SIAM J. IMAGING SCIENCES
Vol. 5, No. 3, pp. 1088–1110

Viewing Direction Estimation in Cryo-EM Using Synchronization*

Yoel Shkolnisky† and Amit Singer‡

Abstract. A central task in recovering the structure of a macromolecule from cryo-electron microscopy (cryo-EM) images is to determine a three-dimensional model of the macromolecule given many of its two-dimensional projection images. The direction from which each image was taken is unknown, and the images are small and extremely noisy. The goal is to determine the direction from which each image was taken and then to combine the images into a three-dimensional model of the molecule. We present an algorithm for determining the viewing direction of all cryo-EM images at once, which is robust to high levels of noise. The algorithm is based on formulating the problem as a synchronization problem; that is, we estimate the relative spatial configuration of pairs of images and then estimate a global assignment of orientations that maximizes the number of satisfied pairwise relations. Information about the spatial relation between pairs of images is extracted from common lines between triplets of images. These noisy pairwise relations are combined into a single consistent assignment of orientations by constructing a matrix whose entries encode the pairwise relations. This matrix is shown to have rank 3, and its nontrivial eigenspace is shown to reveal the projection orientation of each image. In particular, we show that the nontrivial eigenvectors encode the rotation matrix that corresponds to each image.

Key words. angular reconstitution, cryo-electron microscopy, synchronization, tomography

AMS subject classifications. 92E10, 68U10, 94A08, 92C55

DOI. 10.1137/120863642

1. Introduction.

One of the fundamental tasks in structural biology is to recover the three-dimensional structure of molecules. Electron microscopy is one of the popular methods for that task, and in particular single particle reconstruction (SPR), where the structure is determined from images of randomly oriented and positioned identical copies of the investigated molecule. One of the available specimen preparation techniques for SPR is cryo-electron microscopy (cryo-EM), in which many copies of the investigated molecule are rapidly frozen in a thin layer of ice [5, 23]. This method is experimentally attractive as it does not require crystallization as in X-ray crystallography. However, cryo-EM images have very low contrast and are very noisy, due to the small electron dose that can be applied to the specimen.

As each particle in a cryo-EM experiment is free to move prior to the freezing of the specimen, the orientation of each individual particle in the imaged specimen is unknown. The

*Received by the editors January 25, 2012; accepted for publication (in revised form) June 28, 2012; published electronically September 20, 2012.
http://www.siam.org/journals/siims/5-3/86364.html
†Department of Applied Mathematics, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel (yoelsh@post.tau.ac.il). This author's work was partially supported by Israel Science Foundation grant 485/10 and by the Raymond and Beverly Sackler Career Development Chair.
‡Department of Mathematics and PACM, Princeton University, Princeton, NJ 08544-1000 (amits@math.princeton.edu). This author's work was partially supported by award DMS-0914892 from the NSF, award FA9550-09-1-0551 from AFOSR, award R01GM090200 from the National Institute of General Medical Sciences, and the Alfred P. Sloan Foundation.

orientations need not even be uniform, as the molecule might have some preferred orientation due to, for example, its interaction with the supporting grid. Due to the low electron dose used in order to prevent damaging the specimen, each particle can be imaged only once.

The output from the electron microscope is a micrograph (a large two-dimensional image) containing the two-dimensional projections of hundreds of particles, where each two-dimensional projection corresponds to a randomly rotated copy of the underlying three-dimensional volume. In section 3 we define precisely the image formation model in cryo-EM, which explains the relation between the three-dimensional volume and its two-dimensional images. The individual particles are segmented and extracted from the micrograph, producing a data set of thousands of raw projection images. Typical data sets contain 10^4–10^5 particle images. The goal in cryo-EM structure determination is to compute the three-dimensional structure from those two-dimensional noisy images.

The process of determining a three-dimensional structure from two-dimensional images typically consists of two steps. In the first step, some low resolution ab initio model of the structure is estimated. Then this model is refined by an iterative procedure using the set of raw images. The accuracy of the ab initio model is crucial for the quality of the final refined model. For background on cryo-EM imaging, see [5, 21].

There are two main approaches for estimating an ab initio model. If the molecule is known to have some preferred orientation, then it is possible to find an ab initio three-dimensional structure using the random conical tilt method [12, 13]. Algorithms which do not involve tilting are typically based on the "angular reconstitution" method of Van Heel [20] in which a coordinate system is established from three projections and the orientation of the particle giving rise to each image is deduced from common lines among the images. Those algorithms often fail when the particles are too small or when the signal-to-noise ratio is too low, as in such cases it is difficult to correctly identify the common lines. For a detailed discussion of common line methods, see [17]. There also exist other approaches [7, 14], based on the method of moments, which exploit the known analytical relation between the second order moments of the two-dimensional projection images and the second order moments of the (unknown) three-dimensional volume in order to reveal the unknown orientations of the particles. However, the method of moments is very sensitive to errors in the data and is of rather academic interest [10, section 2.1, p. 251].

In this paper we present an algorithm for ab initio reconstruction that is also based on common lines. In particular, we present an algorithm that, given N projection images, estimates for each image the rotation applied to the three-dimensional volume prior to freezing, which resulted in the given image. The algorithm first uses the common line property to estimate the three-dimensional configuration of pairs of images. Then it robustly combines this pairwise information to estimate the three-dimensional rotation of each copy of the volume. The process of finding a global assignment from pairwise relations is known as synchronization [15]. Thus, we refer to the algorithm in this paper as a synchronization algorithm. This point is clarified in section 4. Unlike the angular reconstitution method [20], our method takes into account all common lines between all pairs of images simultaneously. The resulting algorithm is therefore robust to noise, as demonstrated in section 6.

The structure of the paper is as follows. In section 2 we describe previous work on common line algorithms and their relation to the present work. In section 3 we revisit the Fourier

projection-slice theorem and the common line property. In section 4 we show that common lines between triplets of projections enable us to estimate the relative configuration of pairs of projections, and, moreover, that it is possible to estimate the absolute orientation of the molecule giving rise to each of the images from those pairwise relative relations. In section 5 we analyze some important mathematical properties of the algorithm by considering the case where the number of images goes to infinity. Section 6 then presents numerical examples of the algorithm for simulated data sets, as well as for class averages produced from experimental data. We conclude with a discussion and possible extensions of the algorithm in section 7.

2. Relation to previous work. The algorithm in this paper is a continuation of ongoing research on ab initio reconstruction methods. The algorithm of the current paper together with two previously published algorithms [4, 17] provide three different approaches to the ab initio reconstruction problem. All three methods are based on common lines and estimate the viewing parameters of each image from the eigenvectors of some matrix constructed from common line information. However, the geometry and the local information used to construct each matrix are different for each of the algorithms, and therefore they also have different properties and robustness to noise. In [4] we used common line information to construct an averaging operator on the sphere. The three-dimensional eigenspace was shown to encode the three-dimensional direction vector of each of the Fourier rays, obtained by taking the polar Fourier transform of the input images. Although the resulting algorithm in [4] is significantly more robust to noise than the classical angular reconstitution [20], it is less robust than [17] and the algorithm of this paper. The reason is that the spectral gap in the resulting matrix between the eigenspace that encodes the orientations and the next eigenspace is smaller than in the other algorithms. Next, in [17] we presented two algorithms that are based on formulating the orientation assignment problem as an optimization problem, followed by relaxing the optimization problem into a computationally tractable one. The first algorithm in [17] was based on relaxing the optimization problem into a problem of finding eigenvectors of a matrix and required the viewing directions to be uniformly distributed. The spectral gap of this algorithm was computed to be asymptotically equal to 5/12 (see also [8]). The second algorithm was based on a relaxation to a semidefinite program and imposed no assumption on the viewing directions. However, it is much more computationally intensive, and its performance in the presence of noise is more difficult to analyze.

The algorithm in the present paper is also based on constructing a matrix using common line information and computing eigenvectors of this matrix. However, besides the different mathematical approach, there are two important differences between this algorithm and the algorithms in [4, 17], which translate into significantly improved robustness to noise. First, the information used for constructing the matrix is based on triplets of projections, unlike the algorithms in [4, 17] which are based on common lines between pairs of projections. This has the effect of denoising the resulting matrix and thus increasing the robustness to noise (see section 7 for a discussion). Second, the resulting matrix is of rank 3, and its spectral gap is asymptotically 2/3. This gap is 60% larger than the gap for the algorithm in [17], which again translates into further noise robustness.

3. Common lines and local geometry. In this section we review the well-known Fourier projection-slice theorem (see, for example, [9]) and its induced common line property.

Given N projection images, the pixel intensities in each image are given by

$$P_i(x,y) = \int_{-\infty}^{\infty} \phi\left(R_i r\right)\, dz, \tag{3.1}$$

where $r = (x,y,z)^T$, $\phi(r)$ is the electric potential of the molecule in some fixed "laboratory" coordinate system, and $R_1, \ldots, R_N$ are unknown rotations corresponding to the orientation of the molecule at the moment of freezing (elements of SO(3), the group of rotations of the three-dimensional space). Each image is thus formed by projecting ϕ in some random direction and is therefore known as a "projection image."

The "orientation assignment problem" is to estimate the rotations $R_1, \ldots, R_N$ given a finite set of images $P_1, \ldots, P_N$. In the simplified image formation model (3.1), we assume that all projection images P_i correspond to exactly the same molecule (same potential function ϕ) and differ only in the rotation R_i associated to each image, that is, in the imaging orientation that generated each image. In particular, we assume that all images are centered. In practice, each image also contains some unknown shift, which needs to be estimated. However, the estimation of the shifts can be decoupled from the problem of estimating the rotations (see [18] for details).

To reproduce the Fourier projection-slice theorem and to set up the notation for the derivation of the next section, we take the Fourier transform of both sides of (3.1). We use the following definition of the two-dimensional Fourier transform:

$$\hat{f}(\omega_x, \omega_y) = \iint_{\mathbb{R}^2} f(x,y) e^{-\imath(x\omega_x + y\omega_y)}\, dxdy.$$

For the three-dimensional case, we use

$$\hat{f}(\omega_x, \omega_y, \omega_z) = \iiint_{\mathbb{R}^3} f(x,y,z) e^{-\imath(x\omega_x + y\omega_y + z\omega_z)}\, dxdydz.$$

Also, we denote the three columns of the rotation matrix R_i by $R_i^{(1)}$, $R_i^{(2)}$, and $R_i^{(3)}$, and the dot product between two vectors $u, v \in \mathbb{R}^3$ by $\langle u, v\rangle$. By taking the two-dimensional Fourier transform of both sides of (3.1) and using the notation $\omega = (\omega_x, \omega_y, 0)$ and $u = R_i r$, we obtain

$$\begin{aligned}
\hat{P}_i(\omega_x, \omega_y) &= \iint_{\mathbb{R}^2} P_i(x,y) e^{-\imath(x\omega_x + y\omega_y)}\, dxdy \\
&= \iint_{\mathbb{R}^2} \left(\int_{-\infty}^{\infty} \phi\left(R_i r\right) e^{-\imath(x\omega_x + y\omega_y)}\, dz \right) dxdy \\
&= \iiint_{\mathbb{R}^3} \phi(u) e^{-\imath\langle \omega, R_i^T u\rangle}\, dxdydz = \iiint_{\mathbb{R}^3} \phi(u) e^{-\imath\langle R_i\omega, u\rangle}\, dxdydz \\
&= \hat{\phi}(R_i\omega) = \hat{\phi}\left(\omega_x R_i^{(1)} + \omega_y R_i^{(2)}\right).
\end{aligned} \tag{3.2}$$

Equation (3.2) states that the two-dimensional Fourier transform of a projection image equals a planar slice from the three-dimensional Fourier transform of the molecule. Specifically, it shows that the Fourier transform $\hat{P}_i$ of the projection P_i is equal to the values of $\hat{\phi}$ (the

Fourier transform of the molecule ϕ) on the plane $\omega_x R_i^{(1)} + \omega_y R_i^{(2)}$ or, equivalently, on the plane $\langle \omega, R_i^{(3)} \rangle = 0$. This is the well-known Fourier projection-slice theorem.

Thus, the Fourier transforms $\hat{P}_i$ and $\hat{P}_j$ of any two projection images P_i and P_j correspond to two different planar sections (through the origin) from $\hat{\phi}$, and therefore there exists a line that is common to both planes. We will derive this in detail below. We refer to this line as the "common line" between P_i and P_j. We denote by $q_{ij} \in \mathbb{R}^3$ a unit vector in the direction of the common line between P_i and P_j and express q_{ij} in terms of the rotations R_i and R_j corresponding to the projections P_i and P_j. Consider the unit vector

$$q_{ij} = \left(R_i^{(3)} \times R_j^{(3)} \right) / \| R_i^{(3)} \times R_j^{(3)} \|, \tag{3.3}$$

that is, the normalized cross product between the third columns of the matrices R_i and R_j. By definition of the cross product, this vector is orthogonal to both $R_i^{(3)}$ and $R_j^{(3)}$, that is, $\langle q_{ij}, R_i^{(3)} \rangle = 0$ and $\langle q_{ij}, R_j^{(3)} \rangle = 0$, and therefore it belongs to the planes whose equations are given by $\langle \omega, R_i^{(3)} \rangle = 0$ and $\langle \omega, R_j^{(3)} \rangle = 0$.

We can write the unit vector q_{ij} explicitly in the orthonormal bases $\{R_i^{(1)}, R_i^{(2)}\}$ and $\{R_j^{(1)}, R_j^{(2)}\}$ of the planes $\langle \omega, R_i^{(3)} \rangle = 0$ and $\langle \omega, R_j^{(3)} \rangle = 0$, respectively, as

$$q_{ij} = \cos \alpha_{ij} R_i^{(1)} + \sin \alpha_{ij} R_i^{(2)} = \cos \alpha_{ji} R_j^{(1)} + \sin \alpha_{ji} R_j^{(2)} \tag{3.4}$$

for some angles α_{ij} and α_{ji}. Thus, along the three-dimensional line whose direction vector is q_{ij}, we get from (3.2) that for any $\xi \in \mathbb{R}$

$$\begin{aligned} \hat{P}_i(\xi \cos \alpha_{ij}, \xi \sin \alpha_{ij}) &= \hat{\phi} \left(\xi \cos \alpha_{ij} R_i^{(1)} + \xi \sin \alpha_{ij} R_i^{(2)} \right) \\ &= \hat{\phi}(\xi q_{ij}) \\ &= \hat{\phi} \left(\xi \cos \alpha_{ji} R_j^{(1)} + \xi \sin \alpha_{ji} R_j^{(2)} \right) = \hat{P}_j(\xi \cos \alpha_{ji}, \xi \sin \alpha_{ji}), \end{aligned} \tag{3.5}$$

which is an explicit statement of the common line property described above.

Following (3.4) and (3.5), we parameterize the common line between the projections P_i and P_j by the unit vector $c_{ij} = (\cos \alpha_{ij}, \sin \alpha_{ij})^T$ in P_i, and by $c_{ji} = (\cos \alpha_{ji}, \sin \alpha_{ji})^T$ in P_j. With a slight abuse of notation, we consider these vectors as vectors in $\mathbb{R}^3$ by zero padding, that is,

$$c_{ij} = (\cos \alpha_{ij}, \sin \alpha_{ij}, 0)^T, \qquad c_{ji} = (\cos \alpha_{ji}, \sin \alpha_{ji}, 0)^T. \tag{3.6}$$

Equation (3.4) can then be written as

$$R_i c_{ij} = q_{ij} = R_j c_{ji}. \tag{3.7}$$

The vectors c_{ij} and c_{ji} of (3.6), as well as the common line property of (3.7), are illustrated in Figure 1.

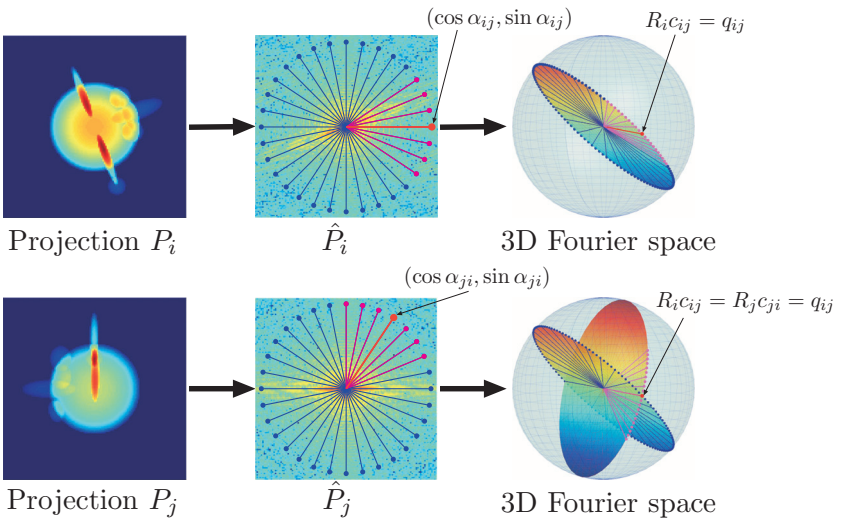


Figure 1. *Fourier projection-slice theorem and the common line property.*

Note that the vector $-q_{ij}$ is also a vector in the direction of the common line between P_i and P_j. However, this poses no problem since we first choose q_{ij} in (3.3) and then choose α_{ij} and α_{ji} accordingly in (3.4). Note, however, that if

$$c_{ij} = (\cos\alpha_{ij}, \sin\alpha_{ij}, 0)^T, \qquad c_{ji} = (\cos\alpha_{ji}, \sin\alpha_{ji}, 0)^T$$

are the common lines between the projections P_i and P_j, then so are

$$\tilde{c}_{ij} = (\cos(\alpha_{ij}+\pi), \sin(\alpha_{ij}+\pi), 0)^T, \quad \tilde{c}_{ji} = (\cos(\alpha_{ji}+\pi), \sin(\alpha_{ji}+\pi), 0)^T, \tag{3.8}$$

since the common line between two projections is a line through the origin whose direction vector can be assigned to be either c_{ij} or $\tilde{c}_{ij}$. Since the last coordinate of c_{ij} and $\tilde{c}_{ij}$ is zero, we get

$$R_i\tilde{c}_{ij} = -R_i c_{ij} = -q_{ij}. \tag{3.9}$$

That is, both vectors q_{ij} and $-q_{ij}$ correspond to the same line through the origin in three-dimensional space, and an arbitrary choice between the two dictates the choice between the angles $(\alpha_{ij}, \alpha_{ji})$ and $(\alpha_{ij}+\pi, \alpha_{ji}+\pi)$ in (3.4).

4. Global synchronization. We next use (3.7) to construct a symmetric matrix of size $2N\times 2N$, whose three-dimensional nontrivial eigenspace encodes the rotations R_i, $i = 1, \ldots, N$, of (3.1). We refer to this matrix as the "synchronization matrix" for a reason that will be explained below.

First, we show that we can compute the upper-left 2×2 block of the unknown matrix $R_i^{-1}R_j$ for any i and j by using only common line information. For given i and j with $i \neq j$, pick some arbitrary k from the range $1, \ldots, N$ with $k \neq i$ and $k \neq j$, and define the matrix $Q_{ijk} \in \mathbb{R}^{3\times 3}$ by

$$Q_{ijk} = (q_{ij}, q_{ik}, q_{jk}), \tag{4.1}$$

where q_{ij}, q_{ik}, and q_{jk} are defined in (3.3). By combining (3.7) with the orthogonality of R_i and R_j we get that the matrix $G_{ijk} = Q_{ijk}^T Q_{ijk}$ can be expressed as

$$G_{ijk} = \begin{pmatrix} 1 & \langle q_{ij}, q_{ik}\rangle & \langle q_{ij}, q_{jk}\rangle \\ \langle q_{ik}, q_{ij}\rangle & 1 & \langle q_{ik}, q_{jk}\rangle \\ \langle q_{jk}, q_{ij}\rangle & \langle q_{jk}, q_{ik}\rangle & 1 \end{pmatrix} = \begin{pmatrix} 1 & \langle c_{ij}, c_{ik}\rangle & \langle c_{ji}, c_{jk}\rangle \\ \langle c_{ik}, c_{ij}\rangle & 1 & \langle c_{ki}, c_{kj}\rangle \\ \langle c_{jk}, c_{ji}\rangle & \langle c_{kj}, c_{ki}\rangle & 1 \end{pmatrix}. \tag{4.2}$$

Equation (4.2) shows that the matrix G_{ijk} can be constructed using only common line information between the triplet of projections P_i, P_j, and P_k. Applying the Cholesky decomposition or the SVD [6] to G_{ijk} gives the factorization $G_{ijk} = Q_{ijk}^T Q_{ijk}$, thus recovering the matrix Q_{ijk} of (4.1), whose columns are three-dimensional unit vectors in the directions of the lines of intersection between the Fourier planes corresponding to the three projections. This decomposition is not unique, as any matrix of the form $\tilde{Q}_{ijk} = O_{ijk} Q_{ijk}$, where O_{ijk} is an orthogonal 3×3 matrix, also satisfies $G_{ijk} = \tilde{Q}_{ijk}^T \tilde{Q}_{ijk}$.

We distinguish between two cases: $\det(O_{ijk}) = 1$ and $\det(O_{ijk}) = -1$. In the first case, O_{ijk} is a rotation. We define the matrices

$$C_i = (c_{ij}, c_{ik}, c_{ij} \times c_{ik}), \quad Q_i = (q_{ij}, q_{ik}, q_{ij} \times q_{ik}), \tag{4.3}$$

$$C_j = (c_{ji}, c_{jk}, c_{ji} \times c_{jk}), \quad Q_j = (q_{ji}, q_{jk}, q_{ji} \times q_{jk}), \tag{4.4}$$

and since for any rotation matrix R and any vectors u and v we have that $R(u \times v) = Ru \times Rv$, we get from (3.7), (4.3) and (4.4) that $R_i C_i = Q_i$ and $R_j C_j = Q_j$ or, equivalently,

$$R_i = Q_i C_i^{-1}, \quad R_j = Q_j C_j^{-1}. \tag{4.5}$$

Thus,

$$R_i^{-1} R_j = C_i Q_i^{-1} Q_j C_j^{-1}. \tag{4.6}$$

Deriving (4.6) requires the unknown matrix Q_{ijk} of (4.1) in order to construct the matrices Q_i and Q_j in (4.3) and (4.4). Instead, by factoring G_{ijk} in (4.2) we obtain the matrix $\tilde{Q}_{ijk} = O_{ijk} Q_{ijk}$, whose columns are $O_{ijk} q_{ij}$, $O_{ijk} q_{ik}$, and $O_{ijk} q_{jk}$ for some unknown rotation O_{ijk}. Thus, we can only construct the matrices

$$\tilde{Q}_i = (O_{ijk} q_{ij}, O_{ijk} q_{ik}, O_{ijk} q_{ij} \times O_{ijk} q_{ik}) = O_{ijk} Q_i, \tag{4.7}$$

$$\tilde{Q}_j = (O_{ijk} q_{ji}, O_{ijk} q_{jk}, O_{ijk} q_{ji} \times O_{ijk} q_{jk}) = O_{ijk} Q_j. \tag{4.8}$$

Using these matrices, we recover the rotations

$$\tilde{R}_i = \tilde{Q}_i C_i^{-1} = O_{ijk} Q_i C_i^{-1}, \qquad \tilde{R}_j = \tilde{Q}_j C_j^{-1} = O_{ijk} Q_j C_j^{-1},$$

which are different from those in (4.5). However, it holds that

$$\tilde{R}_i^{-1} \tilde{R}_j = C_i Q_i^{-1} O_{ijk}^{-1} O_{ijk} Q_j C_j^{-1} = C_i Q_i^{-1} Q_j C_j^{-1} = R_i^{-1} R_j, \tag{4.9}$$

and so this product is invariant to the arbitrary rotation matrix O_{ijk}.

At this point, before considering the case where $\det(O_{ijk}) = -1$, we explain the intuition behind the construction, as well as the name "synchronization matrix." Suppose we were able to use (4.9) to compute all 3×3 matrices $R_i^{-1}R_j$, $i, j = 1, \ldots, N$. Then we could define the $3N \times 3N$ matrix S, consisting of $N \times N$ blocks of size 3×3, where the (i, j) block of S equals $R_i^{-1}R_j$. Such a matrix of ratios of group elements is known in the literature as a synchronization matrix [15]. Now consider the $3 \times 3N$ matrix denoted U, consisting of N blocks of size 3×3, whose jth block equals the (unknown) matrix R_j. The matrix S is then given by $S = U^T U$; that is, in the noise-free case, S is of rank 3. Moreover, all rotations can be extracted from the SVD decomposition of S. Unfortunately, we cannot always recover $R_i^{-1}R_j$ from common line information, as we see next, the reason being that the determinant of the unknown arbitrary matrix O_{ijk} can be either $(+1)$ or (-1).

In the case where $\det(O_{ijk}) = -1$, the matrix $\tilde{Q}_i$ from (4.7) is given by

$$\begin{aligned} \tilde{Q}_i &= (O_{ijk}q_{ij}, O_{ijk}q_{ik}, O_{ijk}q_{ij} \times O_{ijk}q_{ik}) \\ &= (O_{ijk}q_{ij}, O_{ijk}q_{ik}, -O_{ijk}\,(q_{ij} \times q_{ik})) \qquad (4.10) \\ &= O_{ijk}Q_iJ, \qquad (4.11) \end{aligned}$$

where

$$J = \begin{pmatrix} 1 & & \\ & 1 & \\ & & -1 \end{pmatrix},$$

and in (4.10) we have used the fact that for any orthogonal matrix O with $\det(O) = -1$ and any two vectors u and v, it holds that $Ou \times Ov = -O\,(u \times v)$. Similarly, the matrix $\tilde{Q}_j$ from (4.8) is given by

$$\tilde{Q}_j = O_{ijk}Q_jJ.$$

The matrices $\tilde{R}_i$ and $\tilde{R}_j$ are then given by

$$\tilde{R}_i = O_{ijk}Q_iJC_i^{-1}, \qquad \tilde{R}_j = O_{ijk}Q_jJC_j^{-1},$$

and since $J = J^{-1}$, the product $\tilde{R}_i^{-1}\tilde{R}_j$ is given by

$$\tilde{R}_i^{-1}\tilde{R}_j = C_iJQ_i^{-1}Q_jJC_j^{-1}.$$

Since the third coordinate in all vectors c_{ij} is zero, by a direct calculation we get that $C_iJ = JC_i$ and $JC_i^{-1} = C_i^{-1}J$. Thus, from (4.6)

$$\tilde{R}_i^{-1}\tilde{R}_j = JC_iQ_i^{-1}Q_jC_j^{-1}J = JR_i^{-1}R_jJ. \qquad (4.12)$$

Equations (4.9) and (4.12) state that using common line data between a triplet of projections (P_i, P_j, P_k), we can recover either the product $R_i^{-1}R_j$ from (4.6) or the product $JR_i^{-1}R_jJ$ from (4.12). Moreover, there is no way to distinguish which of the two is obtained, since O_{ijk} is unknown. This is a manifestation of the well-known fact in cryo-EM reconstruction which states that the handedness or the chirality of the molecule cannot be deduced from common line information. In other words, there exist two different sets of rotations in (3.7) that satisfy all common line constraints; the two sets result in two different reconstructions.

However, the sum $R_i^{-1}R_j + JR_i^{-1}R_jJ$ is invariant under this ambiguity; that is, no matter if we recover the product $R_i^{-1}R_j$ or $JR_i^{-1}R_jJ$, the sum $R_i^{-1}R_j + JR_i^{-1}R_jJ$ remains the same. To understand this sum, we write the rotation matrix $R_i^{-1}R_j$ explicitly as

$$R_i^{-1}R_j = \begin{pmatrix} r_{11}^{(ij)} & r_{12}^{(ij)} & r_{13}^{(ij)} \\ r_{21}^{(ij)} & r_{22}^{(ij)} & r_{23}^{(ij)} \\ r_{31}^{(ij)} & r_{32}^{(ij)} & r_{33}^{(ij)} \end{pmatrix},$$

from which we get by a direct calculation that

$$R_i^{-1}R_j + JR_i^{-1}R_jJ = \begin{pmatrix} 2r_{11}^{(ij)} & 2r_{12}^{(ij)} & 0 \\ 2r_{21}^{(ij)} & 2r_{22}^{(ij)} & 0 \\ 0 & 0 & 2r_{33}^{(ij)} \end{pmatrix}. \tag{4.13}$$

We define the matrix $B_{ij}^{(k)}$ as the upper-left 2×2 block of $\left(R_i^{-1}R_j + JR_i^{-1}R_jJ\right)/2$, namely

$$B_{ij}^{(k)} = \begin{pmatrix} r_{11}^{(ij)} & r_{12}^{(ij)} \\ r_{21}^{(ij)} & r_{22}^{(ij)} \end{pmatrix}, \tag{4.14}$$

where the division by 2 eliminates the spurious factor of 2 in (4.13). The superscript k of $B_{ij}^{(k)}$ is used to indicate that constructing this matrix required a triplet of projections P_i, P_j, and P_k (see (4.3), (4.4), and (4.6)). Since $R_j^{-1}R_i = (R_i^{-1}R_j)^T$ we get that $B_{ji}^{(k)} = (B_{ij}^{(k)})^T$. In practice, due to common line misidentifications, each block $B_{ij}^{(k)}$ might contain some error. We therefore improve the estimate of the 2×2 upper-left block of the matrix $\left(R_i^{-1}R_j + JR_i^{-1}R_jJ\right)/2$ by averaging the matrices $B_{ij}^{(k)}$ over all k; that is, we define

$$\begin{aligned} S_{ij} &= \frac{1}{N-2}\sum_{k \neq i,j} B_{ij}^{(k)}, \quad i \neq j, \\ S_{ii} &= I_2, \end{aligned} \tag{4.15}$$

where $i, j = 1, \ldots, N$ and I_2 is the 2×2 identity matrix. The synchronization matrix S is then defined as the $2N \times 2N$ matrix whose (i, j) block of size 2×2 is S_{ij}. The matrix S is symmetric and is easily checked to admit the decomposition $S = H^TH$, where H is the $3 \times 2N$ matrix obtained by concatenating the first two columns of all N rotation matrices, namely,

$$H = \left(R_1^{(1)}, R_1^{(2)}, \ldots, R_N^{(1)}, R_N^{(2)}\right). \tag{4.16}$$

Therefore, the matrix S is of rank 3, and its columns are linear combinations of the three vectors that are the columns of H^T of (4.16). Thus, the three nontrivial eigenvectors of S are also linear combinations of the columns of H^T or, equivalently, of the first two columns of $R_1, \ldots, R_N$.

In practice, we do not average in (4.15) over all $k \neq i, j$, since in the presence of noise, the common lines detected among triplets of projections do not necessarily correspond to any physical assignment of orientations to the three projections. This is explained in detail in [16]. Thus, we replace the averaging in (4.15) with an average based on a voting procedure, following the approach in [16]. Namely, we average in (4.15) only over matrices $B_{ij}^{(k)}$ such that the common lines between projections P_i, P_j, and P_k have passed the voting procedure.

Once the three nontrivial eigenvectors of S have been computed, we need to unmix the columns of H from these eigenvectors. The third column of each rotation matrix is then given by the cross product of the first two columns. We denote the three nontrivial eigenvectors of the matrix S by v_1, v_2, v_3, each of length $2N$, and define the $2N \times 3$ matrix V by $V = (v_1, v_2, v_3)$. For an arbitrary 3×3 matrix A, the columns of the matrix VA^T are linear combinations of the columns of V, that is, of the eigenvectors v_1, v_2, and v_3. We are looking for the linear combination that will recover H^T; namely, we are looking for A such that $VA^T = H^T$ or, equivalently, $AV^T = H$. For convenience, we split V^T into two matrices V_1 and V_2, each of size $3 \times N$, consisting of the odd and even columns of V^T, respectively,

$$V_1 = \begin{pmatrix} v_1^{(1)} & v_1^{(3)} & \cdots & v_1^{(2N-1)} \\ v_2^{(1)} & v_2^{(3)} & \cdots & v_2^{(2N-1)} \\ v_3^{(1)} & v_3^{(3)} & \cdots & v_3^{(2N-1)} \end{pmatrix}, \qquad V_2 = \begin{pmatrix} v_1^{(2)} & v_1^{(4)} & \cdots & v_1^{(2N)} \\ v_2^{(2)} & v_2^{(4)} & \cdots & v_2^{(2N)} \\ v_3^{(2)} & v_3^{(4)} & \cdots & v_3^{(2N)} \end{pmatrix}.$$

The unmixing problem is to find a 3×3 matrix A such that

$$AV_1 = \left(R_1^{(1)}, R_2^{(1)}, \ldots, R_N^{(1)}\right), \qquad AV_2 = \left(R_1^{(2)}, R_2^{(2)}, \ldots, R_N^{(2)}\right). \tag{4.17}$$

The orthogonality relations among the columns of the rotation matrices $R_1, \ldots, R_N$ imply that

$$\left(R_i^{(1)}\right)^T R_i^{(1)} = 1, \qquad \left(R_i^{(2)}\right)^T R_i^{(2)} = 1, \qquad \left(R_i^{(1)}\right)^T R_i^{(2)} = 0, \qquad i = 1, \ldots, N,$$

or, using (4.17), that

$$(V_1^T A^T A V_1)_{ii} = 1, \qquad (V_2^T A^T A V_2)_{ii} = 1, \qquad (V_1^T A^T A V_2)_{ii} = 0 \tag{4.18}$$

for $i = 1, \ldots, N$. Equation (4.18) gives $3N$ linear equations for the six unknowns of $A^T A$, which can be solved, for example, by least-squares. Once the entries of $A^T A$ have been computed, the matrix A can be recovered using Cholesky decomposition or SVD, and (4.17) will recover the required rotations.

5. The limit of the matrix S. In this section we show that as N goes to infinity, the matrix S in (4.15) (normalized by N) converges to an integral operator over SO(3). Since we know from (4.15) and the discussion thereafter that the first two columns of the rotation matrices R_i form the nontrivial eigenspace of S, we can explicitly construct the eigenfunctions of this integral operator. Then, by a direct calculation, we show that the top three eigenvalues of the integral operator equal $2/3$. In particular, for large N, the nontrivial eigenvalues of the matrix S of (4.15) are approximately $2N/3$.

Without noise, all blocks $B_{ij}^{(k)}$ from (4.14) are the same for all k and are equal to S_{ij} of (4.15). We rewrite S_{ij} as $S(R_i, R_j)$ to express explicitly its dependence on R_i and R_j. Specifically,

$$
\begin{aligned}
S(R_i, R_j) = S_{ij} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R_i^{-1} R_j \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^T \quad \text{for } i \neq j, \\
S(R_i, R_i) = B_{ii} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{for } i = j.
\end{aligned} \tag{5.1}
$$

We will analyze the multiplication of S by a vector f of length $2N$ as N goes to infinity. For convenience, we will consider f as the concatenation of N vectors in $\mathbb{R}^2$, each obtained by sampling some function $f : \mathrm{SO}(3) \to \mathbb{R}^2$ at R_i, $i = 1, \ldots, N$, that is,

$$
f_i = f(R_i), \qquad i = 1, \ldots, N.
$$

The ith block of size 2×1 of the product Sf is then given by

$$
(Sf)_i = \sum_{j=1}^{N} S(R_i, R_j) f(R_j), \qquad i = 1, \ldots, N. \tag{5.2}
$$

If the rotations are independent samples from the uniform distribution on $\mathrm{SO}(3)$, then the terms $S(R_i, R_j) f(R_j)$ are independent and identically distributed random variables, and by the law of large numbers

$$
\lim_{N \to \infty} \frac{1}{N} (Sf)(R) = \mathbb{E}\left[S(R, U) f(U)\right] = \int_{\mathrm{SO}(3)} S(R, U) f(U) dU, \tag{5.3}
$$

where dU is the Haar measure. That is, in the limit $N \to \infty$, the matrix S converges to the integral operator

$$
(\mathcal{S}f)(R) = \int_{\mathrm{SO}(3)} S(R, U) f(U) dU, \tag{5.4}
$$

where $S(R, U)$ is the kernel of the operator, given by (5.1), and $f : \mathrm{SO}(3) \to \mathbb{R}^2$.

We will compute the nontrivial eigenvalues of $\mathcal{S}$. We define the function $g : \mathrm{SO}(3) \to \mathbb{R}^{2 \times 3}$ by

$$
g(R) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R^{-1}.
$$

The function $g(R)$ extracts the first two rows of the rotation matrix R^{-1}, that is, the first two columns of R. This function can be considered as a concatenation of three functions from $\mathrm{SO}(3)$ to $\mathbb{R}^2$, and thus the operator $\mathcal{S}$ operates on such a function column by column. We will show that $g(R)$ is an eigenfunction of the integral operator $\mathcal{S}$. This will show that the first two columns of all rotation matrices are eigenfunctions of $\mathcal{S}$.

By a direct evaluation of the integrand in (5.4), we get using the notation $e_3 = (0,0,1)^T$ that

$$\begin{aligned} S(R,U)g(U) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R^{-1}U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^T \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} U^{-1} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R^{-1}U(I - e_3 e_3^T)U^{-1} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R^{-1} \left(I - U^{(3)} {U^{(3)}}^T \right), \end{aligned}$$

where $U^{(3)}$ is the third column of the matrix U. By denoting the unit vector $U^{(3)}$ as

$$U^{(3)} = (x,y,z)^T, \qquad x^2 + y^2 + z^2 = 1,$$

we get that

$$U^{(3)} {U^{(3)}}^T = \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix},$$

and

$$\begin{aligned} (\mathcal{S}g)(R) &= \int_{\mathrm{SO}(3)} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R^{-1} \left(I - U^{(3)} {U^{(3)}}^T \right) dU \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R^{-1} \int_{\mathrm{SO}(3)} \left(I - U^{(3)} {U^{(3)}}^T \right) dU \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R^{-1} \int_{\mathrm{SO}(3)} \begin{pmatrix} 1-x^2 & -xy & -xz \\ -xy & 1-y^2 & -yz \\ -xz & -yz & 1-z^2 \end{pmatrix} dU. \end{aligned} \tag{5.5}$$

To calculate (5.5), we parameterize the rotation $U \in SO(3)$ of the molecule by a viewing direction and a rotation around that axis; that is, we write $\mathrm{SO}(3)$ as $S^2 \times [0, 2\pi)$. We choose the measure $d\mu$ on S^2 such that $\int_{S^2} d\mu = 1$. Since the integral in (5.5) is only a function of $U^{(3)}$, the integral collapses to an integral over S^2. From symmetry of S^2 it follows that

$$\int_{S^2} xy \, d\mu = \int_{S^2} xz \, d\mu = \int_{S^2} yz \, d\mu = 0$$

and that

$$\int_{S^2} x^2 d\mu = \int_{S^2} y^2 d\mu = \int_{S^2} z^2 d\mu.$$

Since

$$\int_{S^2} x^2 d\mu + \int_{S^2} y^2 d\mu + \int_{S^2} z^2 d\mu = \int_{S^2} \left(x^2 + y^2 + z^2 \right) d\mu = \int_{S^2} d\mu = 1,$$

we get that

$$\int_{S^2} x^2 d\mu = \int_{S^2} y^2 d\mu = \int_{S^2} z^2 d\mu = 1/3.$$

We conclude that (see (5.5))

$$\int_{\mathrm{SO}(3)} \begin{pmatrix} 1-x^2 & -xy & -xz \\ -xy & 1-y^2 & -yz \\ -xz & -yz & 1-z^2 \end{pmatrix} dU = \begin{pmatrix} 2/3 & 0 & 0 \\ 0 & 2/3 & 0 \\ 0 & 0 & 2/3 \end{pmatrix},$$

and so

$$(\mathcal{S}g)(R) = \frac{2}{3} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R^{-1} = \frac{2}{3} g(R),$$

where the multiplicity of this eigenspace is three as $g(R)$ has three columns in $\mathbb{R}^2$.

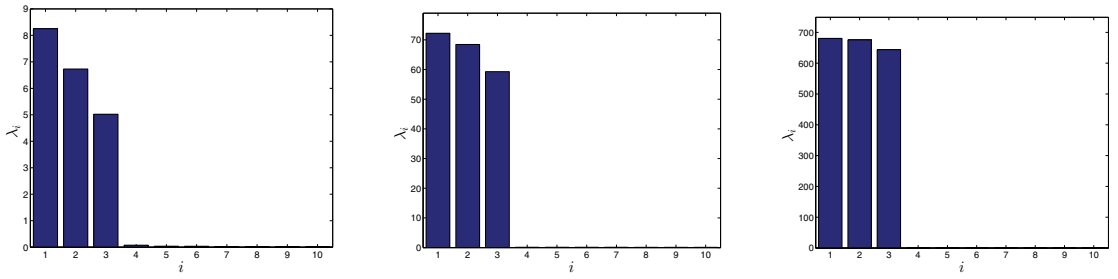
The above calculation shows that the spectral gap of the operator is $2/3$. This gap is larger than the gap of $5/12 \approx 0.4167$ of the algorithm in [17], and it is therefore expected that the algorithm in this paper will be more robust to noise than the one in [17].

6. Numerical experiments. The algorithm described in section 4 was implemented in MATLAB and was tested on simulated projections as well as on experimental class averages of the 50S subunit of the *E. coli* ribosome.

6.1. Simulated data. First, we applied the algorithm in a noiseless environment to demonstrate the properties of the matrix S given by (4.15). To that end, we created three simulated data sets by generating three sets of noiseless images, with $N = 10$, $N = 100$, and $N = 1000$ projections. The images in each set are projections of a three-dimensional density map of the 50S subunit of the *E. coli* ribosome, at orientations drawn uniformly at random from the uniform distribution on $\mathrm{SO}(3)$. Each image is of size 129×129. For each image, we computed $L = 360$ radial Fourier lines, with $n = 100$ samples per Fourier line. The common line between two projections was estimated by computing the correlation between all 360 lines from one image and all 360 lines from the other image, and taking the common line to be the pair with maximum normalized cross-correlation.

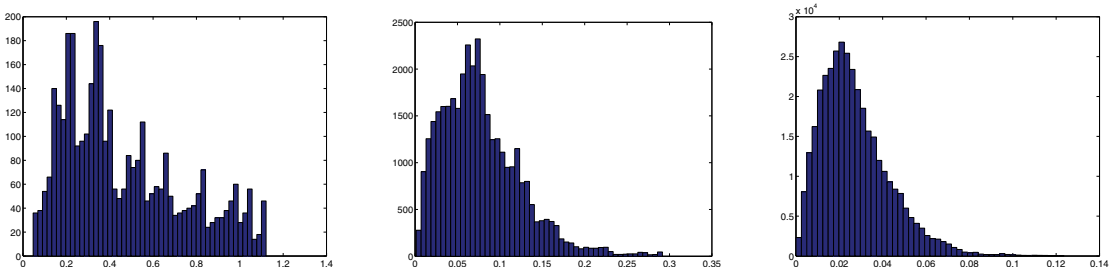
Figures 2(a)–2(c) show the largest 10 eigenvalues of the matrix S for $N = 10$, $N = 100$, and $N = 1000$, respectively. It is evident that the matrix S has three dominant eigenvalues. Note that the fourth and higher eigenvalues are not exactly zero due to the discretization of Fourier space using $L = 360$, but are rather much smaller than the first three. As mentioned in section 5, the top eigenvalue of the matrix S should approximately equal $2N/3$. This is in agreement, for example, with the first three eigenvalues in Figure 2(c), whose values are 681, 676, and 644 (rounded to the nearest integer), whereas the expected value for $N = 1000$ is 667.

Figures 3(a)–3(c) show the error histogram of the recovered orientations. To generate this histogram, we first estimated the rotations from the matrix S as described in section 4. We denote the estimated rotations by $\tilde{R}_1, \ldots, \tilde{R}_N$, to distinguish them from the true rotations $R_1, \ldots, R_N$ used for generating the simulated projections. As described above, for a projection P_i, we sampled its Fourier transform $\hat{P}_i$ along L radial lines, specifically, along the directions $c_i^{(l)} = (\cos\ 2\pi l/L, \sin\ 2\pi l/L)$, $l = 0, \ldots, L-1$, with $n = 100$ samples in each direction, centered around the origin. As before, for convenience, we lift these vectors to three dimensions; that is, we define $c_i^{(l)} = (\cos\ 2\pi l/L, \sin\ 2\pi l/L, 0)$, $l = 0, \ldots, L-1$. Then, we compute for each pair of i and l, where $i = 1, \ldots, N$, $l = 0, \ldots, L-1$, the angle (in degrees) between $R_i c_i^{(l)}$ and $\tilde{R}_i c_i^{(l)}$. This gives the error in estimating the three-dimensional position of Fourier ray $c_i^{(l)}$.



(a) $N = 10$ (b) $N = 100$ (c) $N = 1000$

Figure 2. *Applying the algorithm of section 4 on noiseless projections. Ten largest eigenvalues of the matrix S in (4.15) for* (a) $N = 10$, (b) $N = 100$, *and* (c) $N = 1000$ *images. The x-axis corresponds to the index of the eigenvalue. The y-axis corresponds to its value.*



(a) $N = 10$ (b) $N = 100$ (c) $N = 1000$

Figure 3. *Histogram of angle estimation errors (in degrees) for* (a) $N = 10$, (b) $N = 100$, *and* (c) $N = 1000$ *noiseless images. The x-axis corresponds to the estimation error in degrees.*

Since the true and estimated rotations may differ by an arbitrary global rotation, we first register the two sets of three-dimensional vectors $R_i c_i^{(l)}$ and $\tilde{R}_i c_i^{(l)}$, $i = 1, \ldots, N$, $l = 0, \ldots, L-1$, as described in [1]. The histogram of the estimation errors (after registration) for all pairs of i and l is shown in Figures 3(a), 3(b), and 3(c) for $N = 10$, $N = 100$, and $N = 1000$, respectively. We see that the estimation error is very small even for $N = 10$ and decreases 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 projections, but rather some approximation of it accurate to within $360/(2L)$ degrees, which in our case of $L = 360$ equals 0.5 degrees (assuming correlation always finds the correct common lines between noiseless projections). In other words, each of the equations in (3.7) contains some small noise. Also note that the maximal estimation errors are 1.12, 0.29, and 0.12 degrees for $N = 10$, $N = 100$, and $N = 1000$, respectively, and the average estimation errors are 0.46, 0.0078, and 0.0027 degrees. This is less than the angular resolution used of 1 degree. Thus, without noise the algorithm estimates the rotations $R_1, \ldots, R_N$ to very high accuracy, where the only errors are due to the angular discretization of $L = 360$.

All tests were executed on a dual Intel Xeon X5560 CPU (8 cores in total), with 48GB of RAM running Linux. Whenever possible, all 8 cores were used simultaneously, either

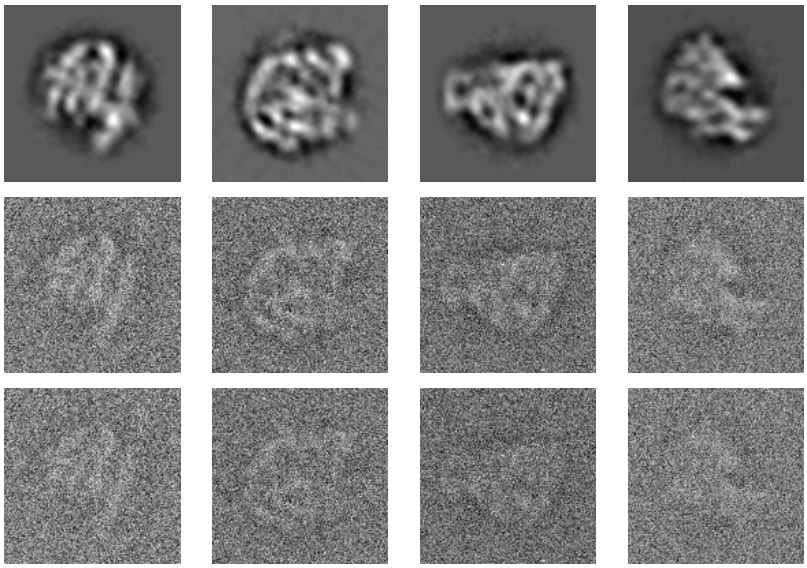
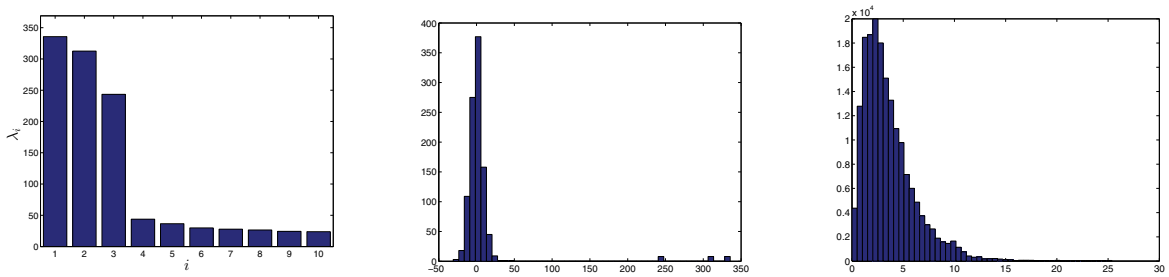


Figure 4. *Top row: Sample of* 4 *simulated noiseless projections of the* $50S$ *subunit of the* E. coli *ribosome. Middle row: Noisy realizations at SNR* $= 1/16$. *Bottom row: Noisy realizations at SNR* $= 1/32$. *The noisy projections are corrupted with additive Gaussian white noise with the prescribed SNR.*

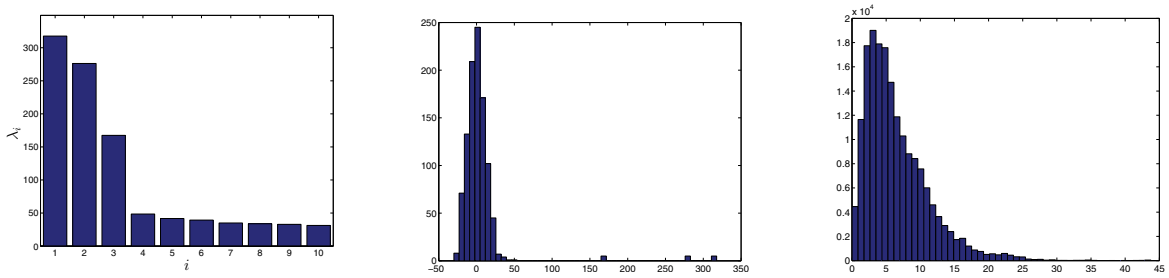
explicitly using the MATLAB `parfor` or implicitly, by taking advantage of the MATLAB implementation of BLAS, which takes advantage of multicore computing. For each experiment we measured the time required to construct the matrix S from (4.15) once the common line matrix has been computed. The time required to find the top three eigenvectors of S and to unmix the rotations from those eigenvectors is negligible. The time required for constructing S for $N = 10$, $N = 100$, and $N = 1000$ was 0.25, 20.6, and 16277 seconds, respectively. The running time of the algorithm scales cubically with N but is acceptable for practical values of N. Moreover, it can be easily parallelized, as each block S_{ij} in (4.15) can be computed independently.

Next, we tested the algorithm on two sets of noisy projections. We took 500 noiseless projections of the same density map as above and corrupted them with additive Gaussian white noise with SNR $= 1/16$ and SNR $= 1/32$, where SNR (signal-to-noise ratio) is defined as the ratio between the energy (variance) of the signal and the energy of the noise. Figure 4 shows four clean projections (top), their noisy realizations at SNR $= 1/16$ (middle row), and their noisy realizations at SNR $= 1/32$ (bottom row). We then applied the algorithm to each of the noisy data sets. The output of the algorithm is illustrated in Figure 5 for SNR $= 1/16$ and in Figure 6 for SNR $= 1/32$. As can be seen from Figures 5(a) and 6(a), in the presence of noise, the matrix S still has three dominant eigenvalues, but the remaining eigenvalues are no longer small due to misidentifications of common lines. Still, there is a significant gap between the third and fourth eigenvalues. Figures 5(b) and 6(b) show the histogram of the eigenvalues, where the gap between the largest three eigenvalues and the remaining bulk of the spectrum is evident but gets smaller as the SNR decreases. The histogram of the angle estimation errors is shown in Figures 5(c) and 6(c), where we see that due to misidentifications



(a) (b) (c)

Figure 5. *Applying the algorithm of section 4 on* 500 *simulated projections at* $SNR = 1/16$ *with* $L = 360$. (a) *Ten largest eigenvalues of the matrix* S. (b) *Histogram of the eigenvalues of* S. *The three largest eigenvalues are clearly separated from the remaining bulk of the spectrum, which is due to misidentifications of common lines (noise).* (c) *Angle estimation errors (in degrees).*

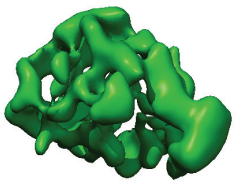


(a) (b) (c)

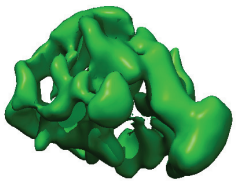
Figure 6. *Applying the algorithm of section 4 on* 500 *simulated projections at* $SNR = 1/32$ *with* $L = 360$. *See Figure 5 for details.*

of common lines (due to noise), the embedding errors become larger. Still, for SNR = 1/16, the mean estimation error is 3.6 degrees, and the embedding error of more than 90% of the Fourier rays is smaller than 10 degrees. For SNR = 1/32, the mean estimation error is 6.55 degrees, and the embedding error of almost 80% of the Fourier rays is smaller than 10 degrees. For SNR = 1/16 only 34% of the common lines were initially identified correctly, where we define a common line as correctly identified if it deviates from the true common line computed according to (3.3) and (3.4) by up to 5 degrees. This means that at least 66% of the blocks S_{ij} are completely wrong (the ones for which the common line between P_i and P_j was misidentified and so all blocks $B_{ij}^{(k)}$ are necessarily wrong), and almost all the other blocks also contain some error, due to the averaging in (4.15) of blocks for which the common lines between P_i and P_k or between P_j and P_k are wrong. For SNR = 1/32 only 18% of the common lines were identified correctly. This demonstrates the impressive robustness to noise of the algorithm.

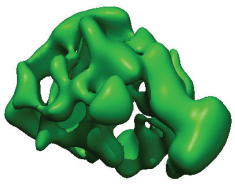
Once the orientations have been estimated, the volume was reconstructed by using the three-dimensional pseudopolar Fourier transform [3, 2]. A two-dimensional rendering of the reconstructed volumes is shown in Figure 7 (all volumes were rendered using UCSF Chimera [11]). All volumes were filtered using a Gaussian filter with $\sigma = 0.9$ pixels. Figures 7(b) and 7(e) show the volume reconstructed from 500 noisy projections and the estimated ori-



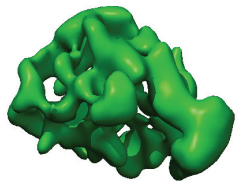
(a) Reference volume.



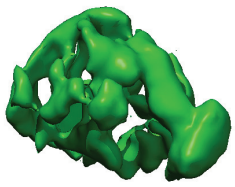
(b) NE 1/16.



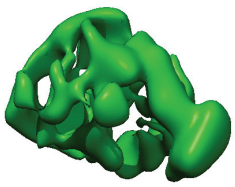
(c) CE 1/16.



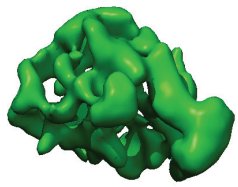
(d) NT 1/16.



(e) NE 1/32.



(f) CE 1/32.



(g) NT 1/32.

Figure 7. *Reconstructed density maps.* (a) *Noiseless density map used to generate the projections. Other figures are labeled as follows: NE, reconstruction from noisy projections and estimated orientations; CE, reconstruction from clean projections and estimated orientations; NT, reconstruction from noisy projections and true orientations.*

entations (for SNR = 1/16 and SNR = 1/32, respectively). For comparison, Figures 7(c) and 7(f) show the reconstruction from the estimated orientations and the noiseless projections, and Figures 7(d) and 7(g) show the reconstruction from noisy projections and the true orientations. The various reconstructions allow one to gauge the effect of noise in the projections versus the effect of errors in estimating the orientations. For reference, we also show in Figure 7(a) the noiseless density map from which the noiseless projections were generated. In Figure 8 we show the Fourier shell correlation curves for the various reconstructions, computed against the reference density map of Figure 7(a). These curves show that the loss of resolution due to errors in estimating the rotations is smaller than that due to noise in the projections.

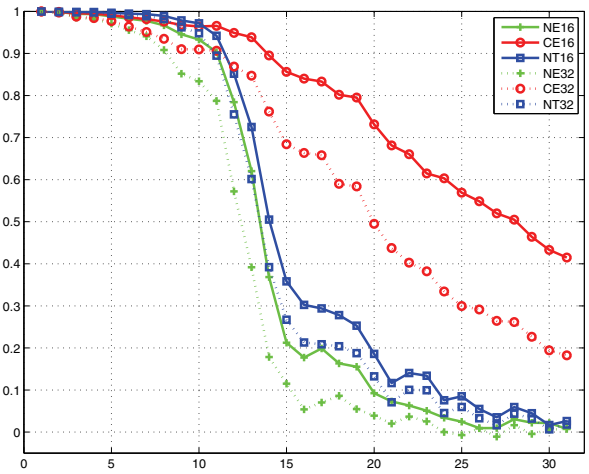


Figure 8. *Fourier shell correlation curves for the various reconstructions. See Figure 7 for a description of the letter codings in the legend.*

Table 1
MSE of the estimated rotations.

SNR	MSE $N = 100$	MSE $N = 500$
1	0.00046	0.00022
1/2	0.00095	0.00060
1/4	0.00234	0.00215
1/8	0.01052	0.00963
1/16	0.05044	0.03626
1/32	0.24335	0.12611
1/64	1.70947	0.61562
1/128	2.74514	2.21980
1/256	4.09763	3.32657
1/512	4.80354	4.66475

To compare the performance of the algorithm with the performance of the algorithms in [17], we show in Table 1 the mean square error (MSE) of the estimated rotations for various levels of noise. Table 1 was generated using $L = 72$ in accordance with the angular resolution in [17]. As above, we denote by $R_1, \ldots, R_N$ the true rotations used to generate the simulated projections, and by $\tilde{R}_1, \ldots, \tilde{R}_N$ the rotations estimated by the algorithm. Then we define

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} \|R_i - \tilde{O}\tilde{R}_i\|_F^2,$$

where $\tilde{O}$ is the global orthogonal transformation that optimally aligns the true and estimated rotations, that is,

$$\tilde{O} = \operatorname*{argmin}_{O \in \mathrm{SO}(3)} \sum_{i=1}^{N} \|R_i - O\tilde{R}_i\|_F^2. \tag{6.1}$$

The solution to (6.1) is given explicitly in [17]. We see from Table 1 that the estimation errors of our algorithm are lower than the errors in Table 5.3 in [17].

6.2. Experimental data. A set of 1500 class averages of the 50S subunit of the *E. coli* ribosome was used to test the algorithm on experimental data. This data set is the same data set used in [16]. The 1500 class averages were generated from a set of 27121 particle images, which were classified using the MSA function of the IMAGIC software package [19, 22] into 1500 classes. Each class average in the data set is of size 90×90 pixels. The raw images used to generate the class averages were phase-flipped to remove the phase-reversals in the contrast transfer function (CTF) and were bandpass filtered at 1/150 Å and 1/8.4 Å. They were also normalized and translationally aligned with the rotationally averaged total sum. Unlike the simulated projections from section 6.1, the resulting class averages are affected by the CTF of the microscope, are contaminated by non-Gaussian noise, and contain unknown two-dimensional shifts. These factors make the detection of common lines much more difficult and also introduce artifacts to the resulting reconstruction.

However, the most severe problem with class averages generated from experimental data is that they do not necessarily correspond to actual projections of the molecule. Ideally, when generating class averages, the class averaging algorithm finds projections that correspond to exactly the same viewing direction, aligns them rotationally and translationally, and averages them. This results in an actual projection of the molecule at improved SNR. In practice, due to the high levels of noise, many times the class averaging algorithm incorrectly identifies completely different images as corresponding to the same viewing direction. The resulting class average thus consists of projections that correspond to completely different viewing directions, and therefore it does not correspond to any actual projection of the molecule.

As a result, a set of class averages generated from experimental data is likely to contain some class averages which are inconsistent with the others, in the sense that they have no common lines with the other class averages. A weakness of the synchronization algorithm, as presented in this paper, is that it requires one to estimate the block S_{ij} for each pair of images i and j (see (4.15)). Thus, even if some of the class averages do not represent any actual projection of the molecule and are therefore inconsistent as explained above, the algorithm still tries to find the rotation matrices corresponding to these images, although no such rotations exist. This necessarily affects all other estimated rotations, introducing large errors in the estimation.

Hence, before processing experimental class averages with the synchronization algorithm, we need to discard from the data set images that are unlikely to correspond to actual projections of the molecule. Although algorithms for that task are important in themselves, here we just want to demonstrate that the synchronization algorithm is relevant in the case of experimental class averages, and so we apply the following heuristic. We take the 1500 class averages and use the synchronization algorithm to estimate their rotations, denoted $R_1, \ldots, R_{1500}$. Then, for each pair of common lines c_{ij} and c_{ji}, we compute the angle between $R_i c_{ij}$ and $R_j c_{ji}$. Ideally, this angle should be zero. For each projection, we count the number of times this angle was above 20 degrees (this threshold was chosen arbitrarily, and its precise value was found to hardly affect the outcome). This counts the number of times the projection was inconsistent with the rest of the projections. Then we find the 100 projections with the

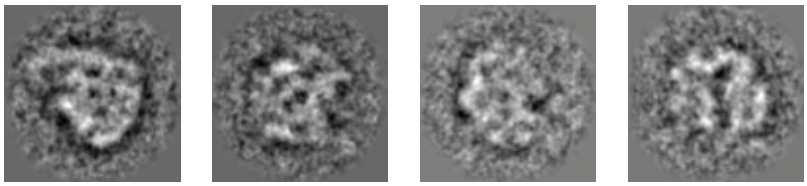
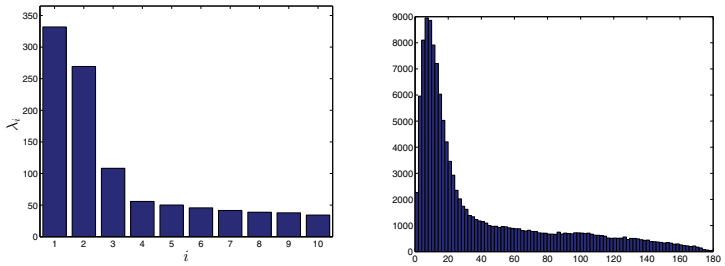


Figure 9. *Sample of* 4 *out of* 500 *experimental class averages.*



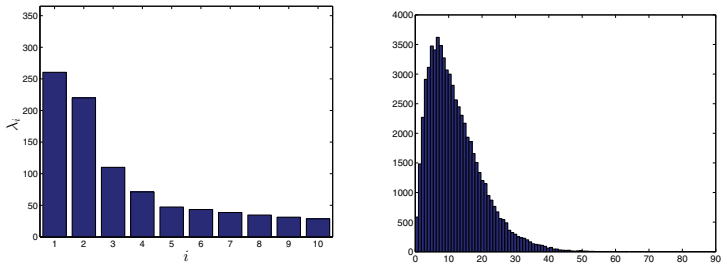
(a) Spectrum of S (b) Common line consistency

Figure 10. *Applying the algorithm of section* 4 *on experimental data.* (a) *Ten largest eigenvalues of the matrix* S. (b) *Histogram of common line embedding errors (in degrees).*

largest number of inconsistencies and discard them from our data set. We repeat this process until we are left with only 500 class averages, which we use as an input to the synchronization algorithm. Four projections from this data set are shown in Figure 9.

We take the set of 500 class averages and apply to them the synchronization algorithm. Figure 10(a) shows the 10 largest eigenvalues of the matrix S. To see if the estimated rotations respect the common lines we have detected, we compute for each pair of common lines c_{ij} and c_{ji} the angle between $R_i c_{ij}$ and $R_j c_{ji}$ (as was done during the projections' filtration process explained above). This gives the embedding error for the pair c_{ij} and c_{ji}. Figure 10(b) shows the histogram of the embedding errors (in degrees). We see that many of the errors are small, but there is a long tail of large errors. We postulate that common lines with large embedding errors correspond to misidentified common lines. We thus discard common lines whose embedding errors are larger than 20 degrees and run the synchronization algorithm again. Figures 11(a) and 11(b) show the spectrum of the synchronization matrix and the histogram of embedding errors, respectively, at the end of this run. As we see in Figure 11(b), the embedding errors have been significantly reduced.

Next, we use the common lines to estimate the two-dimensional shift in each projection as described in [18]. Finally, we take the 500 estimated rotations together with the 500 appropriately reshifted class averages and reconstruct a three-dimensional density map. A two-dimensional rendering of the reconstructed volume is shown in Figure 12. Based on our tests with simulated data and following the results in Figure 8, we speculate that the errors in the reconstruction of the density map are attributed mainly to the effects of the CTF as well as to errors in recentering the projections.



(a) Spectrum of S (b) Common line consistency

Figure 11. *Results of the synchronization algorithm after removing common lines whose embedding errors are larger than* 20 *degrees.* (a) *Ten largest eigenvalues of the matrix* S. (b) *Histogram of common line embedding errors (in degrees).*

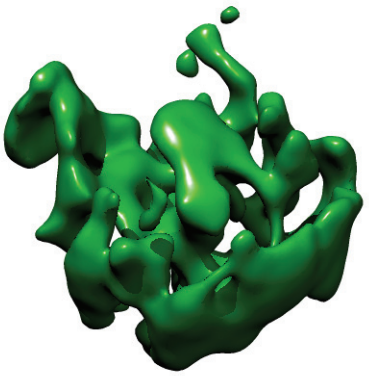


Figure 12. *Density map reconstructed using the* 500 *experimental class averages.*

7. Discussion and future extensions. The algorithm in this paper is based on common lines between projections. Detecting common lines between cryo-EM projections is a nontrivial task, as their SNR is typically very low (less than 1/100). Nevertheless, the encouraging results described in section 6 show that the algorithm is robust to noise. This robustness is attributed to three reasons. First, each block S_{ij} in (4.15) is an average of many blocks $B_{ij}^{(k)}$. This averaging process may help reducing the error in the estimate of each block S_{ij}, as long as the errors in the individual blocks $B_{ij}^{(k)}$ have zero mean. Second, the algorithm is based on computing the three eigenvectors of a symmetric matrix, which correspond to its three largest eigenvalues. This is a global computation that takes into account all local pieces of information about common lines simultaneously. Thus, even if some of the blocks S_{ij} are wrong, those errors are averaged out in the global eigenvector computation. Third, the (clean) matrix S is of rank 3 and has a large spectral gap. A large spectral gap means that high levels of noise are required in order to cause crossing of the eigenvalues, which results in breakdown of the algorithm. The spectral gap of 2/3 of the presented algorithm is larger than was previously obtained in [4, 17].

The presented algorithm also has other properties which make it attractive in practice. First, the algorithm does not depend on the number of projections. Second, although the algorithm scales cubically with the number of images, its running time is acceptable even for several thousands of images, and it can be easily implemented on parallel architectures.

As presented in section 4, the matrix S is a dense matrix and requires all the blocks S_{ij} to be estimated. However, the estimates of the various blocks are not equally reliable, and some are even completely wrong (for example, blocks S_{ij} for which the common line between images i and j was misidentified). Thus, it would be better to try to estimate how reliable a given block S_{ij} is and to construct the matrix S using only the reliable blocks, leaving "holes" in the matrix for the unreliable ones. We can then try to fill in the missing blocks in S by using the properties of the matrix S.

From the computational point of view, storing a $2N \times 2N$ dense matrix is prohibitive if N is very large. However, there is no need to actually store the $2N \times 2N$ matrix S, as we need only its first three eigenvectors ($2N \times 3$ numbers). Computing these eigenvectors does not require us to store the matrix S, but just to apply it several times to three vectors (as in the power method [6]). The large gap in the spectrum of S ensures that it should be applied only a small number of times. Moreover, any such application can be implemented in parallel, taking full advantage of existing parallel architectures.

The algorithm as presented assumes that all input images are projections of the same underlying object. In practice, when class averages are estimated from experimental data, some class averages are likely to be inconsistent in the sense that they have no common lines with the other class averages, as explained in section 6.2. To process class averages produced from experimental data, the algorithm needs to be modified to detect such inconsistent averages and discard them. This is a future research direction for increasing the robustness of the algorithm in order to adapt it to experimental data.

Finally, in its current form, the algorithm assumes that the underlying molecule has no symmetries. This assumption is already implied in our statement of the Fourier projection-slice theorem, which says that any two projections share a single common line. Our algorithm is then based on detecting this single common line between pairs of projections. When the molecule has symmetries, any two projections share more than one common line, and this redundancy should be taken into account when estimating the structure of the molecule. Thus, a possible extension of the algorithm is to adapt it to molecules with symmetries.

Acknowledgments. The authors thank Lanhui Wang (PACM, Princeton University) and Ronny Hadani (Department of Mathematics, University of Texas at Austin) for many valuable discussions. They also thank Fred Sigworth (Department of Cellular and Molecular Physiology, Yale School of Medicine), Alp Kucukelbir (Department of Biomedical Engineering, Yale School of Engineering and Applied Sciences), and Carlos Óscar Sánchez Sorzano (Biocomputing Unit, National Center of Biotechnology CSIC, Cantoblanco, Madrid, Spain) for their advice and fruitful discussions about the processing of experimental data sets.

REFERENCES

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

[2] A. Averbuch, R.R. Coifman, D.L. Donoho, M. Israeli, and Y. Shkolnisky, *A framework for discrete integral transformations* I—*The pseudopolar Fourier transform*, SIAM J. Sci. Comput., 30 (2008), pp. 764–784.
[3] A. Averbuch and Y. Shkolnisky, 3*D Fourier based discrete Radon transform*, Appl. Comput. Harmon. Anal., 15 (2003), pp. 33–69.
[4] R.R. Coifman, Y. Shkolnisky, F.J. Sigworth, and A. Singer, *Reference-free structure determination through eigenvectors of center of mass operators*, Appl. Comput. Harmon. Anal., 28 (2010), pp. 296–312.
[5] J. Frank, *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State*, Oxford University Press, New York, 2006.
[6] G.H. Golub and C.F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1996.
[7] A.B. Goncharov, *Integral geometry and three-dimensional reconstruction of randomly oriented identical particles from their electron microphotos*, Acta Appl. Math., 11 (1988), pp. 199–211.
[8] R. Hadani and A. Singer, *Representation theoretic patterns in three dimensional cryo-electron microscopy* I: *The intrinsic reconstitution algorithm*, Ann. Math. (2), 174 (2011), pp. 1219–1241.
[9] F. Natterer, *The Mathematics of Computerized Tomography*, Classics Appl. Math. 32, SIAM, Philadelphia, 2001.
[10] P.A. Penczek, R.A. Grassucci, and J. Frank, *The ribosome at improved resolution: New techniques for merging and orientation refinement in* 3*D cryo-electron microscopy of biological particles*, Ultramicroscopy, 53 (1994), pp. 251–270.
[11] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, and T.E. Ferrin, *UCSF Chimera—A visualization system for exploratory research and analysis*, J. Comput. Chem., 25 (2004), pp. 1605–1612.
[12] M. Radermacher, T. Wagenknecht, A. Verschoor, and J. Frank, *Three-dimensional reconstruction from a single-exposure, random conical tilt series applied to the* 50*S ribosomal subunit of Escherichia coli*, J. Microsc., 146 (1987), pp. 113–136.
[13] M. Radermacher, T. Wagenknecht, A. Verschoor, and J. Frank, *Three-dimensional structure of the large ribosomal subunit from Escherichia coli*, EMBO J., 6 (1987), pp. 1107–1114.
[14] D.B. Salzman, *A method of general moments for orienting* 2*D projections of unknown* 3*D objects*, Comput. Vision Graph. Image Process., 50 (1990), pp. 129–156.
[15] A. Singer, *Angular synchronization by eigenvectors and semidefinite programming*, Appl. Comput. Harmon. Anal., 30 (2011), pp. 20–36.
[16] A. Singer, R.R. Coifman, F.J. Sigworth, D.W. Chester, and Y. Shkolnisky, *Detecting consistent common lines in cryo-EM by voting*, J. Struct. Biol., 169 (2010), pp. 312–322.
[17] A. Singer and Y. Shkolnisky, *Three-dimensional structure determination from common lines in cryo-EM by eigenvectors and semidefinite programming*, SIAM J. Imaging Sci., 4 (2011), pp. 543–572.
[18] A. Singer and Y. Shkolnisky, *Center of mass operators for cryo-EM—Theory and implementation*, in Modeling Nanoscale Imaging in Electron Microscopy, Nanostructure Science and Technology, Springer, New York, 2012, pp. 147–177.
[19] H. Stark, M.V. Rodnina, H.J. Wieden, F. Zemlin, W. Wintermeyer, and M. van Heel, *Ribosome interactions of aminoacyl-tRNA and elongation factor Tu in the codon-recognition complex*, Nat. Struct. Biol., 9 (2002), pp. 849–854.
[20] M. van Heel, *Angular reconstitution: A posteriori assignment of projection directions for* 3*D reconstruction*, Ultramicroscopy, 21 (1987), pp. 111–123.
[21] M. van Heel, B. Gowen, R. Matadeen, E.V. Orlova, R. Finn, T. Pape, D. Cohen, H. Stark, R. Schmidt, M. Schatz, and A. Patwardhan, *Single-particle electron cryo-microscopy: Towards atomic resolution*, Q. Rev. Biophys., 33 (2000), pp. 307–369.
[22] M. van Heel, G. Harauz, E.V. Orlova, R. Schmidt, and M. Schatz, *A new generation of the IMAGIC image processing system*, J. Struct. Biol., 116 (1996), pp. 17–24.
[23] L. Wang and F.J. Sigworth, *Cryo-EM and single particles*, Physiology (Bethesda), 21 (2006), pp. 13–18.