

## Research Article

## Open Access

Guoliang Xu, Xia Wang, Ming Li, and Zhucui Jing

# Fast and Robust Orientation of Cryo-Electron Microscopy Images

DOI 10.1515/mlbmb-2015-0010

Received August 14, 2015; accepted October 30, 2015

**Abstract:** We present an efficient and reliable algorithm for determining the orientations of noisy images obtained from projections of a three-dimensional object. Based on the linear relationship among the common line vectors in one image plane, we construct a sparse matrix, and show that the coordinates of the common line vectors are the eigenvectors of the matrix with respect to the eigenvalue 1. The projection directions and in-plane rotation angles can be determined from these coordinates. A robust computation method of common lines in the real space using a weighted cross-correlation function is proposed to increase the robustness of the algorithm against the noise. A small number of good leading images, which have the maximal dissimilarity, are used to increase the reliability of orientations and improve the efficiency for determining the orientations of all the images. Numerical experiments show that the proposed algorithm is effective and efficient.

**Keywords:** Cryo-EM Images, Common line, Orientation, Single-Particle Reconstruction

## 1 Introduction

In the recent decades, cryo-electron microscopy (cryo-EM) single-particle reconstruction (SPR) has become an indispensable tool in three-dimensional (3D) electron microscopy [6, 20] in order to obtain the 3D structures of macromolecular complexes. Starting from a set of two-dimensional (2D) images taken by an electron microscope from identical macromolecular particles, SPR utilizes several image-processing techniques, mainly including image classification, orientation and 3D reconstruction. The reconstruction result is further refined using projection matching methods [15, 17, 26] to improve the resolutions.


Our objective is to solve the problems of image classification, orientation and 3D reconstruction separately and reliably. We have already developed some methods, including classification [27] and 3D reconstruction [2, 9, 11, 28]. The goal of this paper is to target the orientation problem.

**Orientation problem.** In the cryo-EM SPR, every 2D image  $I$  approximately represents a 2D projection of the 3D object, which can be described mathematically using

$$Xf_{(\alpha,\beta,\gamma)}(x,y) := \int_{\mathbb{R}} f(x\tilde{\mathbf{e}}_1 + y\tilde{\mathbf{e}}_2 + t\mathbf{d})dt,$$

where  $f$  is the 3D object, and  $(\alpha, \beta, \gamma)$  are the three Euler angles which define the orientation of the projection. The vector  $\mathbf{d} \in S^2$  is the projection direction and  $S^2$  the unit sphere in  $\mathbb{R}^3$ . The two vectors  $\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2$  determine the in-plane rotation.  $\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2$  and  $\mathbf{d}$  are defined in the subsection 2.2. The orientation problem of the image  $I$  is to find the three Euler angles or equivalently the projection direction and the in-plane rotation. The orientation problem is critical, since we need the orientations of the images to recover the 3D structure of the object.

**Guoliang Xu, Xia Wang, Ming Li, Zhucui Jing:** LSEC, Institute of Computational Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100190, China

 © 2015 Guoliang Xu et al., licensee De Gruyter Open.

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License.

However, since the cryo-EM SPR images have a very low signal-to-noise ratio (SNR), exactly recovering the orientations is impossible. Hence, we seek to recover the orientations of these images approximately.

**Previous Work.** Several methods for solving the orientation problem have been proposed. One is the random-conical data collection method [18] in which the data were collected by tilting and untilting the specimen grid. The orientations of the tilting series were determined easily from the tilt geometry and untilted series. Another is the common line-based method (see section 2.1 for the definition of common line) which determines the orientations mathematically by using of the central slice theorem which states that the 2D Fourier transform of the projections in the  $\mathbf{d}$ -direction of a 3D function  $f$  is the same as the central slice of the 3D Fourier transform of  $f$  at the plane  $\mathbf{d}^\perp$ . The first common line based orientation method was proposed in [3, 4] for highly symmetrical particles by searching in a complete asymmetric unit of the orientation space. An improved common line technique named angular reconstruction was proposed in [10] for asymmetric object in which a minimum of three projections was required to calculate the orientations of the projections of the asymmetric object. In [5], the self-common lines and cross-common lines were used to calculate the orientations and determine the hand of the icosahedral particles.

Recently, a series of common line based orientation methods were proposed in [22–25]. In [22] the orientations of all the discrete radial Fourier lines of projections were calculated from first three eigenvectors of a sparse adjacency matrix. [24] proposed two algorithms which both used the computed common lines to determine the orientations, one computing the three largest eigenvectors of a symmetric matrix and another solving a semidefinite programming. The method was improved in [25] using least unsquared deviations via semidefinite relaxation. Different from algorithms in [22, 24, 25] which were based on common lines between pairs of projections, the method in [23] used triplets of projections to construct a synchronization matrix using triplets of projections which improved the noise robustness. Orientation algorithms have been implemented in the softwares. In AUTO3DEM, a polar Fourier transform method is adopted (see [1]) that facilitates determination and refinement of orientations of individual biological macromolecules imaged with cryoelectron microscopy techniques. In the software EMAN (see [12]) and FREALIGN (see [19]), a project matching method is used for orientation. In RELION (see [21]), a Bayesian approach to cryo-EM structure determination is implemented.

**Our Contributions.** In this paper, we present a method that significantly improves the efficiency of the method presented in [22]. Different from the method in [22] which uses all the radial Fourier lines of the projections, we only use the common lines to determine the orientations. Our algorithm also differs from [10, 23–25] in that instead of determining the Euler angles or the rotation matrices, our orientation method calculates the coordinates of the common line vectors, then determines the projection directions and the in-plane rotation angles from these coordinates. In addition, in order to increase the noise robustness, we use the weighted cross-correlation to compute the common lines in the real space. The computational complexity of the algorithm is linear with respect to the number of images.

The organization of this paper is as follows. In section 2, we set up the related mathematical background. Section 3 is devoted to the robust computation of common lines and the recovery algorithm of the projection orientations, together with a few technical implementation details. Examples of the application of this algorithm to a phantom are given in section 4. We conclude this paper in section 5.

## 2 Preliminaries

In this section, we introduce the background materials related to the proposed method, including the definitions of the common line, Radon transform, Euler angles and image coordinate systems.

## 2.1 Radon Transform and Common Line

The 2D projection transform is equivalent to the following 2D Radon transform.

**Radon Transform** (see [13]). Let  $f(\mathbf{x}) \in L^2(\mathbb{R}^2)$ . Then for any  $\theta \in S^1$ , the 2D Radon transform is defined by

$$(R_2 f)(\theta, s) := \int_{\theta^\perp} f(s\theta + \mathbf{y}) d\mathbf{y} = \int_{-\infty}^{\infty} f(s\theta + t\mathbf{d}) dt,$$

where  $\theta^\perp = \{\mathbf{x} \in \mathbb{R}^2 : \mathbf{x}^T \theta = 0\}$  is the subspace orthogonal to  $\theta$  and  $\mathbf{d}$  is a unit vector in  $\theta^\perp$ .

**Theorem 2.1.** (see [14]) Let  $f \in L^2(\mathbb{R}^2)$ . Then for  $\theta \in S^1, s \in \mathbb{R}^1$ ,

$$(R_2 f)^\wedge(\theta, s) = \sqrt{2\pi} \hat{f}(s\theta),$$

where the 1D Fourier transform on the left-hand side is applied to the second variable of  $R_2 f$  with  $\theta$  fixed.

The theorem above is named as central slice theorem for Radon transform. From Theorem 2.1, we have the following corollary.

**Corollary 2.1.** Let  $f \in L^2(\mathbb{R}^3)$ . Then for  $(\alpha_1, \beta_1, \gamma_1) \neq (\alpha_2, \beta_2, \gamma_2)$ , there exist the directions  $\theta_{12}, \theta_{21} \in S^1$  such that

$$R_2(X_{(\alpha_1, \beta_1, \gamma_1)} f)(\theta_{12}, s) = R_2(X_{(\alpha_2, \beta_2, \gamma_2)} f)(\theta_{21}, s).$$

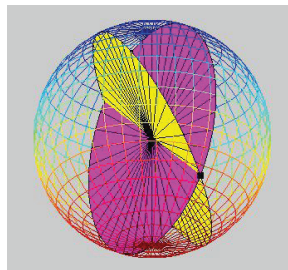
For the 2D Radon transform, the following result is useful in the computation of common lines.

**Theorem 2.2.** [13] Let  $f, g \in L^2(\mathbb{R}^2)$ , then

$$R_2 f \star R_2 g = R_2(f \star g).$$

The common line in the Fourier space is defined as follows.

**Common line.** Given a 3D function  $f$ , let  $I_1$  and  $I_2$  be two projections of  $f$  in the directions  $\mathbf{d}_1$  and  $\mathbf{d}_2$ , respectively, with  $\mathbf{d}_1 \neq \mathbf{d}_2$ . Let  $\hat{I}_1$  and  $\hat{I}_2$  be the Fourier transforms of  $I_1$  and  $I_2$ , respectively. Then, from the central slice theorem, we know that  $\hat{I}_1$  and  $\hat{I}_2$  are the two central slices of the Fourier transform of  $f$  at the planes perpendicular to the directions  $\mathbf{d}_1$  and  $\mathbf{d}_2$ , respectively. The intersection line of the two planes is called the *common line* of  $\hat{I}_1$  and  $\hat{I}_2$ , denoted by  $\mathbf{l}(\lambda) = \lambda \mathbf{p}_{12} = \lambda \mathbf{p}_{21}$ , where  $\mathbf{p}_{12} = \mathbf{p}_{21} \in S^2$  are two unit vectors named as *common line vectors* which define the direction of the common line lying in  $I_1$  and  $I_2$  respectively. Fig. 2.1 is used to illustrate the definition of common line vectors in the Fourier space.



**Figure 2.1:** Definition of the common line vector in the Fourier space.

For  $M$  ( $M > 2$ ) images  $\{I_i\}_{i=1}^M$  with distinct projection directions  $\{\mathbf{d}_i\}_{i=1}^M$  and in-plane rotation angles  $\{\gamma_i\}_{i=1}^M$ , there exists a common line between any two Fourier transforms of the images  $I_i$  and  $I_j$  with  $i \neq j$ . Denote all the common lines vectors in the image  $I_i$  by the set  $\mathbf{P}_i = \{\mathbf{p}_{i1}, \mathbf{p}_{i2}, \dots, \mathbf{p}_{i,M-1}\}$ . Our goal is to

determine the coordinates of the points  $\mathbf{P}_i$ , for  $i = 1, \dots, M$ . Then, we can derive the projection directions  $\{\mathbf{d}_i\}_{i=1}^M$  and in-plane rotation angles  $\{\gamma_i\}_{i=1}^M$ .

In the real space, the common lines mean that the 2D Radon transforms of the two images coincide along certain lines  $\theta_{12}$  and  $\theta_{21}$ . In this paper, we compute the common lines in the Radon transform.

## 2.2 Image Coordinate System and In-plane Rotation

The orientation of a 3D object can be described by three Euler angles  $\alpha$ ,  $\beta$  and  $\gamma$ . In this paper, we use the following convention to define these angles. Denote by  $(x, y, z)$  the original right-hand coordinate system. Then  $\alpha$ ,  $\beta$  and  $\gamma$  are the angles of three successive axial rotations. (1) A rotation about the  $z$ -axis with angle  $\alpha \in (-\pi, \pi]$ . This rotation results in new axes  $x'$ ,  $y'$  and  $z'$  with  $z' = z$ . (2) A rotation about the  $y'$ -axis with angle  $\beta \in [0, \pi]$ . This rotation results in new axes  $x''$ ,  $y''$  and  $z''$  with  $y'' = y'$ . (3) A rotation about the  $z''$ -axis with angle  $\gamma \in (-\pi, \pi]$ . This rotation results in new axes  $X$ ,  $Y$  and  $Z$  with  $Z = z''$ . These successive rotations have the following matrix representation:

$$R_1 R_2 R_3 = [X, Y, Z]^T = [\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \mathbf{d}]^T,$$

with

$$R_1 = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_2 = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix}, R_3 = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and

$$\tilde{\mathbf{e}}_1 = [\cos \gamma \cos \alpha \cos \beta - \sin \gamma \sin \alpha, \cos \gamma \sin \alpha \cos \beta + \sin \gamma \cos \alpha, -\cos \gamma \sin \beta]^T \quad (2.1)$$

$$\tilde{\mathbf{e}}_2 = [-\sin \gamma \cos \alpha \cos \beta - \cos \gamma \sin \alpha, -\sin \gamma \sin \alpha \cos \beta + \cos \gamma \cos \alpha, \sin \gamma \sin \beta]^T \quad (2.2)$$

$$\mathbf{d} = [\cos \alpha \sin \beta, \sin \alpha \sin \beta, \cos \beta]^T. \quad (2.3)$$

Note that  $\tilde{\mathbf{e}}_1 \tilde{\mathbf{e}}_2 \mathbf{d}$ -coordinate system is also of right-handed.

Assume that we have determined the projection direction  $\mathbf{d}$  for image  $I$ , we need to find the Euler angles from  $\mathbf{d}$ . Let  $\mathbf{d} = [d_x, d_y, d_z]^T \in S^2$ , if  $\mathbf{d} \neq [0, 0, \pm 1]^T$ , the angles  $\alpha$  and  $\beta$  are uniquely defined by the equation (2.3). In fact, we do not need explicitly to calculate  $\alpha$  and  $\beta$ . What we need are:

$$\cos \beta = d_z, \quad \sin \beta = \sqrt{1 - d_z^2}, \quad (2.4)$$

$$\cos \alpha = d_x / \sqrt{1 - d_z^2}, \quad \sin \alpha = d_y / \sqrt{1 - d_z^2}. \quad (2.5)$$

If  $\mathbf{d} = [0, 0, \pm 1]^T$ ,  $\beta$  is uniquely defined. We can set  $\alpha = 0$ . Therefore,

$$\cos \beta = d_z, \quad \sin \beta = 0, \quad \cos \alpha = 1, \quad \sin \alpha = 0. \quad (2.6)$$

The measured 2D image  $I$  in the projection direction  $\mathbf{d}$  can be regarded as a function defined on the plane  $\mathbf{d}^\perp \subset \mathbb{R}^3$ . The two orthonormal vectors  $\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2$  are the coordinate system of  $\mathbf{d}^\perp$  for the measured image  $I$ . The common lines are computed using the coordinate system. However, From the projection direction  $\mathbf{d}$ , we can get

$$\mathbf{e}_1 = [\cos \alpha \cos \beta, \sin \alpha \cos \beta, -\sin \beta]^T, \quad (2.7)$$

$$\mathbf{e}_2 = [-\sin \alpha, \cos \alpha, 0]^T, \quad (2.8)$$

for  $\gamma = 0$  from (2.1) and (2.2). The difference of the  $\mathbf{e}_1 \mathbf{e}_2$  coordinate system from the  $\tilde{\mathbf{e}}_1 \tilde{\mathbf{e}}_2$  coordinate system is an in-plane rotation with rotation angle  $\gamma$ . The Algorithm 3.4 utilizes the difference to solve the in-plane rotation angle.

### 2.3 B-spline Radial Basis Functions and their Radon Transform

In order to compute the common lines in the real space based on 2D Radon transform efficiently, we represent the 2D image  $I$  with cubic B-spline radial basis functions  $\phi_i$ .

Given an even integer  $m \geq 4$ , suppose the domain  $\Omega = [-\frac{n}{2} - 1, \frac{n}{2} + 1]^2$  is uniformly partitioned with grid points

$$\mathbf{x}_i = \mathbf{i}h, \quad -\frac{m}{2} \leq \mathbf{i} \leq \frac{m}{2}, \quad \mathbf{i} = (i_1, i_2),$$

where  $h = \frac{n+2}{m}$ . The image  $I$  is represented as

$$I(\mathbf{x}) \approx \sum_{-\frac{m}{2}+2 \leq \mathbf{i} \leq \frac{m}{2}-2} c_i \phi_i(\mathbf{x}, h), \quad \mathbf{x} = [x_1, x_2]^T \in \Omega, \quad (2.9)$$

where  $\phi_i(\mathbf{x}, h) = N(\|\mathbf{x} - \mathbf{i}h\|/h)$  with  $\|\cdot\|$  denoting the  $L^2$ -norm. The cubic B-spline basis function  $N(s)$  (see [16]) is defined on the uniform knots  $-2, -1, 0, 1, 2$ , i.e.,

$$N(s) = \begin{cases} \frac{2}{3} - s^2 + \frac{|s|^3}{2}, & 0 \leq |s| < 1, \\ \frac{1}{6} (2 - |s|)^3, & 1 \leq |s| < 2, \\ 0, & 2 \leq |s|. \end{cases} \quad (2.10)$$

The support of  $N(s/h)$  is the interval  $(-2h, 2h)$ .

Let  $\mathbf{d} = \theta^\perp \in S^1$  be a given direction. Then the 2D Radon transform of the cubic B-spline radial function  $\phi_i$  in the direction  $\mathbf{d}$  defined as

$$\begin{aligned} (R_2 \phi_i)(\theta, s) &= \int_{-\infty}^{\infty} \phi_i(s\theta + t\mathbf{d}, h) dt \\ &= \int_{-\infty}^{\infty} N\left(\left\|\left(s/h - \mathbf{i}^T \theta\right) \theta + t\mathbf{d}/h\right\|\right) dt \\ &= h \int_{-\infty}^{\infty} N\left(\left\|\mathbf{a}_i(s) + t\mathbf{d}\right\|\right) dt \\ &= h \int_{-\infty}^{\infty} N\left(\sqrt{a^2 + t^2}\right) dt, \end{aligned}$$

where  $a = \left\|\left(s/h - \mathbf{i}^T \theta\right) \theta\right\|$ . From (2.10), we have

$$(R_2 \phi_i)(\theta, s) = \begin{cases} 2h \int_0^{\sqrt{1-a^2}} N(\phi(t)) dt \\ + 2h \int_{\sqrt{4-a^2}}^{\sqrt{4-a^2}} N(\phi(t)) dt, & 0 \leq a \leq 1, \\ 2h \int_0^{\sqrt{1-a^2}} N(\phi(t)) dt, & 1 < a < 2, \\ 0, & 2 \leq a < \infty, \end{cases} \quad (2.11)$$

where  $\phi(t) = (t^2 + a^2)^{\frac{1}{2}}$ . Using the expression (2.10) and the integral formulas for  $\int (t^2 + a^2)^{\frac{1}{2}} dt$  and  $\int (t^2 + a^2)^{\frac{3}{2}} dt$ , the 2D Radon transform of the function  $\phi_i$  can be exactly computed.

The Radon transform  $R_2 I$  of the image  $I$  is given by

$$(R_2 I)(\theta, s) \approx R_2 \left( \sum_{\mathbf{i}} c_{\mathbf{i}} \phi_{\mathbf{i}} \right) (\theta, s) = \sum_{\mathbf{i}} c_{\mathbf{i}} (R_2 \phi_{\mathbf{i}})(\theta, s),$$

where  $-\frac{m}{2} + 2 \leq \mathbf{i} \leq \frac{m}{2} - 2$ , and  $(R_2 \phi_{\mathbf{i}})(\theta, s)$  is computed using (2.11). Note that  $N(s)$  is locally supported. The cost of computing  $R_2 I$  is  $O(m^2)$ . The total cost of the projection is  $O(Mm^2)$ , where  $M$  denotes the total number of projections. Compared with using FFT, the cost of this approach is one order higher. However, its performance is quite satisfactory. The computation could be accelerated by removing small coefficients.

$$(R_2 I)(\theta, s) \approx \sum_{\{c_{\mathbf{i}} > \epsilon\}} c_{\mathbf{i}} (R_2 \phi_{\mathbf{i}})(\theta, s), \quad (2.12)$$

where  $\epsilon > 0$  is a given small number.

### 3 Common Line-Based Orientation Algorithm

In this section, we propose the common line-based orientation algorithm. Given a set of cryo-EM images  $\{I_i\}_{i=1}^M$ , and assuming that these centers of the images have been aligned before orientation. Our goal is to determine projection directions  $\{\mathbf{d}_i\}_{i=1}^M$  and in-plane rotation angle  $\{\gamma_i\}_{i=1}^M$  of these images.

We first determine the coordinates of the common line vectors of a subset of the projection images, named as leading images. Then the information is used to determine the coordinates of the common line vectors of the remaining images. After that, we calculate the projection directions and the in-plane rotation angles of these images so as to do 3D reconstruction in the real space. The leading images are critical for obtaining reliable common lines and orientations. In the following subsection, we present an algorithm to compute the leading images which have maximal dissimilarity.

#### 3.1 Compute the leading images

Let  $\{I_i\}_{i=1}^M$  be a set of cryo-EM images. The algorithm in the following computes the leading images.

**Algorithm 3.1.** *Compute the leading images*

1. Classify all the input images into  $K$  ( $\leq M$ ) classes  $\{\mathbf{C}_i\}_{i=1}^K$  using the FTTR (Fourier-transform-based translation and rotation-invariant) method (see [27] for the classification method), where  $K$  is a user-specified number. Denote by  $\tilde{\mathbf{C}}_i$  the set consisting of the FTTR invariants of all the elements in  $\mathbf{C}_i$ .
2. Compute the center  $\tilde{c}_i$  of class  $\tilde{\mathbf{C}}_i$  by averaging,  $i = 1, \dots, K$ .
3. Choose a representative element  $I_i$  in the class  $\mathbf{C}_i$  such that the FTTR invariant of  $I_i$  is the nearest element of  $\tilde{\mathbf{C}}_i$  to the center  $\tilde{c}_i$ .
4. Denoise representative  $I_i$ ,  $i = 1, \dots, K$ , using the following average-based denoising algorithm.
  - (a) In  $\mathbf{C}_i$ , find the nearest  $D$  (we take  $D = 10$ ) projections to  $I_i$  with respect to the distances defined by their FTTR invariants.
  - (b) Align the nearest  $D$  projections to  $I_i$ .
  - (c) Average the aligned images computed in the previous step. The averaged image, still denoted by  $I_i$ , is defined as the average denoising projection.
5. Compute  $J$  leading images, from  $K$  denoised representatives, as follows:
  - (a) Compute the distance  $d_{ij}$  between  $I_i$  and  $I_j$  as

$$d_{ij} = \min_{\substack{l=0, \dots, L; \\ m=0, \dots, 2L-1}} \|R_2 I_i(\theta_l, s) - R_2 I_j(\theta_m, s)\|, \quad (3.13)$$

for  $i, j = 1, \dots, K$ , where  $\theta_l = l\pi/L$ ,  $\theta_m = m\pi/L$ . Note that  $d_{ij} = d_{ji}$ .

(b) Compute the score  $s_i$  for  $I_i$  as

$$s_i = \sum_{j=1, \dots, K} d_{ij}, \quad i = 1, \dots, K. \quad (3.14)$$

(c) Sort  $s_i$  and then find the  $J$  smallest scores. The  $J$  denoised representatives with smallest scores are taken as the leading images.

**Remark 3.1.** Since the leading images are determined using a clustering algorithm, these images have maximal dissimilarities and the obtained directions of the leading images are not close each other. This makes the computation of common line robust. The number  $J$  is small, and we require that  $J$  is larger than 4. Using larger  $J$  leads to larger amount of calculations. In our experiments,  $J$  is taken as 6.

### 3.2 Robust Common Line Computation in the Real Space

Owing to the extremely high noise presented in the cryo-EM images, direct computation of the common lines between two images will result in large errors. Hence, denoising should be used before common line computation. We propose the following robust common line computation algorithm.

Let  $I_i$  and  $I_j$  ( $i \neq j$ ) be two 2D projection images, with size  $h \times h$ , and  $R_2$  be the 2D Radon transform. According to Corollary 2.1, there exists directions  $\theta_{ij}$  and  $\theta_{ji}$  such that

$$R_2 I_i(\theta_{ij}, s) = R_2 I_j(\theta_{ji}, s). \quad (3.15)$$

That is, in the real space, the common lines are two one-dimensional (1D) projections of the two images  $I_i$  and  $I_j$  in the direction  $\mathbf{d}_i = \theta_{ij}^\perp$  and  $\mathbf{d}_j = \theta_{ji}^\perp$ , respectively. Since the projection images are highly noisy, the common lines are computed after Gaussian low-pass filtering of the two images. Let  $G$  be the Gaussian filter. According to Theorem 2.2, we have  $R_2(I * G) = R_2 I * R_2 G$ .

**Weighted cross-correlation function.** Let  $R_{ik} \in \mathbb{R}^{L+1}$  be  $L+1$  samples of 1D projections of  $I_i$  in the direction  $\mathbf{d}_{ik} = \theta_{ik}^\perp$ , ( $i = 1, \dots, M$ ). Define a set of 2D Gaussian filters  $G_0, G_1, \dots, G_m$  with

$$G_r(x, y) = \frac{1}{2\pi\sigma_r^2} e^{-\frac{x^2+y^2}{2\sigma_r^2}}, \quad \sigma_r = \sqrt{\frac{h_r^2 - 1}{3}}, \quad r = 0, 1, \dots, m,$$

where  $h_r = E[\lambda^r h]$ , and  $0 < \lambda < 1$ .  $E[\cdot]$  denotes taking the integer part. Compute

$$\begin{aligned} H_r(s) &= R_2 G_r = \frac{1}{2\pi\sigma_r^2} e^{-\frac{s^2}{2\sigma_r^2}} \int_{-\infty}^{\infty} e^{-\frac{t^2}{2\sigma_r^2}} dt \\ &= \frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{s^2}{2\sigma_r^2}}, \quad r = 0, 1, \dots, m, \end{aligned}$$

and their discrete matrix form  $D_r$ . We take  $\lambda = \frac{1}{2}$ ,  $m = 4$ . Then we define the following *weighted cross-correlation function* for the two image  $I_i$  and  $I_j$  with projection direction  $\mathbf{d}_{ik} = \theta_{ik}^\perp$  and  $\mathbf{d}_{jl} = \theta_{jl}^\perp$ ,

$$\begin{aligned} c_{kl} &= \frac{1}{\|R_{ik}\| \|R_{jl}\|} \sum_{r=0}^m w_r (D_r R_{ik})^T D_r R_{jl} \\ &= \frac{1}{\|R_{ik}\| \|R_{jl}\|} (R_{ik})^T \left( \sum_{r=0}^m w_r D_r^T D_r \right) R_{jl}, \end{aligned} \quad (3.16)$$

where  $w_r$  are positive weights with the constraint  $\sum_{r=0}^m w_r = 1$ . We take  $m = 4$  and  $w_0 = \frac{250}{594}$ ,  $w_1 = \frac{202}{594}$ ,  $w_2 = \frac{106}{594}$ ,  $w_3 = \frac{32}{594}$ ,  $w_4 = \frac{4}{594}$ .

The robust computation of the common line between  $I_i$  and  $I_j$ , using (3.15) directly, is difficult because of the low SNR of the images. Our strategy is to smooth the images  $I_i$  and  $I_j$  using Gaussian filters, obtaining



two sets smoothed images at different levels. The common line is computed using the smoothed versions of the images. The Radon transforms of the smoothed images are computed using Theorem 2.2.

Given a set of cryo-EM images  $\{I\}_{i=1}^M$ , we propose the following algorithm to compute the common lines.

**Algorithm 3.2.** *Compute the common lines*

1. Represent  $I_i$  ( $i = 1, \dots, M$ ) with the sum of cubic B-spline basis functions using the least square approximation.
2. Convert the cubic B-spline representation of  $I_i$  into the cubic B-spline radial basis representation.
3. For  $i = 1, \dots, M-1$ ,  $j = i+1, \dots, M$  do the following:
  - (a) For  $k = 0, \dots, K-1$ , take  $\theta_{ik} = [\cos \frac{k}{K}\pi, \sin \frac{k}{K}\pi]^T$  and compute  $R_{ik}$  using (2.12). For  $l = 0, \dots, L-1$ , take  $\theta_{jl} = [\cos \frac{l}{L}2\pi, \sin \frac{l}{L}2\pi]^T$  and compute  $R_{jl}$  using (2.12); compute  $c_{kl}$  between  $R_{ik}$  and  $R_{jl}$  using (3.16).
  - (b) Find  $(k^*, l^*)$  such that  $a_{k^*l^*} = \max a_{kl}$ , where

$$a_{kl} = \sum_{r=-n}^n \omega_r c_{k-r, l-r},$$

and  $\omega_r$  are positive weights.

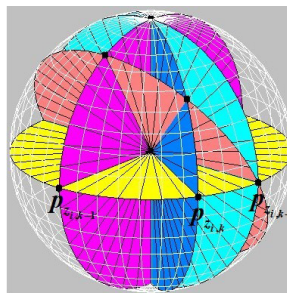
- (c) Output the common lines between image  $I_i$  and  $I_j$ :  $\theta_{ik^*}$  and  $\theta_{jl^*}$ .

In this paper, we take  $n = 2$  and  $\omega_{\pm 2} = \frac{27}{44}$ ,  $\omega_{\pm 1} = \frac{15}{44}$  and  $\omega_0 = \frac{1}{22}$  in step 3 (b) of Algorithm 3.1.

**Remark 3.2.** In the formula (3.16),  $D := \sum_r w_r D_r^T D_r$  can be previously computed. Since  $H_r$  can be truncated,  $H_r$  can be regarded as locally supported, hence  $D$  is a sparse matrix. To speed up the computation of  $c_{kl}$ , the discrete convolutions  $D_r R_{ik}$  and  $D_r R_{jl}$  can also be computed by fast Fourier transformation (FFT).

### 3.3 Determine the coordinates of the common line vectors

Assume  $I_j$ ,  $j = 1, \dots, J$  ( $J \ll M$ ), be the leading images with maximal dissimilarity. Denote the set of all the common line vectors in the Fourier transform  $\hat{I}_j$  by  $\mathbf{P}_j = \{\mathbf{p}_{j1}, \mathbf{p}_{j2}, \dots, \mathbf{p}_{jk_j}\}$ . We assume that  $\mathbf{p}_{j1}, \mathbf{p}_{j2}, \dots, \mathbf{p}_{jk_j}$  are arranged in increasing order with respect to their polar angles. Fig. 3.2 is used to illustrate the definition of the points  $\mathbf{p}_{jk}$ .



**Figure 3.2:** Intersection points  $\mathbf{p}_{jk}$  of the common lines between  $\hat{I}_j$  and  $\hat{I}_k$  with the sphere  $S^2$ .

Let  $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^N = \cup_j \mathbf{P}_j$  be the set of all the distinct common line vectors among these images. Denote the index set of points of  $\mathbf{P}_j$  in  $\mathbf{P}$  as  $\mathbf{Z}_j = \{z_{j1}, z_{j2}, \dots, z_{jk_j}\}$ . Namely,  $z_{jk} = i$  if  $\mathbf{p}_{jk} = \mathbf{p}_i \in \mathbf{P}$ . Let  $\mathbf{p} \in \mathbf{P}_i$ . If  $\mathbf{p} = \mathbf{p}_\alpha \in \mathbf{P}$ , we denote the index of  $\mathbf{p}$  as  $Z(\mathbf{p})$  which is  $\alpha$ .

Although  $\mathbf{P}$  is a set in  $\mathbb{R}^3$ , the points in  $\mathbf{P}_j$  locate in a 2D plane. Let  $(u, v)$  be the local coordinate of the plane and then each point  $\mathbf{p}_{jk}$  in  $\mathbf{P}_j$  has a  $uv$ -coordinate. Let  $\mathbf{q}_{jk} \in \mathbb{R}^2$  be the  $uv$ -coordinate representation of  $\mathbf{p}_{jk}$  and  $\mathbf{Q}_j = \{\mathbf{q}_{j1}, \dots, \mathbf{q}_{jk_j}\}$ . We assign an integer to  $\mathbf{q}_{jk}$  as  $Z(\mathbf{q}_{jk}) = Z(\mathbf{p}_{jk})$ , which is the index of  $\mathbf{p}_{jk}$ .



Suppose we have obtained  $\mathbf{Q}_j$  (using Algorithm 3.2) and  $\mathbf{Z}_j$ ,  $j = 1, \dots, J$ . For the adjacent three points  $\mathbf{q}_{z_{i,k-1}}$ ,  $\mathbf{q}_{z_{ik}}$  and  $\mathbf{q}_{z_{i,k+1}}$  in the image  $I_i$ , a linear equation is constructed as follows.

$$\mathbf{q}_{z_{ik}} = \frac{t}{\theta} \mathbf{q}_{z_{i,k-1}} + \frac{1-t}{\theta} \mathbf{q}_{z_{i,k+1}}, \quad (3.17)$$

where

$$t = \frac{u_3 v_2 - u_2 v_3}{v_3(u_1 - u_2) - u_3(v_1 - v_2)},$$

$$\theta = \frac{u_1 v_2 - u_2 v_1}{v_3(u_1 - u_2) - u_3(v_1 - v_2)},$$

and  $(u_1, v_1)$ ,  $(u_2, v_2)$  and  $(u_3, v_3)$  are the coordinates of points  $\mathbf{q}_{z_{i,k-1}}$ ,  $\mathbf{q}_{z_{i,k+1}}$  and  $\mathbf{q}_{z_{ik}}$ , respectively. Then the correspondent points in the 3D space  $\mathbf{p}_{z_{i,k-1}}$ ,  $\mathbf{p}_{z_{ik}}$  and  $\mathbf{p}_{z_{i,k+1}}$  have the same equation

$$\mathbf{p}_{z_{ik}} = \frac{t}{\theta} \mathbf{p}_{z_{i,k-1}} + \frac{1-t}{\theta} \mathbf{p}_{z_{i,k+1}}, \quad (3.18)$$

Equation (3.18) is added to the  $z_{ik}$ -th equation of the total system. Specifically, adding 1 to  $n_{z_{ik}}$ , adding  $t/\theta$  to  $a_{z_{ik}, z_{i,k-1}}$  and finally adding  $(1-t)/\theta$  to  $a_{z_{ik}, z_{i,k+1}}$ , we get the total system

$$\text{diag}[n_1, \dots, n_N]P = AP,$$

where  $P = [\mathbf{p}_1, \dots, \mathbf{p}_N]^T \in \mathbb{R}^{N \times 3}$  is unknown. Now we normalize the system as

$$P = BP \quad \text{with} \quad B = \text{diag}[n_1^{-1}, \dots, n_N^{-1}]A. \quad (3.19)$$

That is to say,  $P$  is the eigen-vector of  $B$  with respect to eigenvalue 1.

By solving the linear system (3.19) for the unknown matrix  $P$ , we determine the coordinates of the points. Finally, the orientations and the in-plane rotations of all the leading images are determined. However, as a homogeneous system,  $P = BP$  cannot be solved uniquely. From a geometry point of view, this fact can be observed by noting that the orientation problem has a rotational degree of freedom of around the origin. In order to solve the system (3.19), fix the domain of  $\hat{I}_1$  as  $XY$ -plane, then all the points in  $\mathbf{P}_1$  are known. Substituting these points into system (3.19) and moving the known terms to the right-handed side, we obtain a system in the following form

$$C\mathbf{X} = \mathbf{R}, \quad (3.20)$$

where  $C \in \mathbb{R}^{N \times (N-k_1)}$ ,  $\mathbf{X} \in \mathbb{R}^{(N-k_1) \times 3}$  consists of the unknowns from  $\mathbf{P}$  except the points on  $XY$ -plane and  $\mathbf{R} \in \mathbb{R}^{N \times 3}$ . Note that since the known points are on the  $XY$ -plane, the third column of  $\mathbf{R}$  consists of zeros. Solving system (3.20) by solving the following normal equation

$$C^T C \mathbf{X} = C^T \mathbf{R},$$

using the singular value decomposition of  $C^T C$ . Let  $C^T C = U \Sigma V^T$  be the singular value decomposition (SVD) of  $C^T C$  (see [8]) and  $\mathbf{Y} = V^T \mathbf{X}$ . Then we have

$$\Sigma \mathbf{Y} = U^T C^T \mathbf{R}, \quad \Sigma = \text{diag}[\sigma_1, \dots, \sigma_{N-k_1}],$$

Theoretically, the rank of the matrix  $C^T C$  is  $N - k_1 - 1$  in general. We therefore set  $\sigma_{N-k_1} = 0$ , and then have

$$[\mathbf{y}_1, \dots, \mathbf{y}_{N-k_1-1}]^T = \text{diag}[\sigma_1^{-1}, \dots, \sigma_{N-k_1-1}^{-1}] [U^T C^T \mathbf{R}]_{N-k_1-1},$$

where  $\mathbf{y}_i^T$  is the  $i$ -th row of  $\mathbf{Y}$  and  $[U^T C^T \mathbf{R}]_{N-k_1-1}$  consists of the first  $N - k_1 - 1$  rows of  $U^T C^T \mathbf{R}$ . Since  $\mathbf{y}_{N-k_1} = \xi = [\xi_1, \xi_2, \xi_3]^T \in \mathbb{R}^3$  is a free vector parameter, we have

$$\mathbf{X} = V_{N-k_1-1} [\mathbf{y}_1, \dots, \mathbf{y}_{N-k_1-1}]^T + \mathbf{v}_{N-k_1} \xi^T, \quad (3.21)$$

where  $V_{N-k_1-1}$  consists of the first  $N - k_1 - 1$  columns of  $V$  and  $\mathbf{v}_{N-k_1}$  is the last column of  $V$ . From (3.21), we know that the last column of  $X$  is  $\mathbf{v}_{N-k_1} \xi_3$ .

Determine the parameter  $\xi$  in  $\mathbf{X}$  such that each point has length 1. Let

$$\mathbf{x}_i^T = \mathbf{z}_i^T + v_i \xi^T, \quad i = 1, \dots, N - k_1 - 1,$$

be the  $i$ -th row of (3.21). Then, by noticing that the third component of  $\mathbf{z}_i$  is zero, we have the following equations

$$\beta v_i^2 + 2v_i \mathbf{z}_i^T \xi + \|\mathbf{z}_i\|^2 = 1, \quad i = 1, \dots, N - k_1,$$

for the unknowns  $\xi_1$ ,  $\xi_2$  and  $\beta \in \mathbb{R}$ . Solving the system in the least square sense, we obtain  $\xi = [\xi_1, \xi_2, \xi_3]^T$  with  $\xi_3 = \sqrt{\beta - \xi_1^2 - \xi_2^2}$ . We could also choose  $\xi_3$  as  $-\sqrt{\beta - \xi_1^2 - \xi_2^2}$ , and this choice leads to the  $z$ -components of points  $\mathbf{p}_i$  changing sign. Hence, the sign of  $\xi_3$  determines whether the coordinate system is left-handed or right-handed. However, using the common line information only, we cannot distinguish which system is used.

After determining the coordinates of the common line vectors of the leading images, we need orientate the remaining images. For every image  $I_\omega$ , but not the leading images, compute the common lines between  $I_\omega$  and  $I_j$ ,  $j = 1, \dots, J$ , using the algorithm in Algorithm 3.1, obtaining the set  $\mathbf{Q}_\omega = \{\mathbf{q}_{\omega,1}, \dots, \mathbf{q}_{\omega,k_\omega}\}$ . Compute the intersection point set  $\mathbf{P}_\omega = \{\mathbf{p}_{\omega,1}, \dots, \mathbf{p}_{\omega,k_\omega}\}$  of these common lines with  $S^2$ , using (3.18) and the known data  $\mathbf{P}_j$ ,  $j = 1, \dots, J$ . In (3.18), for each left-hand side in  $\mathbf{P}_\omega$ , we choose the equation such that the right-hand side is known, so that the left-hand side can be computed from the right-hand side directly.

**Remark 3.3.** When forming equation (3.17), if  $k = 1$ ,  $\mathbf{q}_{i,k-1}$  is taken as  $-\mathbf{q}_{i,k_1}$  and  $\frac{t}{\theta}$  is added to  $a_{z_{ik}, z_{i,k_1}}$ . Similarly, if  $k = k_i$ ,  $\mathbf{q}_{i,k+1}$  is taken as  $-\mathbf{q}_{i1}$  and  $\frac{1-t}{\theta}$  is added to  $a_{z_{ik}, z_{i1}}$ .

**Remark 3.4.** If the neighboring point  $\mathbf{q}_{i,k-1}$  or  $\mathbf{q}_{i,k+1}$  of  $\mathbf{q}_{ik}$  is too close to  $\mathbf{q}_{ik}$ , we will take the next neighbor instead. The closeness is judged by computing the difference of the polar angles of the points. If the difference is less than 0.1, we regard the two points as being too close.

**Computation Complexity.** Since the non-leading images are orientated separately, hence, the complexity of the orientation algorithm in this paper is linear with respect to the number of images.

After obtaining the coordinates  $\mathbf{P}_1, \dots, \mathbf{P}_M$  of the common line vectors of all the images, we use the following algorithm to compute the projection directions  $\{\mathbf{d}_i\}_{i=1}^M$  and in-plane rotation angles  $\{\gamma_i\}_{i=1}^M$ .

### 3.4 Determine projection direction $\{\mathbf{d}_i\}_{i=1}^M$ and in-plane rotation angle $\{\gamma_i\}_{i=1}^M$

**Algorithm 3.3.** Orientating one image.

Determine a vector  $\mathbf{d}_\omega \in \mathbb{R}^3$  such that

$$\|\mathbf{d}_\omega\| = 1, \quad \mathbf{p}^T \mathbf{d}_\omega = 0, \quad \forall \mathbf{p} \in \mathbf{P}_\omega. \quad (3.22)$$

This system can be solved by singular value decomposition (SVD) of the coefficient matrix. We have that  $\mathbf{d}_\omega$  is the projection direction of the image  $I_\omega$ . Let  $P$  be the coefficient matrix of the system  $\mathbf{p}^T \mathbf{d}_\omega = 0$ , and  $\mathbf{M} = P^T P \in \mathbb{R}^{3 \times 3}$ . Then,  $\mathbf{d}_\omega$  is computed as follows. Let  $\mathbf{M} = U \Sigma V^T$  be the SVD of  $\mathbf{M}$ , where  $U$  and  $V$  are orthogonal matrices in  $\mathbb{R}^{3 \times 3}$ ,  $\Sigma = \text{diag}[\sigma_1, \sigma_2, \sigma_3]$  and  $\sigma_i$  are the singular values of  $\mathbf{M}$  with  $\sigma_1 \geq \sigma_2 \geq \sigma_3$ . Since all the vectors in  $\mathbf{P}_\omega$  are located theoretically in a plane, the rank of  $\mathbf{M}$  is two. In practice, due to the noise and computational errors, the rank of  $\mathbf{M}$  is usually three. However,  $\sigma_3$  is small relative to  $\sigma_1$  and  $\sigma_2$ . We replace equation (3.22) with

$$\text{diag}[\sigma_1, \sigma_2, 0] V^T \mathbf{d}_\omega = 0, \quad \|\mathbf{d}_\omega\| = 1.$$

Since  $\|V^T \mathbf{d}_\omega\| = \|\mathbf{d}_\omega\|$ , we have  $\mathbf{d}_\omega = V[0, 0, 1]^T$ , the third column of  $V$ .

Having  $\mathbf{d}_\omega$ ,  $\mathbf{P}_\omega$  and  $\mathbf{Q}_\omega$  for image  $I_\omega$ , we compute the in-plane rotation angle  $\theta_\omega$  by the following algorithm.

**Algorithm 3.4.** Compute the in-plane rotation.

1. For  $\mathbf{d} = \mathbf{d}_\omega$ , compute firstly  $\sin \alpha$ ,  $\cos \alpha$ ,  $\sin \beta$  and  $\cos \beta$  using (2.4)-(2.6). Then compute  $\mathbf{e}_1$  and  $\mathbf{e}_2$  using (2.7) and (2.8).
2. Set

$$\tilde{\mathbf{q}}_{\omega,k} = [\mathbf{e}_1, \mathbf{e}_2]^T \mathbf{p}_{\omega,k}, \quad k = 1, \dots, k_\omega. \quad (3.23)$$

Since  $\mathbf{q}_{\omega,k}$  are computed using the  $\tilde{\mathbf{e}}_1 \tilde{\mathbf{e}}_2$  coordinate system (2.1) and (2.2), then the in-plane rotation angle  $\theta_\omega$  is determined by minimizing the following energy function

$$E_1(\theta_\omega) = \sum_{k=1}^{k_\omega} \|R(\theta_\omega) \mathbf{q}_{\omega,k} - \tilde{\mathbf{q}}_{\omega,k}\|^2, \quad (3.24)$$

or

$$E_2(\theta_\omega) = \sum_{k=1}^{k_\omega} \|R(\theta_\omega) \mathbf{q}_{\omega,k} - E \tilde{\mathbf{q}}_{\omega,k}\|^2, \quad (3.25)$$

where  $R(\theta_\omega)$  is the rotation matrix defined by  $\theta_\omega$ :

$$R(\theta_\omega) = \begin{bmatrix} \cos \theta_\omega & -\sin \theta_\omega \\ \sin \theta_\omega & \cos \theta_\omega \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

$\theta_\omega$  can be computed by solving the equation  $E'_1(\theta_\omega) = 0$  or  $E'_2(\theta_\omega) = 0$ . The equation  $E'_1(\theta_\omega) = 0$  can be represented as

$$\sum_{k=1}^{k_\omega} \tilde{\mathbf{q}}_{\omega,k}^T \begin{bmatrix} \sin \theta_\omega & \cos \theta_\omega \\ -\cos \theta_\omega & \sin \theta_\omega \end{bmatrix} \mathbf{q}_{\omega,k} = 0.$$

It can be written as  $\cos(\theta_\omega)a = \sin(\theta_\omega)b$  with

$$a = \sum_{k=1}^{k_\omega} \tilde{\mathbf{q}}_{\omega,k}^T \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{q}_{\omega,k}, \quad b = \sum_{k=1}^{k_\omega} \tilde{\mathbf{q}}_{\omega,k}^T \mathbf{q}_{\omega,k}.$$

- (a) If  $a = b = 0$ ,  $\theta_\omega$  can be any number, and then we take  $\theta_\omega = 0$ .
- (b) If  $a = 0$  and  $b \neq 0$ ,  $\theta_\omega = 0$  or  $\theta_\omega = \pi$ .
- (c) If  $a \neq 0$ ,  $\theta_\omega = \cot^{-1}(\frac{b}{a}) \in (0, \pi)$  or  $\theta_\omega = \cot^{-1}(\frac{b}{a}) + \pi \in (\pi, 2\pi)$ .

It is easy to see that one of the two zero points of equation  $E'_1(\theta_\omega) = 0$  is the minimal point of  $E_1(\theta_\omega)$  and that another one is the maximal point. By computing the second order derivative of  $E_1(\theta_\omega)$ , we have

$$E''_1(\theta_\omega) = \sum_{k=1}^{k_\omega} \tilde{\mathbf{q}}_{\omega,k}^T R(\theta_\omega) \mathbf{q}_{\omega,k}.$$

The zero point with  $E''_1(\theta_\omega) = \sum_{k=1}^{k_\omega} \|\mathbf{q}_{\omega,k}\|^2 > 0$  is the minimal point of  $E_1$ .

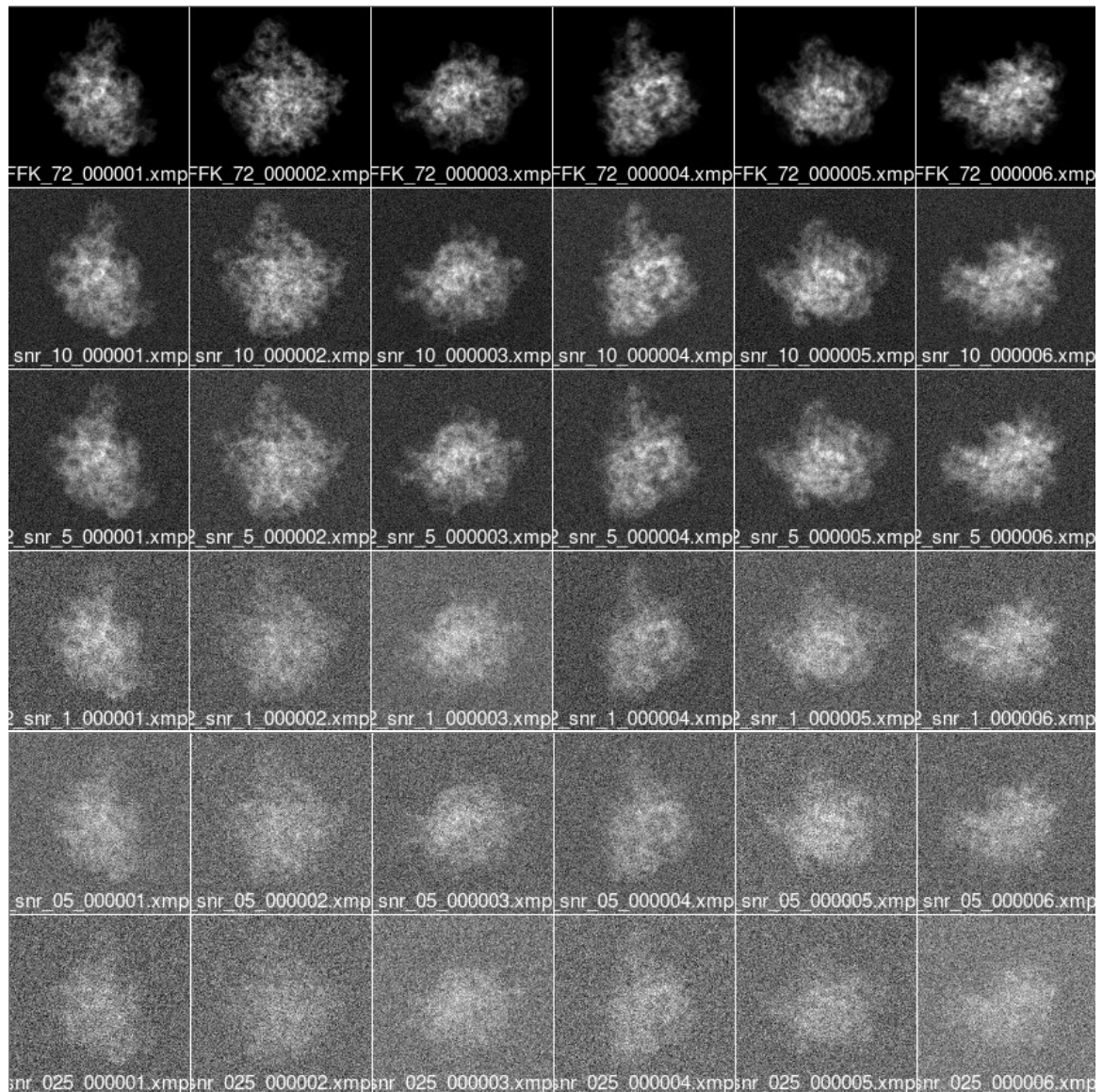
Equation  $E'_2(\theta_\omega) = 0$  is solved similarly. Among the two minimal points of  $E_1$  and  $E_2$ , the one making the energy function smaller is what we want.

By now, we give the whole orientation algorithm.

### Orientation Algorithm Outline

1. Select  $J (\ll M)$  images named as leading images, from the set  $\{I_i\}_i^M$  using Algorithm 3.1.
2. Compute the common lines between leading images using Algorithm 3.2.
3. Set the orientation of first leading image  $I_0$  as  $\mathbf{d} = [0, 0, 1]^T$  and set the in-plane rotation angle of  $I_0$  as zero. Then determine the coordinates of the common line vectors of the leading images.
4. Compute the coordinates of the common line vectors of the non-leading images.
5. Determine the projection directions and the in-plane rotation angles of all the images using Algorithm 3.3 and 3.4.

## 4 Experimental Results

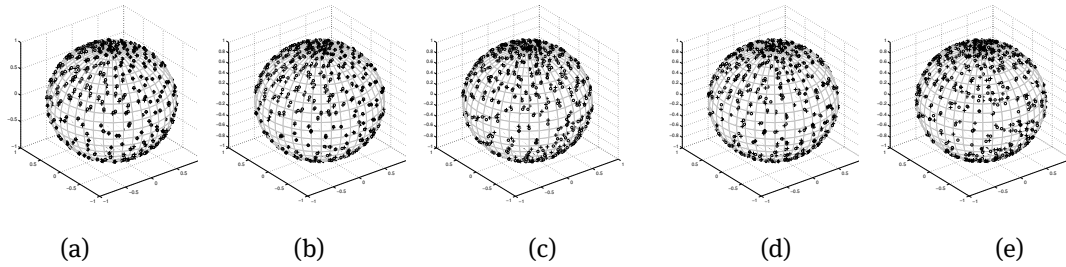


**Figure 4.3:** From the top row to the bottom row, the projection images without noise and with noise at SNR=10.0, 5.0, 1.0, 0.5, 0.25, respectively.

In this section, we use the simulated cryo-EM images to validate our algorithm. In order to produce the simulated cryo-EM images, a 3D density map (a volume data) with size  $140^3$  is synthesized using the crystal structure file 1FFK.pdb<sup>1</sup>, which is the largest ribosomal subunit. Then, a dataset containing 10000 images is produced by projecting the 3D map using 10000 randomly distributed orientations. In order to explore the robustness of the proposed algorithm, we then add white Gaussian noise to the dataset with different SNRs

<sup>1</sup> <http://www.rcsb.org/pdb/explore.do?structureId=1ffk>





**Figure 4.4:** (a)-(e) show respectively the orientation results of  $K$  representatives with SNR=10, 5, 1, 0.5, 0.25. The circle points are the correct directions and the cross points are the computed directions by the proposed orientation algorithm.

to produce another five noisy datasets. We take SNR = 10.0, 5.0, 1.0, 0.5, 0.25, respectively. Fig. 4.3 shows six images of the original dataset and their noisy versions at different noise levels.

In the experiment, we apply the proposed orientation algorithm to the 6 datasets. We take  $K = 256$  and  $J = 6$ . For the image sets with higher SNR (SNR=10.0, 5.0), the average-based denoising is not applied in the step 4 of Algorithm 3.1.

In the following, we first show the orientation results of the 256 representatives. Table 4.1 gives the percentages of the orientation angular error less than 5 and 10 degrees in all directions. Fig. 4.4 shows the computed directions (the points labeled with a cross) and the exact projection directions (labeled with a circle) on the unit sphere. From Fig. 4.4(a)–(e), we can see that, as the SNR decreases, the orientation results become less and less accurate. For SNR = 10.0 and SNR = 5.0, the errors between the computed and the exact projection directions are very small.

**Table 4.1:** The percentage of the orientation error less than 10 and 5 degrees in all directions under different SNRs.

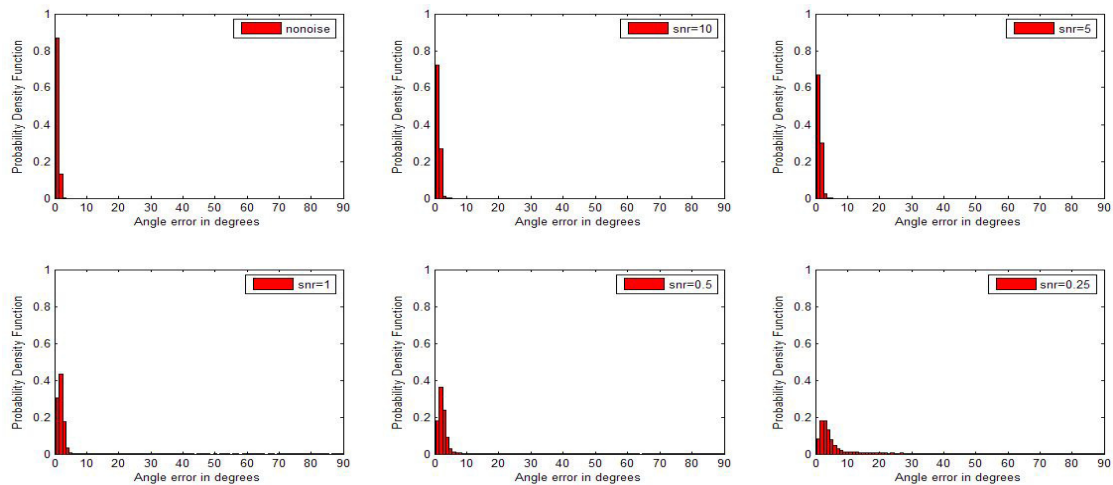
SNR	Errors less than 10°	Errors less than 5°
SNR = 10	100%	100%
SNR = 5	100%	100%
SNR = 1	99.61%	96.48%
SNR = 0.5	98.85%	84.35%
SNR = 0.25	93.36%	79.69%

The orientation results of 10000 projections without noise and with noise for SNR = 10, 5, 1, 0.5, 0.25 are then presented. Fig. 4.5 shows the histogram of the errors between the computed directions and the exact directions of the projection images. From this figure, we can see that the orientation results are reasonably good even if the noise level is high. Furthermore, the orientation algorithm is robust since the average-based denoising algorithm is used. The algorithm is also very efficient. It takes about 40 minutes to compute the projection directions of 10000 projections using a computer with an Intel Xeon 2.40 GHz CPU.

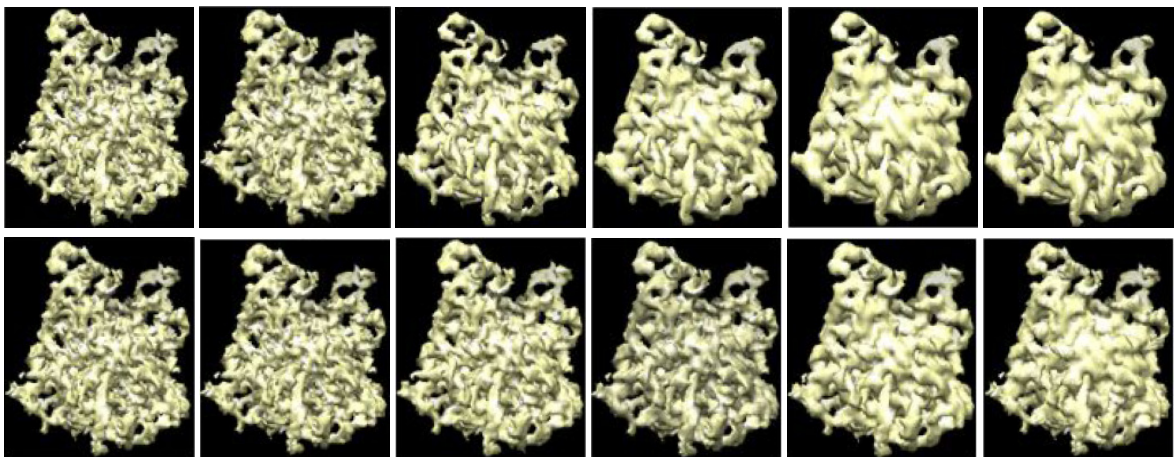
Finally, two sets of volume data are reconstructed using the weighted back-projection (WBP) method in the software *xmipp* (*xmipp\_reconstruct\_wbp* (see [7])) for the following two cases:

1. Using 10000 projection images with noises at different SNRs and using the exact projection directions.
2. Using 10000 projection images with noise at different SNRs and using the projection directions computed from the noised data.

To show the similarity of these volume data, iso-surfaces, as shown in Fig. 4.6, are extracted from them, taking the iso-values as the middle values of the volume data. It can be seen that the reconstructed results using the computed projection directions are similar to the reconstructed results using the exact projection directions. Hence, the orientation algorithm yields reliable results.



**Figure 4.5:** The histogram of the error between the computed directions and the totally correct directions under SNR= $\infty$ , 10, 5, 1, 0.5, 0.25. The error 90 degrees is worst result.



**Figure 4.6:** Iso-surfaces of the reconstructed data. Figures in the first row show the iso-surfaces of the reconstructed volume data using 10000 projections with noises at different SNRs and using the exact projection directions. Figures in the second row show the iso-surfaces of the reconstructed volume data using 10000 projections with noise at different SNRs and using the projection directions computed from the noised data.

In Table 4.2, we list the resolutions for each of the reconstructed volumes. The resolution is computed as the point at where the Fourier Shell Correlation (FSC) value is 0.5. The Fourier shell is computed between the reconstructed volume and the exact volume data. It can be seen that when SNR is lower ( $\text{SNR} \leq 1$ ), the resolutions of the reconstructed volume using the computed projection directions are even better than that of the reconstructed volumes using the exact projection directions. This amazing fact is owing to that when the adding noise to the projected images is high, the exact project directions are possibly no longer good to the exact volume. Hence, the reconstructed volumes may further away from the exact volume data.

## 5 Conclusions

An efficient and reliable orientation algorithm based on common lines has been presented. The algorithm is not sensitive to the noise, since the leading images, the weighted cross-correlation function and average-based denoising are used. The computational complexity of the algorithm is linear with respect to the number

**Table 4.2:** The resolutions of the reconstruction volumes using exact projection directions (EPD) and computed projection directions (CPD) under different SNRs.

SNR	5	1	0.5	0.25
Resolutions using EFDs	FSC > 0.5	5.74 Å	6.95 Å	8.97 Å
Resolutions using CPDs	4.26 Å	5.67 Å	6.82 Å	8.77 Å

of projection images. The experiments show that the proposed orientation algorithm gives reasonably good results for projection images with different noise levels.

**Acknowledgement:** Project support in part by NSFC under the grants 81173663, NSFC Fund for Creative Research Groups of China (grant No. 11321061).

**Conflict of interest:** Authors state no conflict of interest.

## References

- [1] T. S. Baker, Chen and R. H. Cheng. A Model-Based Approach for Determining Orientations of Biological Macromolecules Imaged by Cryoelectron Microscopy. *Journal of Structural Biology*, 116(1):120-130, 1996.
- [2] C. Chen and G. Xu. Gradient-flow-based semi-implicit finite-element method and its convergence analysis for image reconstruction. *Inverse Problems*, 28(3):035006, 2012.
- [3] R. A. Crowther, D. J. DeRosier, and A. Klug. The reconstruction of a three-dimensional structure from projections and its application to electron microscopy. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 317(1530):319–340, 1970.
- [4] R. A. Crowther. Procedures for Three-Dimensional Reconstruction of Spherical Viruses by Fourier Synthesis from Electron Micrographs. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 261(837):221–228, 1971.
- [5] S. D. Fuller, S. J. Butcher, R. H. Cheng, and T. S. Baker. Three-Dimensional Reconstruction of Icosahedral Particles-The Uncommon Line. *J. Struct. Biol.*, 116:48–55, 1996.
- [6] J. Frank. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State*. Oxford, 2006.
- [7] Frank.J.(Ed.). *Electron Tomography Methods for Three-dimensional Visualization of Structures in the cell*. Springer, New York.
- [8] G. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.
- [9] A. Gopinath, G. Xu, D. Ress, O. Oktem, S. Subramaniam, and C. Bajaj. Shape-based Regularization of Electron Tomographic Reconstruction. *IEEE Trans Med Imaging*, 31(12):2241–2252, 2012.
- [10] M. Van Heel. Angular reconstitution: a posteriori assignment of projection directions for 3d reconstruction. *Ultramicroscopy*, 21(2):111–124, 1987.
- [11] M. Li, G. Xu, C. O. S. Sorzano, F. Sun, and C. L. Bajaj. Single-Particle Reconstruction Using  $L^2$ -Gradient Flow. *Journal of Structural Biology*, 176:259–267, 2011.
- [12] S. J. Ludtke, P. R. Baldwin, and W. Chiu. EMAN: semiautomated software for high-resolution single-particle reconstructions. *Journal of Structural Biology*, 128(1):82-97, 1999.
- [13] F. Natterer. *The Mathematics of Computerized Tomography*. SIAM: Society for Industrial and Applied Mathematics, 2001.
- [14] F. Natterer and F. Wübbeling. *Mathematical Methods in Image Reconstruction*. SIAM, 2001.
- [15] P. A. Penczek, R. A. Grasucci, and J. Frank. The ribosome at improved resolution: New techniques for merging and orientation refinement in 3D cryo-electron microscopy of biological particles. *Ultramicroscopy*, 53:251–270, 1994.
- [16] L. Piegl and W. Tiller. *The NURBS Book*. Springer, 1997.
- [17] M. Radermacher. Three-dimensional reconstruction from random projections—orientational alignment via Radon transforms. *Ultramicroscopy*, 53:121–136, 1994.
- [18] M. Radermacher. The three-dimensional reconstruction of single particles from random and non-random tilt series. *J. Electron Microsc. Tech.*, 9:359–394, 1988.
- [19] P. B. Rosenthal and R. Henderson. Optimal Determination of Particle Orientation, Absolute Hand, and Contrast Loss in Single-particle Electron Cryomicroscopy. *Journal of Molecular Biology*, 333(4):721–745, 2003.
- [20] H. R. Saibil. Macromolecular structure determination by cryo-electron microscopy. *Acta Cryst. D*, 56:1215–1222, 2000.



- [21] S. H. W. Scheres. RELION: Implementation of a Bayesian approach to cryo-EM structure determination. *J. Struct. Biol.*, 180(3):519–530, 2012.
- [22] 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.*, 28:296–312, 2010.
- [23] Y. Shkolnisky and A. Singer. Viewing direction estimation in cryo-em using synchronization. *SIAM J. Imageing Sciences*, 5(3):1088–1110, 2012.
- [24] A. Singer and Y. Shkolnisky. Three-dimensional structure determination from common lines in cryo-em by eigenvectors and semidefinite programming. *SIAM J. Imageing Sciences*, 4(2):543–572, 2011.
- [25] L. Wang, A. Singer and Z. Wen. Orientation Determination from Cryo-EM images Using Least Unsquared Deviation. <http://arxiv.org/pdf/1211.7045v1>.
- [26] C. O. S. Sorzano, S. Jonic, C. El-Bez, J. M. Carazo, S. De Carlo, P. Thévenaz, and M. Unser. A multiresolution approach to pose assignment in 3-D electron microscopy of single particles. *J. Struct. Biol.*, 146:381–392, 2004.
- [27] X. Wang and G. Xu. A Fast Classification Method for Single-Particle Projections with a Translation and Rotation Invariant. *Journal of Comp. Math.*, 31:2(2013), 137-153.
- [28] G. Xu, M. Li, A. Gopinath, and C. L. Bajaj. Inversion of Electron Tomography Images Using  $L^2$ -Gradient Flows—Computational Methods. *Journal of Computational Mathematics*, 29(5):501–525, 2011.