7) yields exactly the identity matrix. Thus, instead, we search for the product that is as close as possible to the identity matrix. Specifically, we minimize

$$C_{ijk}(\mu_{ij}, \mu_{jk}, \mu_{ki}) = \|J^{\mu_{ij}} R_{ij} J^{\mu_{ij}} \cdot J^{\mu_{jk}} R_{jk} J^{\mu_{jk}} \cdot J^{\mu_{ki}} R_{ki} J^{\mu_{ki}} - I\|_F$$

over $\mu_{ij}, \mu_{jk}, \mu_{ki} \in \{0, 1\}$, where each possible triplet $(\mu_{ij}, \mu_{jk}, \mu_{ki}) \in \{0, 1\}^3$ corresponds to one of the products in (7), and $\|\cdot\|_F$ is the Frobenius norm. We refer to each triplet $(\mu_{ij}, \mu_{jk}, \mu_{ki})$ as a J -configuration of (R_{ij}, R_{jk}, R_{ki}) .

Algorithm 1

J -Synchronization

```

1: Input: Relative rotations  $R_{ij}, i < j \in [n]$ .
2:
   Initialization: Matrix  $\Sigma$ , of size  $\binom{n}{2} \times \binom{n}{2}$ , with all entries set to zero.
3: for  $i < j < k \in [n]$  do
4:    $(\mu_{ij}^*, \mu_{jk}^*, \mu_{ki}^*) \leftarrow \underset{\mu_{ij}, \mu_{jk}, \mu_{ki} \in \{0,1\}}{\operatorname{argmin}} C_{ijk}(\mu_{ij}, \mu_{jk}, \mu_{ki})$  ▷ See (9)
5:    $\sum_{(i,j),(j,k)} \leftarrow (-1)^{(\mu_{ij}^* - \mu_{jk}^*)}$  ▷ assign 1 if  $\mu_{ij}^* = \mu_{jk}^*$  and -1 otherwise
6:    $\sum_{(j,k),(k,i)} \leftarrow (-1)^{(\mu_{jk}^* - \mu_{ki}^*)}$  ▷ assign 1 if  $\mu_{jk}^* = \mu_{ki}^*$  and -1 otherwise
7:    $\sum_{(k,i),(i,j)} \leftarrow (-1)^{(\mu_{ki}^* - \mu_{ij}^*)}$  ▷ assign 1 if  $\mu_{ki}^* = \mu_{ij}^*$  and -1 otherwise
8: end for
9:  $\Sigma \leftarrow \Sigma + \Sigma^T$ 
10:  $u_s \leftarrow \underset{v}{\operatorname{argmax}} \frac{v^T \Sigma v}{\|v\|^2}$  ▷  $u_s$  is the eigenvector that corresponds to the leading eigenvalue of  $\Sigma$ 
11: for  $i < j \in [n]$  do

```

```

12:   if  $u_s(i, j) < 0$  then
13:        $R_{ij} \leftarrow J R_{ij} J$ 
14:   end if
15: end for
16: Output:  $R_{ij}$ ,  $i < j \in [n]$ .

```

III. Algorithm Analysis

In this section, we prove the correctness of Algorithm 1. Following are some definitions that will be used in the analysis of Algorithm 1.

Definition 3.1—Let the eigenvalues of an $m \times m$ symmetric matrix A be $\lambda_1 > \lambda_2 > \dots > \lambda_p \in \mathbb{R}$, and let their respective multiplicities be $k_1, k_2, \dots, k_p \in \mathbb{N}$, $\sum_{i=1}^p k_i = m$. The spectrum of A , denoted by $\sigma(A)$, is written as

$$\sigma(A) = \begin{pmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_p \\ k_1 & k_2 & \dots & k_p \end{pmatrix}.$$

For a (weighted) undirected graph G , we define its spectrum to be the spectrum of its (weighted) adjacency matrix.

Definition 3.2—For an $m \times m$ matrix A , we denote by A^+ the $m \times m$ matrix whose entries are given by $A_{i,j}^+ = |A_{i,j}|$.

In particular, for the adjacency matrix Σ of (8), the matrix $\Sigma^+ \in \{0, 1\}^{\binom{n}{2} \times \binom{n}{2}}$ is the adjacency matrix that corresponds to the same graph structure as Σ , but with all its nodes belonging to the same class (i.e., all edges have weight 1).

Definition 3.3—The line graph of an undirected graph G is an undirected graph, denoted by $L(G)$, specified by the following two properties:

1. Each node of $L(G)$ corresponds to an edge of G , and vice versa.
2. Two nodes of $L(G)$ are adjacent if and only if their corresponding edges in G share a common node.

Definition 3.4—The undirected graph K_n is the graph with n nodes in which any two nodes are connected by an edge. It is also referred to as the complete graph with n nodes.

Equipped with these definitions, we shall prove the following theorem.

Theorem 3.5—The spectrum of the J -synchronization graph Σ is

$$\sigma(\Sigma) = \begin{pmatrix} 2n-4 & n-4 & -2 \\ 1 & n-1 & \begin{pmatrix} n \\ 2 \end{pmatrix} - n \end{pmatrix}.$$

In order to prove Theorem 3.5, we will first find the spectrum of K_n (Lemma 3.6), and show (Claim 3.7) that Σ^+ is the line graph of K_n . We then prove Theorem 3.8, which relates the spectrum of K_n with that of Σ^+ . Finally, we show (Claim 3.9) that Σ and Σ^+ have the same spectrum, enabling us to obtain the spectrum of the matrix Σ . In addition, we demonstrate in Claims 3.10 and 3.11 how the partitioning from (5) may be deduced from the leading eigenvector of the matrix Σ .

Lemma 3.6—The spectrum of the complete graph K_n is

$$\sigma(K_n) = \begin{pmatrix} n-1 & -1 \\ 1 & n-1 \end{pmatrix}.$$

Proof: By definition, every two nodes in K_n are adjacent. Therefore, K_n 's adjacency matrix is $\mathbb{1}\mathbb{1}^T - I$, where $\mathbb{1}$ is the all-ones vector of length n . Hence, K_n 's eigenvalues are obtained by subtracting 1 from the eigenvalues of $\mathbb{1}\mathbb{1}^T$ (and have the same multiplicities). Since $\mathbb{1}\mathbb{1}^T$ is a rank-one matrix, it can be easily verified that

$$\sigma(\mathbb{1}\mathbb{1}^T) = \begin{pmatrix} n & 0 \\ 1 & n-1 \end{pmatrix},$$

which concludes the proof.

Claim 3.7— $L(K_n) = \Sigma^+$. That is, the line graph of the complete graph with n nodes is the absolute-valued J -synchronization graph Σ^+ with $\begin{pmatrix} n \\ 2 \end{pmatrix}$ nodes.

Proof: Let us label the n nodes of K_n by $R_1, R_2, \dots, R_n$. To prove the claim, we show that the requirements of Definition 3.3 hold.

1. By the construction of Σ^+ , for each $i < j$, $i, j \in [n]$, R_{ij} is a node in $V(\Sigma^+)$. By the definition of K_n , for each $i < j$, $i, j \in [n]$, $(R_i, R_j) \in E(K_n)$. Since K_n is an undirected graph, the edges (R_i, R_j) and (R_j, R_i) coincide and we need only consider pairs of indices $i, j \in [n]$ such that $i < j$. Therefore, the mapping from $V(\Sigma^+)$ to $E(K_n)$ given by $R_{ij} \mapsto (R_i, R_j)$ is one-to-one and onto. Thus, as required by Definition 3.3, each node in Σ^+ corresponds to an edge in K_n , and vice versa.
2. Let R_{ij} and R_{kl} be two adjacent nodes in Σ^+ . Then, by construction, $|\{i, j\} \cap \{k, l\}| = 1$. Without loss of generality, we assume that $j = k$. In addition, let (R_i, R_j)

and (R_k, R_l) be the edges in K_n that correspond to the nodes R_{ij} and R_{kl} in Σ^+ . Then, since $j = k$, we have that these two edges share a common node in K_n , namely R_j . Conversely, if R_{ij} and R_{kl} are two non-adjacent nodes in Σ^+ , then by construction $|\{i, j\} \cap \{k, l\}| = 0$. Therefore, the corresponding edges (R_i, R_j) and (R_k, R_l) in K_n indeed do not share a common node.

Theorem 3.8—The spectrum of Σ^+ is given by

$$\sigma(\Sigma^+) = \left(\begin{array}{cc} 2n-4 & n-4 \\ 1 & n-1 \end{array} \begin{array}{c} -2 \\ \left(\begin{array}{c} n \\ 2 \end{array} \right) - n \end{array} \right).$$

A generalization of Theorem 3.8 may be found in [12]. However, we give in the Appendix a proof that applies to our particular case.

Claim 3.9—The adjacency matrices of Σ and Σ^+ are similar (in the matrix sense).

Proof: Without loss of generality, we assume that the indices of the columns of Σ that correspond to the estimates (nodes) R_{ij} in class G are $\{1, 2, \dots, |G|\}$. The remaining indices,

$\{|G| + 1, |G| + 2, \dots, \left(\begin{array}{c} n \\ 2 \end{array} \right)\}$, correspond to the estimates R_{ij} in class B . Let us define the $\left(\begin{array}{c} n \\ 2 \end{array} \right) \times \left(\begin{array}{c} n \\ 2 \end{array} \right)$ matrix D by

$$D = \text{diag}(\underbrace{1, \dots, 1}_{|G|}, \underbrace{-1, \dots, -1}_{|B|}). \quad (9)$$

We prove that Σ and Σ^+ are similar by showing that

$$\Sigma = D \Sigma^+ D^{-1}. \quad (10)$$

To see this, first note that the off-diagonal entries in both D and D^{-1} are all zero, and that $D^{-1} = D$. Therefore,

$$(D \Sigma^+ D^{-1})_{(i,j),(k,l)} = D_{(i,j),(i,j)} \sum_{(i,j),(k,l)}^+ D_{(k,l),(k,l)}.$$

Next, by the definition of D in (9), we get that the matrix $D \Sigma^+ D^{-1}$ is given by

$$(D\Sigma_*^+ D^{-1})_{(i,j),(k,l)} = \begin{cases} \Sigma_{(i,j),(k,l)}^+, & \text{if } R_{ij}, R_{kl} \in G, \\ \Sigma_{(i,j),(k,l)}^+, & \text{if } R_{ij}, R_{kl} \in B, \\ -\Sigma_{(i,j),(k,l)}^+, & \text{otherwise.} \end{cases} \quad (11a) \quad (11b) \quad (11c)$$

Using the definition of Σ in (8) and Definition 3.2 of Σ^+ , we show that (10) holds for each of the above three cases (11a),(11b),(11c).

In case (11a), the nodes R_{ij} and R_{kl} belong to class G . Therefore, by the construction of Σ , it holds that $\Sigma_{(i,j),(k,l)} = 1$ if $|\{i, j\} \cap \{k, l\}| = 1$ (i.e., the nodes are adjacent in Σ), and $\Sigma_{(i,j),(k,l)} = 0$ if $|\{i, j\} \cap \{k, l\}| = 0$ (i.e., the nodes are not adjacent in Σ). Either way, since

$\Sigma_{(i,j),(k,l)}^+ = |\Sigma_{(i,j),(k,l)}|$, we get that

$$\Sigma_{(i,j),(k,l)} = \Sigma_{(i,j),(k,l)}^+ = (D\Sigma_*^+ D^{-1})_{(i,j),(k,l)}. \quad (12)$$

In case (11b), the nodes R_{ij} and R_{kl} belong to class B . Therefore, as in case (11a) we conclude that (12) holds. Finally, case (11c) corresponds to the case where the nodes R_{ij} and R_{kl} belong to different classes. If the nodes are adjacent in Σ , then $\Sigma_{(i,j),(k,l)} = -1$, and since

$\Sigma_{(i,j),(k,l)}^+ = |\Sigma_{(i,j),(k,l)}|$, we conclude that $\Sigma_{(i,j),(k,l)} = -\Sigma_{(i,j),(k,l)}^+$. Alternatively, if the

nodes are not adjacent in Σ , then $\Sigma_{(i,j),(k,l)} = 0$, and therefore $\Sigma_{(i,j),(k,l)} = -\Sigma_{(i,j),(k,l)}^+$ as well. In both cases,

$$\Sigma_{(i,j),(k,l)} = -\Sigma_{(i,j),(k,l)}^+ = (D\Sigma_*^+ D^{-1})_{(i,j),(k,l)}.$$

Proof of Theorem 3.5: The spectrum of Σ^+ was determined in Theorem 3.8. According to Claim 3.9, Σ^+ and Σ are similar. Therefore, they have the same spectrum.

Claim 3.10—The all-ones vector $\mathbb{1}$ of length $\binom{n}{2}$ is the eigen-vector of the matrix Σ^+ that corresponds to its leading eigenvalue $2n - 4$.

Proof: That $2n - 4$ is the leading eigenvalue of the matrix Σ^+ , and that it is simple has already been shown in Theorem 3.8. It therefore remains to show that its corresponding eigenvector is the all-ones vector. Let R_{ij} be a node in the graph Σ^+ . By the construction of the set of edges E in (6), all nodes adjacent to this node are of the form R_{ki} or R_{jk} , with $k \neq i, j$. Thus, there are $n - 2$ possible values for k , and therefore the node R_{ij} has $2n - 4$ adjacent nodes. As a result, each row of the adjacency matrix Σ^+ contains $2n - 4$ ones and zero otherwise. Therefore,

$$\sum^+ \cdot = (2n-4) \cdot, \quad (13)$$

as needed.

Claim 3.11—The eigenvector u_s of the matrix Σ that corresponds to eigenvalue $2n-4$ is given by

$$(u_s)_{(i,j)} = \begin{cases} 1, & \text{if } R_{ij} \in G, \\ -1, & \text{if } R_{ij} \in B. \end{cases}$$

Proof: By Theorem 3.5, $2n-4$ is a simple eigenvalue of Σ . Moreover, by (10) and (13) we get that $u_s = D\mathbf{1}$. The claim now immediately follows, since D is a diagonal matrix with

$$D_{(i,j),(i,j)} = \begin{cases} 1, & \text{if } R_{ij} \in G, \\ -1, & \text{if } R_{ij} \in B. \end{cases}$$

IV. Weighting the Relative Rotations

Based on Algorithm 1 and Eqs. (2) and (3), the procedure for estimating the rotations $\{R_k\}_{k=1}^n$ is summarized in Algorithm 2. As Algorithm 2 uses the output of Algorithm 1, which synchronizes all estimates R_{ij} to be in one of the classes of (5), we assume without loss of generality that $R_{ij} = R_i^T R_j$ for all i and j . An advantage of Algorithm 2 is that it allows to incorporate confidence information regarding the reliability of each estimate R_{ij} . Assuming we are given some weights $w_{ij} \geq 0$ describing the reliability of each estimate R_{ij} , then the following lemma holds.

Lemma 4.1—Let $\tilde{S}$ be a matrix of size $3n \times 3n$ whose (i, j) -th block of size 3×3 is given by $w_{ij} R_{ij}$ where $w_{ij} \geq 0$ and $\sum_{j=1}^n w_{ij} = n$, $i = 1, \dots, n$. Moreover, suppose that $R_{ij} = R_i^T R_j$. Then, the matrix $\tilde{S}$ satisfies $\tilde{S}H^T = nH^T$, with n being the largest eigenvalue of $\tilde{S}$.

Algorithm 2

3nSync

-
- 1: **Input:** Synchronized relative rotations estimates R_{ij}
 $i < j \in [n]$ (computed by Algorithm 1).
  - 2: **Initialization:** Matrix S of size $3n \times 3n$.
  - 3: **for** $i < j \in [n]$ **do**
  - 4: $S_{ij} \leftarrow R_{ij}$
  - 5: **end for**

6: $S \leftarrow S + S^T + I$ $\triangleright$ Since $S_{ii} = R_i^T R_i = I_{3 \times 3}$ $R_i = I_{3 \times 3}$ (see (2)).

7: $H^T H \leftarrow S$ $\triangleright$ Diagonalize S .
 $H = (R_1, \dots, R_n)$.

8: **Output:** $R_i, i \in [n]$.

Proof: From the definition of H in (3), the i -th 3×3 block of $\tilde{S}H^T$ is given by

$$(\tilde{S}H^T)_i = \sum_{j=1}^n w_{ij} R_i^T R_j R_j^T = n R_i^T = n (H^T)_i.$$

Now, let $\lambda \in \mathbb{C}$ be an eigenvalue of $\tilde{S}$ with corresponding eigenvector $u \in \mathbb{C}^{3n}$, and let i_0 be the index of the 3×1 block of u whose 2-norm is maximal, namely, $i_0 = \arg\max_{1 \leq i \leq n} \|u_i\|_2$, where $u_i \in \mathbb{R}^3$ is the i th block of u of size 3×1 . Now

$$|\lambda| \|u_{i_0}\|_2 = \|\lambda u_{i_0}\|_2 = \left\| \sum_{j=1}^n \tilde{S}_{i_0 j} u_j \right\|_2 \leq \sum_{j=1}^n \|\tilde{S}_{i_0 j}\|_2 \|u_j\|_2.$$

Thus, since $\|u_j\|_2 \leq \|u_{i_0}\|_2$ for all j , and since $\|R_{ij}\|_2 = 1$,

$$|\lambda| \leq \sum_{j=1}^n \|\tilde{S}_{i_0 j}\|_2 = \sum_{j=1}^n \|w_{i_0 j} R_{i_0 j}\|_2 = \sum_{j=1}^n |w_{i_0 j}| = n,$$

where the last equality holds because $w_{ij} \geq 0$.

Lemma 4.1 states that the eigenspace that corresponds to the largest eigenvalue n of $\tilde{S}$ encodes the rotations $\{R_k\}_{k=1}^n$. To recover H^T from the top eigenspace of $\tilde{S}$, denoted $\tilde{H}^T$, we compute the SVD of $\tilde{H}^T$, and from the uniqueness of the SVD decomposition, its left singular vectors must be of the form $H^T O$, where O is an orthogonal 3×3 matrix that represents the arbitrary, unknown coordinate system of the underlying molecule.

We show in Section V below that it is advantageous to use the matrix $\tilde{S}$ instead of S , even when using some simple heuristic weights as derived below. More elaborate weighting schemes may result in further improvement. We note that it is currently not known if a similar weighting can be applied to the $2n \times 2n$ synchronization algorithm presented in [20].

To derive the weights w_{ij} , we define

$$(\mu_{ij}^*, \mu_{jk}^*, \mu_{ki}^*) = \underset{\mu_{ij}, \mu_{jk}, \mu_{ki} \in \{0,1\}}{\operatorname{argmin}} C_{ijk}(\mu_{ij}, \mu_{jk}, \mu_{ki}),$$

where C_{ijk} is given by (9). In addition, we denote $C_{ijk}^{\min} = C_{ijk}(\mu_{ij}^*, \mu_{jk}^*, \mu_{ki}^*)$, that is,

$$C_{ijk}^{\min} = \min_{\mu_{ij}, \mu_{jk}, \mu_{ki} \in \{0,1\}} C_{ijk}(\mu_{ij}, \mu_{jk}, \mu_{ki}),$$

and denote by

$$C_{ijk}^{\text{alt}} = \min_{\substack{(\mu_{ij}, \mu_{jk}, \mu_{ki}) \neq (\mu_{ij}^*, \mu_{jk}^*, \mu_{ki}^*) \\ \mu_{ij}, \mu_{jk}, \mu_{ki} \in \{0,1\}}} C_{ijk}(\mu_{ij}, \mu_{jk}, \mu_{ki}),$$

the second lowest value of C_{ijk} .

In the noiseless case, $C_{ijk}(\mu_{ij}^*, \mu_{jk}^*, \mu_{ki}^*) = 0$ if and only if

$J^{\mu_{ij}^*} R_{ij} J^{\mu_{ij}^*} \cdot J^{\mu_{jk}^*} R_{jk} J^{\mu_{jk}^*} \cdot J^{\mu_{ki}^*} R_{ki} J^{\mu_{ki}^*}$ is equal to the identity matrix. Clearly, the J -configuration of a triplet (R_{ij}, R_{jk}, R_{ki}) (defined after (9)) is more likely to be correct as the minimum of C_{ijk} is closer to 0, while $C_{ijk}(\mu_{ij}, \mu_{jk}, \mu_{ki})$ for $(\mu_{ij}, \mu_{jk}, \mu_{ki}) \neq (\mu_{ij}^*, \mu_{jk}^*, \mu_{ki}^*)$ are larger. Therefore, we define the score s_{ijk} of the triplet (R_{ij}, R_{jk}, R_{ki}) to be the normalized difference between C_{ijk}^{alt} and $C_{ijk}^{\min}$, that is

$$s_{ijk} = \frac{C_{ijk}^{\text{alt}} - C_{ijk}^{\min}}{C_{ijk}^{\text{alt}}} = 1 - \frac{C_{ijk}^{\min}}{C_{ijk}^{\text{alt}}}. \quad (14)$$

Note that $s_{ijk} = 1$ if and only if $C_{ijk}^{\min} = 0$ (i.e., there is a “perfect” J -configuration), and $s_{ijk} = 0$ if and only if $C_{ijk}^{\min} = C_{ijk}^{\text{alt}}$ (i.e., there are two equivalent J -configurations). After computing all the scores s_{ijk} for fixed i and j , the weight w_{ij} of the estimate R_{ij} is defined as the mean score over k of s_{ijk} . Finally, the weights are normalized so that $\sum_{j=1}^n w_{ij} = n$, for all $i \in [n]$ (see Lemma 4.1).

As noted above, this weighting scheme is heuristic and is used only for the purpose of demonstrating the advantage in assigning weights to the blocks of $\mathcal{S}$.

V. Numerical Experiments

We implemented Algorithms 1 and 2 in Matlab and compared their performance to the algorithm in [20] and to the LUD algorithm [25], on both clean and noisy simulated projection-images of the 70S subunit of the E. coli ribosome. For Algorithm 2, we tested both the unweighted implementation, as well as an implementation using the weights derived in Section IV. We refer to these two implementations as “unweighted” and “weighted”, respectively.

All tests were executed on a dual Intel Xeon X5560 CPU (12 cores in total), with 96GB of RAM running Linux. Whenever possible, all 12 cores were used simultaneously, either explicitly using Matlab's `parfor`, or implicitly, by employing Matlab's implementation of BLAS, which takes advantage of multi-core computing. Some loop-intensive parts of the algorithm were implemented in c as Matlab mex files.

A. Simulated Clean Data

We created three simulated data sets by generating three sets of n noiseless projection-images, with $n = 10$, $n = 100$, and $n = 1000$. The images in each set are projection-images of a three-dimensional density map of the 70 S subunit of the E. coli ribosome, at orientations drawn uniformly at random from the uniform distribution on $SO(3)$. Each image is of size 129×129 pixels, with each pixel of size 2.74 angstrom. For each image, we computed $L = 360$ radial Fourier lines, with $N = 100$ samples per Fourier line. The common-line between each two projection-images was estimated by computing the correlation between all 360 Fourier lines from one image and all 360 Fourier lines from the other image, and taking the common-line to be the pair with maximum normalized cross-correlation.

Fig. 2(a) shows the spectrum of the matrix Σ of (8) for $n = 10$, which matches the theoretical spectrum predicted by Theorem 3.5. Fig. 2(b)–(c) shows the spectrum of the $3n \times 3n$ synchronization matrix S of (2), for $n = 10, 100, 1000$, respectively. We see that the top three eigenvalues are equal to n with very high accuracy. The remaining eigenvalues are very close to zero. The computed spectrum of the matrix S deviates from the theoretically predicted spectrum due to the discretization of Fourier space using $L = 360$ radial Fourier lines.

Next, we verify that Algorithm 2 accurately restores the rotation matrices $\{R_k\}_{k=1}^n$ from the clean projection-images. To that end, we first estimate the rotations from the matrix S as described by Algorithm 2. We denote the estimated rotations by $\{\tilde{R}_k\}_{k=1}^n$, to distinguish them from the true rotations $\{R_k\}_{k=1}^n$ used for generating the simulated projection-images. As described above, for each projection-image P_{R_i} , we sample its Fourier transform $\hat{P}_{R_i}$ along L radial lines, specifically, along the directions $c_i^{(l)} = (\cos 2\pi l/L, \sin 2\pi l/L, 0)$, $l = 0, \dots, L-1$. For convenience, we lift these vectors to three dimensions, that is, we define $c_i^{(l)} = (\cos 2\pi l/L, \sin 2\pi l/L, 0)$, $l = 0, \dots, L-1$. Then, we compute for each pair of i and l , where $i = 1, \dots, n$, $l = 0, \dots, L-1$, the angle (in degrees) between $R_i c_i^{(l)}$ and $\tilde{R}_i c_i^{(l)}$, that is, we define

$$\varepsilon_{il} = \arccos \langle R_i c_i^{(l)}, \tilde{R}_i c_i^{(l)} \rangle, \quad i = 1, \dots, n, \quad l = 0, \dots, L-1. \quad (15)$$

This gives the error in the estimation of the three-dimensional position of the Fourier ray $c_i^{(l)}$. Since the true and estimated rotations may differ by an arbitrary global orthogonal

transformation, we first register the two sets of three-dimensional vectors $R_i c_i^{(l)}$ and $\tilde{R}_i c_i^{(l)}$, $i = 1, \dots, n$, $l = 0, \dots, L - 1$, as described in [2].

As expected in the noiseless case, the estimation errors (after registration) for all pairs of i and l are lower than 1° (see Fig. 3(a)–(c)), and become smaller as n increases. Note that the error is not zero even in the noiseless case. This is due to the discretization of Fourier space into $L = 360$ radial lines, which means that we do not find the precise common-line between two projection-images, but rather some approximation of it accurate to within $360/(2L)$ degrees, which in our case of $L = 360$ equals 0.5° (assuming correlation always finds the correct common-lines between noiseless projection-images). In other words, when estimating each relative rotation $R_i^T R_j$ using common-lines, we inevitably introduce in the estimate some error due to discretization.

The $\binom{n}{2} \times \binom{n}{2}$ matrix Σ of (8) is sparse and contains only $6 \binom{n}{3}$ non-zero entries.

However, even storing $6 \binom{n}{3}$ elements in memory may be prohibitive for n of the order of a few thousands. Thus, we compute the top eigenvector of Σ using the power method [14], by computing its entries on the fly and storing only the product of Σ by a vector. This way,

only a vector of length $\binom{n}{2}$ needs to be stored in memory.

The total running time of the unweighted algorithm (from images to a three-dimensional density map) was 48 seconds for $n = 10$, 121 seconds for $n = 100$, and 2739 seconds for $n = 1000$. For $n = 1000$, detecting common-lines between all projection-images required 774 seconds, estimating relative rotations based on common-lines required 788 seconds, executing Algorithm 1 required 693 seconds, executing Algorithm 2 required 93 seconds, and reconstructing the density map required 389 seconds. As the required additional computations of the weighted variant of Algorithm 2 were negligible, its running times were essentially the same as above. Note that the running time of the whole algorithm scales cubically with n , but it is acceptable for practical values of n . Moreover, most of the algorithm can be parallelized and hence accelerated as more cores are used, since each element in Algorithms 1 and 2 can be computed independently—each common-line, each relative rotation, and each triplet of the J -synchronization.

B. Simulated Noisy Data

Next, we tested Algorithms 1 and 2 on several sets of noisy projection-images. We took $n = 1000$ noiseless projection-images of the same density map as in Section V-A, and corrupted them by additive Gaussian white noise with $1/64 \leq \text{SNR} \leq 1/8$, where SNR (signal-to-noise ratio) is defined as the ratio between the energy (variance) of the signal and the energy of the noise. Fig. 4 shows four clean projection-images (top row), their noisy realizations at $\text{SNR} = 1/32$ (middle row), and their noisy realizations at $\text{SNR} = 1/48$ (bottom row).

We start by inspecting the gap in the matrix Σ of (8) in the presence of noise. As the analysis in Section III suggests, the spectral gap in the noiseless case is $\frac{\lambda_1}{\lambda_2} = \frac{2n-4}{n-4} \approx 2$. In the presence of noise, Σ contains errors, and therefore the spectral gap decreases up to the point where the top eigenvector no longer encodes the correct partition into the two classes of (5) (see also Claim 3.11).

Fig. 5 shows the spectral gap of the matrix Σ as a function of the SNR. It is evident that the spectral gap decreases as the SNR decreases. However, it also increases as n increases. For example, for $n = 1000$, the spectral gap remains larger than 1.1 up to a noise level of SNR $\approx 1/48$.

Note that for high levels of noise, the spectral gap is quite a weak indication of the success of the global J -synchronization (Algorithm 1). We currently have no appropriate measures to assess the correctness of the J -synchronization of any specific estimate. In fact, the best measure we have for the success of the J -synchronization, is a success in estimating the orientations with a small error, as described below.

Next, we inspect the gap of the (unweighted) matrix S as a function of the SNR. Table I shows the spectral gap, defined as λ_3/λ_4 , for both the $2n \times 2n$ matrix of [20] and the matrix S of (2). The spectrum of the weighted matrix $\tilde{S}$ (see Lemma 4.1) is very similar to that of the unweighted matrix, and is thus not shown. It can be seen that for both the $2n \times 2n$ synchronization matrix of [20], and for the matrix S from (2), in the presence of noise, the matrix still has three dominant eigenvalues, though the remaining eigenvalues are no longer small due to misidentifications of common-lines. Still, there is a significant gap between the third and the fourth eigenvalues, which gets smaller as the noise increases. We also observe that the spectral gap is consistently larger in the matrix S compared to the $2n \times 2n$ matrix of [20].

The percentiles of the angle estimation errors e_{ih} given by (15), for SNR = 1/32 and SNR = 1/48, are shown in Fig. 6, which shows that the errors increase with the noise level. For Algorithm 2 (named $3n\text{Sync}$), for SNR = 1/32, more than 95% of the errors e_{ih} are lower than 20° , while for SNR = 1/48 about 80% of the errors are less than 20° . We also see that at both noise levels, Algorithm 2 achieves lower errors than the algorithms in [20], [25]. The weighted variant of Algorithm 2 is not shown in Fig. 6 in order to make it easy to read. Its performance is summarized in Table II discussed next.

Table II summarizes the performance of Algorithm 2 for various levels of noise. We see that both the unweighted and weighted variants of Algorithm 2 restore the projection orientations with low errors even when the percentage of correctly detected common-lines is low. For SNR = 1/32, for example, only 13% of the common-lines are detected with an error lower than 5° , and thus at least 87% of the blocks of S are wrong (other blocks may be inaccurate as well, since each block is estimated using triplets of images). Nevertheless, the average error of e_{ih} (over all i and h) is merely 7.7° . This demonstrates the robustness of the algorithm to noise. Moreover, Table II also demonstrates that Algorithm 2 consistently outperforms both [20] and [25], and that the weighted variant of Algorithm 2 achieves additional consistent improvement.

Once the rotations have been estimated by Algorithm 2, the density maps were reconstructed using the FIRM algorithm in the ASPIRE software package [1]. Two-dimensional renderings of some of the reconstructed density maps are shown in Fig. 7. Fig. 8 shows the Fourier shell correlation curves [24] for the various reconstructions, computed with respect to the reference density map from which the simulated projection-images were initially generated. Fig. 9 shows the resolutions achieved by the various algorithms as a function of the noise, using the 0.143 correlation criterion [24]. The curves show that—as expected—high levels of noise cause loss of resolution. In addition, it is evident once again that improvement is achieved by Algorithm 2, and that further improvement is attained by its weighted variant.

VI. Conclusion

We have presented an algorithm for estimating the projection orientations that match a given set of input projection-images. The algorithm first synchronizes the handedness of all of the estimates for the relative rotations between the images, and then factorizes the matrix containing all synchronized estimates. The proposed algorithm has two important advantages. First, unlike the algorithm in [20], its performance is independent of the distribution of the projection orientations. Second, it can exploit information about the reliability of the estimates of the relative rotations. We demonstrated using simulated data that the algorithm is robust to high levels of noise, and that further robustness is achieved by using a heuristic reliability measure for the relative rotations.

There are several possible extensions to the algorithm. First, while we analyzed the spectra of the matrices in the algorithm in the noiseless setting, it may be possible to analyze them in the presence of noise. This may allow us to derive theoretical performance bounds for the algorithm, as a function of the noise level. Second, as it was demonstrated that even a simple heuristic weighting scheme improves the performance of the algorithm, designing more advanced weighting schemes may also result in further improvement. Finally, the proposed algorithm may be used in an iterative scheme, where in each iteration we use information deduced from previous iterations to estimate the reliability of the various relative rotations, and redefine their weights accordingly.

Acknowledgments

This work was supported in part by the Israel Science Foundation under Grant 578/14 and in part by the NIGMS under Grant R01GM090200. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Edwin Marengo.

The authors would like to thank D. Dynerman and M. Shagam for their assistance in preparing this manuscript for publication.

References

1. Algorithms for single particle reconstruction. [Online]. Available: <http://spr.math.princeton.edu/>
2. Arun KS, Huang TS, Blostein SD. Least-squares fitting of two 3-D point sets. *IEEE Trans Pattern Anal Mach Intell.* Sep; 1987 9(5):698–700. [PubMed: 21869429]
3. Bai X, McMullan G, Scheres SHW. How cryo-EM is revolutionizing structural biology. *Trends Biochemical Sci.* 2015; 40(1):49–57.

4. Bartesaghi A, Merk A, Banerjee S, Matthies D, Wu X, Milne JLS, Subramaniam S. 2.2 Å resolution cryo-EM structure of β -galactosidase in complex with a cell-permeant inhibitor. *Science*. 2015; 348(6239):1147–1151. [PubMed: 25953817]
5. Callaway E. The revolution will not be crystallized: A new method sweeps through structural biology. *Nature*. 2015; 525(7568):172–174. [PubMed: 26354465]
6. Campbell MG, Veesler D, Cheng A, Potter CS, Carragher B. 2.8 Å resolution reconstruction of the thermoplasma acidophilum 20S proteasome using cryo-electron microscopy. *eLife*. 2015; 4:e06380.
7. Cheng Y, Grigorieff N, Penczek PA, Walz T. A primer to single-particle cryo-electron microscopy. *Cell*. 2015; 161(3):438–449. [PubMed: 25910204]
8. Coifman RR, Shkolnisky Y, Sigworth FJ, Singer A. Reference-free structure determination through eigenvectors of center of mass operators. *Appl Comput Harmonic Anal*. 2010; 28(3):296–312.
9. Crowther RA, Amos LA, Finch JT, Klug A. Three-dimensional reconstruction of spherical viruses by Fourier synthesis from electron micrographs. *Nature*. 1970; 226:421–425. [PubMed: 4314822]
10. Crowther RA, DeRosier DJ, Klug A. The reconstruction of a three-dimensional structure from projections and its application to electron microscopy. *Proc Roy Soc B*. 1970; 317:319–340.
11. Elmlund D, Davis R, Elmlund H. Ab initio structure determination from electron microscopic images of single molecules coexisting in different functional states. *Structure*. 2010; 18:777–786. [PubMed: 20637414]
12. Fiol MA, Mitjana M. The spectra of some families of digraphs. *Linear Algebra Appl*. 2007; 423:109–118.
13. Frank J. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State*. New York, NY, USA: Oxford; 2006.
14. Golub, GH., Van Loan, CF. *Matrix Computation* (Johns Hopkins Studies in Mathematical Sciences). Baltimore, MD, USA: The Johns Hopkins Univ. Press; 1984.
15. Goncharov AB. Integral geometry and three-dimensional reconstruction of randomly oriented identical particles from their electron microphotos. *Acta Appl Math*. 1988; 11:199–211.
16. Herman, GT. *Fundamentals of Computerized Tomography: Image Reconstruction From Projections*. 2. London, UK: Springer; 2009.
17. Natterer, F. *The Mathematics of Computerized Tomography* (Classics in Applied Mathematics). Philadelphia, PA, USA: SIAM; 2001.
18. Natterer, F., Wübbeling, F. *Mathematical Methods in Image Reconstruction* (SIAM Monographs on Mathematical Modeling and Computation). Philadelphia, PA, USA: SIAM; 2001.
19. Penczek PA, Zhu J, Frank J. A common-lines based method for determining orientations for $N > 3$ particle projections simultaneously. *Ultramicroscopy*. 1996; 63:205–218. [PubMed: 8921628]
20. Shkolnisky Y, Singer A. Viewing directions estimation in cryo-EM using synchronization. *SIAM J Imag Sci*. 2012; 5(3):1088–1110.
21. Singer A, Coifman RR, Sigworth FJ, Chester DW, Shkolnisky Y. Detecting consistent common lines in cryo-EM by voting. *J Structural Biol*. 2010; 169(3):312–322.
22. Singer A, Shkolnisky Y. Three-dimensional structure determination from common lines in cryo-EM by eigenvectors and semidefinite programming. *SIAM J Imag Sci*. 2011; 4(2):543–572.
23. van Heel M. Angular reconstitution: A posteriori assignment of projection directions for 3D reconstruction. *Ultramicroscopy*. 1987; 21(2):111–123. [PubMed: 12425301]
24. van Heel M, Schatz M. Fourier shell correlation threshold criteria. *J Structural Biol*. 2005; 151(3): 250–262.
25. Wang L, Singer A, Wen Z. Orientation determination of cryo-EM images using least unsquared deviations. *SIAM J Imag Sci*. 2013; 6(4):2450–2483.

Biographies

Gabi Pragier received the B.Sc. degree in mathematics and computer science from the Bar-Ilan University, Ramat Gan, Israel, in 2006, and the M.Sc. degree in computer science from the Hebrew University of Jerusalem, Jerusalem, Israel, in 2010. He is currently working toward the Ph.D. degree at the School of Mathematical Sciences, Tel-Aviv University, Tel Aviv, Israel. His research interests include machine learning, computer vision and scientific computing.

Ido Greenberg received the B.Sc. degree in mathematics and physics from the Hebrew University of Jerusalem, Jerusalem, Israel, in 2011. He is currently working toward the M.Sc. degree at the School of Mathematical Sciences, Tel-Aviv University, Tel Aviv, Israel. Since 2008, he has been serving the Israeli Defense Forces. His research interests include data analysis, computer vision and machine learning.

Xiuyuan Cheng received the Ph.D. degree from the Program of Applied and Computational Mathematics, Princeton University, Princeton, NJ, USA, in 2013, advised jointly by Prof. Amit Singer and Prof. Weinan E. She is currently a Gibbs Assistant Professor at the Program of Applied Mathematics, Yale University, New Haven, CT, USA. From 2013 to 2015, she was a Postdoctoral Researcher at the Département d'Informatique of École normale supérieure, France. Her research focuses on high dimensional data analysis and mathematical theories of machine learning.

Yoel Shkolnisky received the B.Sc. degree in mathematics and computer science and the M.Sc. and Ph.D. degrees in computer science, from the Tel-Aviv University, Tel Aviv, Israel, in 1996, 2001, and 2005, respectively. From July 2005 to July 2008, he worked as a Gibbs Assistant Professor in Applied Mathematics at the Department of Mathematics, Yale University, New Haven, CT, USA. Since October 2009, he has been with the Department of Applied Mathematics, School of Mathematical Sciences, Tel-Aviv University, Tel Aviv, Israel. His research interests include computational harmonic analysis, scientific computing, and data analysis.

Appendix. Proof of Theorem 3.8

Definition A.1

The incidence matrix of an undirected graph G with n nodes and m edges is a $n \times m$ matrix whose (i, e) th entry equals 1 if node i is an endpoint of edge e , and equals 0 otherwise.

Claim A.2

Let B be the incidence matrix of the complete graph K_n . Then, the adjacency matrix of K_n is equal to $BB^T - (n-1)I$.

Proof

By Definition 3.4, each node in K_n is an endpoint of $n-1$ edges. Thus, according to Definition A.1, we get by direct calculation that

$$(BB^T)_{i,j} = \begin{cases} n-1, & \text{if } i=j, \\ 1, & \text{if } i \neq j \text{ and } i \text{ and } j \text{ are adjacent in } K_n, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.16})$$

Since K_n is a complete graph, nodes i and j are adjacent for all $i \neq j$. Therefore $BB^T - (n-1)I = \mathbb{1}\mathbb{1}^T - I$, which according to the proof of Lemma 3.6 is the adjacency matrix of K_n .

Claim A.3

Let B be the incidence matrix of the complete graph K_n . Then, the adjacency matrix of Σ^+ is $B^T B - 2I$.

Proof

Using Definition A.1 we get by direct calculation that $(B^T B)_{e,u}$ is the number of nodes in K_n that edges e and u have in common. In other words, for $e \neq u$ it holds that

$$(B^T B)_{e,u} = \begin{cases} 1, & \text{if } e, u \text{ have a common node in } K_n, \\ 0, & \text{if } e, u \text{ do not have a common node in } K_n, \end{cases}$$

and for $e = u$, since every edge has exactly two incident nodes, we get that $(B^T B)_{e,e} = 2$.

Thus, by Definition 3.3, we conclude that $B^T B - 2I$ is the adjacency matrix of the line graph $L(K_n)$ which equals Σ^+ (Claim 3.7).

Claim A.4

If λ is an eigenvalue of the adjacency matrix of K_n , then $\lambda + (n-1) - 2$ is an eigenvalue of the adjacency matrix of Σ^+ , with the same multiplicity.

Proof

Let B be the incidence matrix of K_n . By Claim A.2, the adjacency matrix of K_n is $BB^T - (n-1)I$. Therefore, if λ is an eigenvalue of K_n , then there exists a non-zero vector u such that $(BB^T - (n-1)I)u = \lambda u$. Thus,

$$\begin{aligned} (B^T B - 2I)B^T u &= (B^T B - 2I)B^T u - (n-1)B^T u + (n-1)B^T u \\ &= B^T \underbrace{(BB^T - (n-1)I)u}_{\lambda u} + B^T((n-1)-2)u \\ &= (\lambda + (n-1) - 2)B^T u. \end{aligned} \tag{A.17}$$

This means that $B^T u$ is an eigenvector of the matrix $B^T B - 2I$ (the adjacency matrix of Σ^+ , as shown in Claim A.3) with corresponding eigenvalue $\lambda + (n-1) - 2$, provided that it is not the zero vector. To show that this is indeed the case, let us assume the contrary, i.e., $B^T u = 0$. By Definition A.1, each row e in B^T is all zeros except for the two entries i and j that correspond to the end-nodes of the edge e . Therefore, for $B^T u$ to be the zero vector, u must have entries ± 1 for the end-nodes of every edge in K_n . But since K_n has odd cycles (e.g., cliques of three nodes), at least one of its edges must have both of its end-nodes entries with the same value (i.e., either both 1 or both -1), contradicting our assumption.

Next we show that the multiplicity k of any eigenvalue λ of K_n equals the multiplicity of its counterpart eigenvalue $\lambda + (n-1) - 2$ in Σ^+ . Let $u_1, u_2, \dots, u_k$ be k linearly independent eigenvectors of K_n that correspond to λ . In light of (A.17), it suffices to show that the vectors $B^T u_1, B^T u_2, \dots, B^T u_k$ are also linearly independent. To prove this, we note that for any $\beta_1, \beta_2, \dots, \beta_k \in \mathbb{R}$, if $\sum_{i=1}^k \beta_i B^T u_i = 0$, then

$$\sum_{i=1}^k \beta_i B B^T u_i = 0.$$

Since the adjacency matrix of K_n is $BB^T - (n-1)I$ (Claim A.2), and so $(BB^T - (n-1)I)u_i = \lambda u_i$, $i = 1 \dots, k$, we get that

$$\sum_{i=1}^k \beta_i (\lambda + n - 1) u_i = 0.$$

Since $\lambda \neq -(n-1)$ (Lemma 3.6), this reduces to

$$\sum_{i=1}^k \beta_i u_i = 0.$$

Finally, since $u_1, u_2, \dots, u_k$ are linearly independent, we conclude that $\beta_i = 0$ for all i , showing that the vectors $B^T u_1, B^T u_2, \dots, B^T u_k$ are indeed linearly independent.

Corollary A.5

The adjacency matrix of Σ^+ has an eigenvalue $2n-4$ with multiplicity 1 and an eigenvalue $n-4$ with multiplicity $n-1$.

Proof

According to Lemma 3.6, $n-1$ and -1 are eigenvalues of the adjacency matrix of K_n with multiplicities 1 and $n-1$, respectively. The corollary now follows directly from Claim A.4.

Claim A.6

Let B be the incidence matrix of K_n . Then, $\text{rank}(B^T B) = n$.

Proof

As B is of size $n \times \binom{n}{2}$, we know that $\text{rank}(B^T B) \leq n$. It therefore remains to show that $\text{rank}(B^T B) = n$, namely, that the sum of the multiplicities of all non-zero eigenvalues of $B^T B$ is at least n .

A direct consequence of Claim A.4 is that if λ is an eigenvalue of the adjacency matrix of K_n then $\lambda + (n-1)$ is an eigenvalue of $B^T B$ with the same multiplicity. Let then λ be such an eigenvalue. Then, $\lambda + (n-1) \neq 0$. For otherwise, $\lambda = -(n-1)$ is an eigenvalue of the adjacency matrix of K_n in contradiction to Lemma 3.6. In addition, the sum of the multiplicities of all eigenvalues of the adjacency matrix of K_n equals n . Therefore, by Claim

A.4, the sum of all multiplicities of the non-zero eigenvalues $\lambda + (n - 1)$ of the matrix $B^T B$ also equals n , proving that indeed $\text{rank}(B^T B) = n$.

Claim A.7

The adjacency matrix of Σ^+ has an eigenvalue -2 with multiplicity $\binom{n}{2} - n$.

Proof

Let B be the incidence matrix of K_n . By Claim A.3, the adjacency matrix of Σ^+ is $B^T B - 2I$.

Therefore, it suffices to show that 0 is an eigenvalue of $B^T B$ with multiplicity $\binom{n}{2} - n$.

Recall that $B^T B$ is of size $\binom{n}{2} \times \binom{n}{2}$. Thus, by the rank-nullity theorem

$$\dim \text{null}(B^T B) = \binom{n}{2} - \text{rank}(B^T B). \quad (\text{A.18})$$

As $\text{rank}(B^T B) = n$ (Claim A.6), we conclude that

$$\dim \text{null}(B^T B) = \binom{n}{2} - n,$$

as needed.

Proof of Theorem 3.8

By Claim A.7, we know that -2 is an eigenvalue of the adjacency matrix of Σ^+ with

multiplicity $\binom{n}{2} - n$. In addition, by Corollary A.5, this matrix has an eigenvalue $2n - 4$ with multiplicity 1 and an eigenvalue $n - 4$ with multiplicity $n - 1$. The sum of all these

multiplicities is $\binom{n}{2}$, which is exactly the dimension of the square adjacency matrix of Σ^+ , and so there are no additional eigenvalues besides those.

Fig. 1.

Subgraphs corresponding to the first two configurations in (7). The underlined node (cyan) belongs to class B; other nodes (magenta) belong to class G. The edges have weight equal to 1 if the incident nodes belong to the same class, and weight -1 otherwise. (a) Configuration 1. (b) Configuration 2.

Fig. 2.

Spectra of Σ and S . (a) Spectrum of Σ for $n = 10$. (b) Spectrum of S for $n = 10$. (c) Spectrum of S for $n = 100$. (d) Spectrum of S for $n = 1000$.

Fig. 3. Histogram of estimation errors in the noiseless case. (a) $n = 10$. (b) $n = 100$. (c) $n = 1000$.

Fig. 4.

Top row: Sample of 4 simulated noiseless projection-images of the 70S subunit of the *E. coli* ribosome. Middle row: Noisy realizations at $\text{SNR} = 1/32$. Bottom row: Noisy realizations at $\text{SNR} = 1/48$. The noisy projection-images were corrupted by additive Gaussian white noise with the prescribed SNR. (a) Clean projection-images. (b) $\text{SNR} = 1/32$. (c) $\text{SNR} = 1/48$.

Fig. 5.
Spectral gap of the matrix Σ as a function of the SNR.

Fig. 6.

Percentiles of the estimation errors ε_{II} of (15) for SNR = 1/32 and SNR = 1/48.

Fig. 7.

Reconstructed density maps. (a) Original. (b) $3n\text{Sync}$ weighted, $\text{SNR} = 1/32$. (c) $3n\text{Sync}$ weighted, $\text{SNR} = 1/48$. (d) $3n\text{Sync}$ unweighted, $\text{SNR} = 1/32$. (e) $3n\text{Sync}$ unweighted, $\text{SNR} = 1/48$. (f) $2n \times 2n$ [20], $\text{SNR} = 1/32$. (g) $2n \times 2n$ [20], $\text{SNR} = 1/48$. (h) LUD [25], $\text{SNR} = 1/32$. (i) LUD [25], $\text{SNR} = 1/48$.

Fig. 8.

Fourier shell correlation curves for the density maps reconstructed by the various algorithms (the curve for LUD [25] was omitted to improve readability), for SNR=1/32 and SNR=1/48. The curves were computed with respect to the reference density map from which the simulated projection-images were initially generated.

Fig. 9.

Resolution of the reconstructed density maps as a function of the noise level in the projection-images. The resolution is computed using the 0.143 correlation criterion.

TABLE I Spectral Gap of the Matrix $\mathcal{S}$, Expressed by the Ratio λ_3/λ_4 , for Various Levels of Noise

SNR	Spectral gap (λ_3/λ_4)	
	$2n \times 2n$ [20]	S (Eq. (2)) (unweighted)
Clean	421.2	521.5
1/8	7.1	9.9
1/16	3.8	5.7
1/24	2.7	4.4
1/32	2.1	3.6
1/40	1.6	3.1
1/48	1.4	2.8
1/56	1.2	2.4
1/64	1.2	1.9

TABLE II

Mean Estimation Errors for Various Levels of Noise

SNR	% of correct common-lines	LUD [25] Mean error	$2n \times 2n$ [20] Mean error	$3n$ Sync unweighted Mean error	$3n$ Sync weighted Mean error
Clean	100	0.1°	0.1°	0.1°	0.1°
1/8	56	1.7°	2.4°	1.8°	1.1°
1/16	31	3.9°	5.0°	3.9°	2.6°
1/24	20	7.3°	7.6°	5.8°	4.5°
1/32	13	11.1°	10.5°	7.7°	6.8°
1/40	10	15.4°	14.3°	10.1°	9.4°
1/48	7	20.8°	19.4°	13.2°	12.6°
1/56	6	28.4°	26.6°	17.4°	16.9°
1/64	5	35.7°	34.5°	27.7°	27.3°