

# Mathematical Methods in Biomedical Imaging

## Final Project

Itamar Tzafrir

Based on: Shasha Geva A. and Shkolnisky Y. "A common lines approach for ab-initio modeling of molecules with tetrahedral and octahedral symmetry" SIAM Journal on Imaging Sciences, 16(4) 1978-2014, 2023

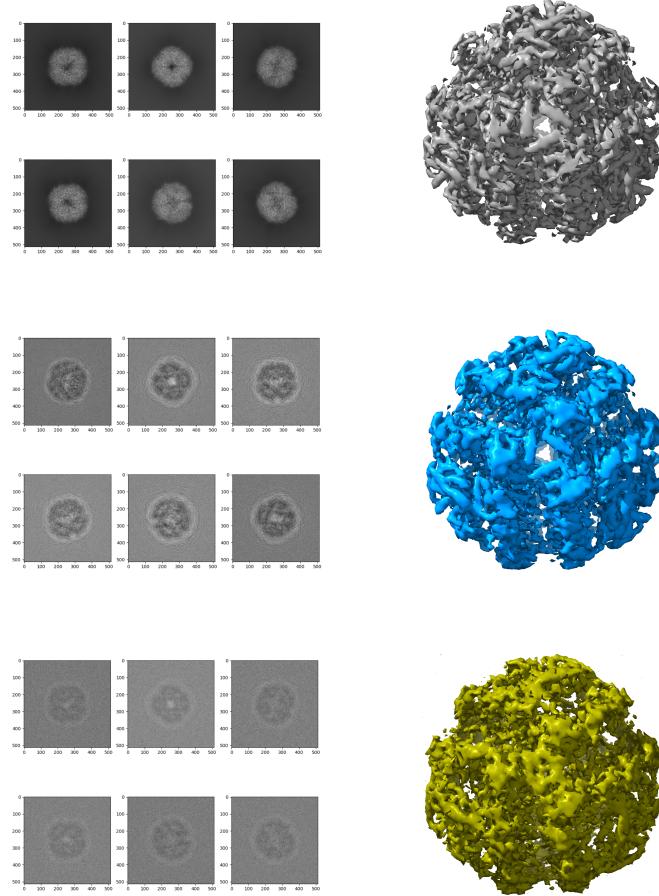


Figure 0: Reconstructions of simulated projections of the 'T' symmetric EMD-10835 using the common-lines python algorithm. For each simulation, 70 projections of the molecule were downsampled to size 129x129 under CTF distortion and various noise levels. On the left are the projections before downsampling, on the right are the reconstructed molecules. Top result is without induced noise or CTF distortion.

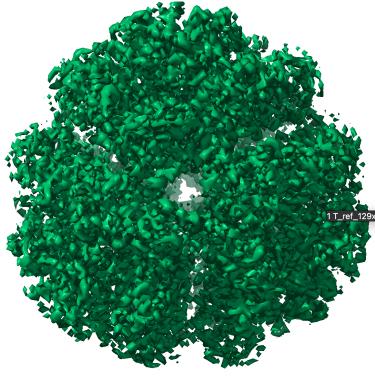


Figure 1: Reference model for EMD-10835 downsampled to resolution 129x129x129

## Tetrahedral/Octahedral Reconstruction

Determining the 3D structure of molecules is important to scientists as it reveals how molecules function, interact, and evolve, enabling breakthroughs in drug design, structural biology, materials science, and biotechnology.

Some molecular structures exhibit certain kinds of symmetries which can play a crucial role in accurately determining their structures. Among these symmetry types are Tetrahedral (T) and Octahedral (O) which are the focus of this work.

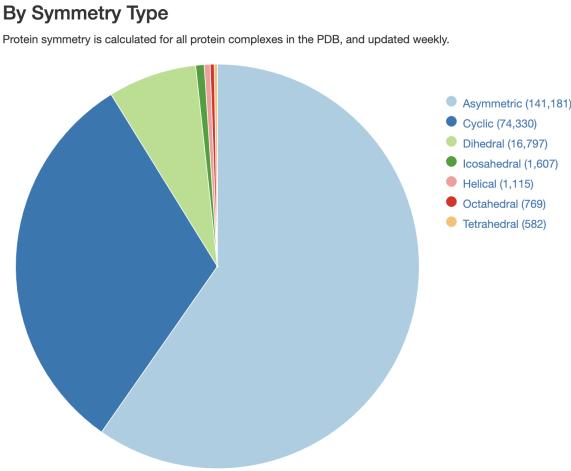


Figure 2: Proteins are macromolecules (large molecules) made up of amino acid chains. Proteins whose 3D structure has been determined are typically stored in the PDB. This figure shows the distribution of the proteins in the PDB by their symmetry type.

One of the methods for obtaining a molecule's 3D structure is cryo-electron microscopy (cryo-EM) which involves imaging frozen copies of the investigated molecule by an electron-microscope, with each copy as-

suming some unknown random orientation fixed at the moment of freezing.

The task of finding the orientation of the molecule, corresponding to each projection-image when frozen, is known as the “**orientation assignment problem**”, and is the main objective of this work. Once the orientations have been estimated it is possible to reconstruct the volume using the projection-images.

We denote the electrostatic potential of the molecule by  $\psi : \mathbb{R}^3 \rightarrow \mathbb{R}$  and a set of  $N$  rotation matrices by  $R_i \in SO(3)$ ,  $i \in \{1, \dots, N\}$ . Then the projection-image that was generated by imaging the molecule frozen at rotation  $R_i$ , denoted  $P_{R_i}$  is given by

$$P_{R_i}(x, y) = \int_{-\infty}^{\infty} \psi \left( R_i \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \right) dz = \int_{-\infty}^{\infty} \psi(x R_i^1 + y R_i^2 + z R_i^3) dz$$

Here  $R_i^1, R_i^2, R_i^3$  are the columns of matrix  $R_i$ .

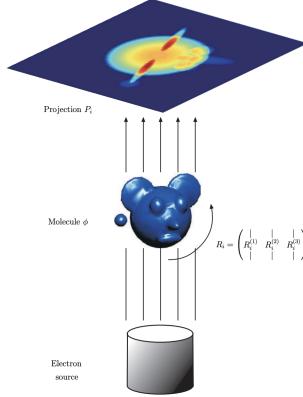


Figure 3: Imaging setup

Every projection-image is compatible with two distinct orientations (“**handedness ambiguity**”). If we denote the reflection matrix through the  $xy$  plane:  $J = \text{diag}(1, 1, -1)$  and  $\tilde{\psi}$  the mirrored image of the molecule  $\psi$  then:

$$P_{R_i}(x, y) = \tilde{P}_{JR_iJ}(x, y)$$

where  $\tilde{P}$  is a projection-image generated from  $\tilde{\psi}$ . Thus a projection-image of the molecule  $\psi$  at orientation  $R_i$  is identical to the projection-image of  $\tilde{\psi}$  at orientation  $JR_iJ$ .

In this project our algorithm solves the “orientation assignment problem” for molecules with tetrahedral or octahedral symmetry denoted by  $\mathbb{T} \subset SO(3)$  and  $\mathbb{O} \subset SO(3)$  respectively.  $\mathbb{T}$  contains 12 elements and  $\mathbb{O}$  contains 24. Molecules exhibiting these symmetries create “**symmetry ambiguity**”. If a molecule  $\psi$  has symmetry  $G$  (either  $\mathbb{T}$  or  $\mathbb{O}$ ) then for each group element  $g \in G$  we get

$$P_{R_i}(x, y) = P_{gR_i}(x, y)$$

Thus all rotations of form  $gR_i$  are consistent with the projection  $P_{R_i}$  for any group element  $g \in G$ .

Next we define the two-dimensional Fourier transform of a projection-image:

$$\hat{P}_{R_i}(\omega_x, \omega_y) = \iint_{\mathbb{R}^2} P_{R_i}(x, y) e^{-i(x\omega_x + y\omega_y)} dx dy$$

As well as the three-dimensional Fourier transform of the molecule:

$$\hat{\psi}(\omega_x, \omega_y, \omega_z) = \iiint_{\mathbb{R}^3} \psi(x, y, z) e^{-i(x\omega_x + y\omega_y + z\omega_z)} dx dy dz$$

The Fourier projection slice theorem states that:

$$\hat{P}_{R_i}(\omega_x, \omega_y) = \hat{\psi}(\omega_x R_i^1 + \omega_y R_i^2), \quad (\omega_x, \omega_y) \in \mathbb{R}^2$$

Essentially, the Fourier transform of a projection-image equals the restriction of the Fourier transform of the molecule to the plane spanned by  $R_i^1, R_i^2$ .

If we consider the planes corresponding to projections  $\hat{P}_{R_i}$  and  $\hat{P}_{R_j}$ , as they both cross the origin, they must share a **common line** and thus any pair of projections share of pair lines on which their Fourier transforms are identical.

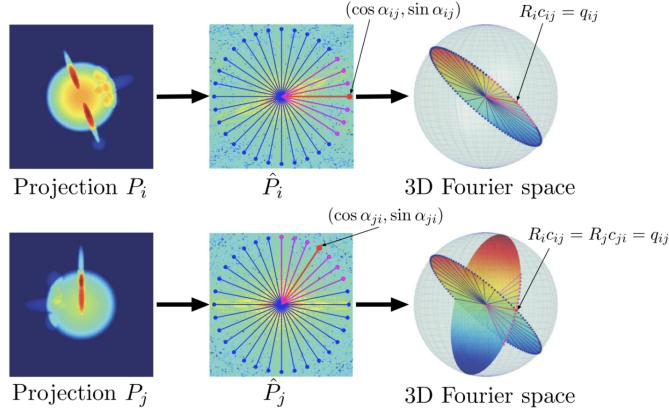


Figure 4: Fourier projection-slice theorem and the common line property

Due to the symmetric structure of the molecule, each pair of Fourier transforms of projections  $\hat{P}_{R_i}$  and  $\hat{P}_{R_j}$  share  $n$  pairs of common lines together where  $n$  is the order of the symmetry group. Similarly each Fourier transform of a projection  $\hat{P}_{R_i}$  shares a common line with  $\hat{P}_{gR_i}$  for any non trivial group element  $g \in G$ .

## Algorithm

### Generating the rotation set $\text{SO}_G(3)$

Throughout our algorithm, we're interested in optimizing over the set of rotation matrices  $SO(3)$ . As this is impossible to perform efficiently, we limit ourselves to a finite subset  $\text{SO}_G(3) \subset SO(3)$  over which we perform our optimization. We start creating  $\text{SO}_G(3)$  by generating a grid of approximately equally spaced rotations based on a given resolution parameter “rotation\_resolution” which signifies the the number of sample angles per  $2\pi$ .

Resolution	# Rotations in grid
50	4,484
75	15,236
100	39,365
150	129,835

However due to the symmetric nature of the molecule, we can reduce the number of rotations whilst maintaining the accuracy of the algorithm by removing rotations that have close proximity (below a specified threshold) to other rotations in the set after being multiplied by some group element. The proximity is determined based on their representation using viewing direction and in-plane rotation, it is possible to specify these thresholds for the algorithm. After symmetry based filtering with the default thresholds used by the algorithm, the size of the set  $\text{SO}_G(3)$  is as follows:

Resolution	# Rotations in grid	# Rotations in $\text{SO}_{\mathbb{T}}(3)$	# Rotations in $\text{SO}_{\mathbb{O}}(3)$
50	4,484	1,195	724
75	15,236	1,541	854
100	39,365	1,704	887
150	129,835	2,084	1,068

Next, for each pair of rotation matrices in the set of candidate matrices  $\text{SO}_{\mathbb{T}}(3)$ , we calculate the pairs of common lines. Again as we're dealing with finite precision and memory (as well as the limited resolution of the images) we limit ourselves to some angular resolution ("set of all possible common lines") which is determined by the parameter "n\_theta" which is 360 by default. Thus for each pair of common lines we store the indices  $(0, 1, \dots, n_{\theta} - 1)$  corresponding to the common lines for each rotation. These common line indices are calculated using the formula

$$\begin{aligned}\alpha_{R_i, R_j}^{k,1} &= \text{atan2} \left( \left( R_i^T g^{(k)} R_j \right)_{1,3} - \left( R_i^T g^{(k)} R_j \right)_{2,3} \right) \\ \alpha_{R_i, R_j}^{k,2} &= \text{atan2} \left( - \left( R_i^T g^{(k)} R_j \right)_{3,1}, \left( R_i^T g^{(k)} R_j \right)_{3,2} \right)\end{aligned}$$

Here we denote by  $\alpha_{R_i, R_j}^{k,1}$  and  $\alpha_{R_i, R_j}^{k,2}$  the angles between  $q_{R_i, R_j}^k$  and the local  $x$  axes of the planes.

We also calculate the self-common-lines indices for each rotation matrix in  $\text{SO}_G(3)$ . Instead of computing  $|G| - 1$  of these for each matrix in  $\text{SO}_G(3)$  we discard some of the common lines due to redundancy. The group elements used for self common lines can be found in "scl\_inds" and contains 7 elements for  $T$  symmetry and 17 elements for  $O$  symmetry. Same as with the common-lines, the self common lines are stored in values from  $(0, 1, \dots, n_{\theta} - 1)$  and calculated by:

$$\begin{aligned}\alpha_{R_i, R_i}^{k,1} &= \text{atan2} \left( \left( R_i^T g^{(k)} R_i \right)_{1,3} - \left( R_i^T g^{(k)} R_i \right)_{2,3} \right) \\ \alpha_{R_i, R_i}^{k,2} &= \text{atan2} \left( - \left( R_i^T g^{(k)} R_i \right)_{3,1}, \left( R_i^T g^{(k)} R_i \right)_{3,2} \right)\end{aligned}$$

Where  $k$  is an element from the "scl\_inds" subset of  $G$ .

The data calculated in this step is independent from the molecule and thus can be reused for future calculating. It is thus stored as a cache file and can be supplied as an argument for the algorithm in future runs for efficient computation.

## Polar Fourier transform of projections

For each projection-image we compute its polar Fourier transform, with resolution based on the arguments provided to the algorithm –  $n_{\theta}$  rays each with  $n_r$  coefficients.  $n_r$  is by default half the width of the projection images' width. The advantage of using the polar transform as opposed to the cartesian is that it's natural when extracting lines and comparing them as they are already in the representation. Using cartesian

coordinates would require interpolating to extract the lines which is prone to errors and less efficient. The transformations are than normalized.

### Approximating $\{R_i^T g^{(k)} R_j\}_{k=1}^n$

Our first goal, before estimating the rotation matrices themselves, will be to find an approximation to a pair of rotation matrices  $(R_{ij}, R_{ji})$  satisfying

$$\left\{R_{ij}^T g^{(k)} R_{ji}\right\}_{k=1}^n = \left\{R_i^T g^{(k)} R_j\right\}_{k=1}^n$$

for each pair of rotation matrices  $R_i, R_j$ . We denote these estimates as  $(\tilde{R}_{ij}, \tilde{R}_{ji})$ . To find such estimates, we use a scoring function  $\pi_{ij} : \mathrm{SO}_G(3) \times \mathrm{SO}_G(3) \rightarrow [0, 1]$  which for any two rotations  $Q_r, Q_s \in \mathrm{SO}_G(3)$  computes a score that indicates how well  $\{Q_r^T g^{(k)} Q_s\}_{k=1}^n$  approximates  $\{R_i^T g^{(k)} R_j\}_{k=1}^n$ . We search through every pair of rotations in the candidate set  $\mathrm{SO}_G(3)$  and choose the pair with the highest score as our estimate  $(\tilde{R}_{ij}, \tilde{R}_{ji})$ .

For each pair of projection-images  $P_{R_i}$  and  $P_{R_j}$  we iterate through every pair of rotation matrices  $Q_r, Q_s \in \mathrm{SO}_G(3)$ . We've previously calculated the common lines (and self common lines) for these two rotations. If these rotations serve as a good approximation, the corresponding lines in the Fourier projections should match and be equal. To check how well they match we calculate the vector inner product (complex) of each pair, since the rays in the transforms are normalized, we'd expect the result to be 1 for a match.

Formally, we define  $v_{i,\theta}(\xi)$  to be the value of coefficient  $\xi$  in ray  $\theta$  of the polar Fourier transform of  $P_{R_i}$  where  $\theta \in \{0, 1, \dots, n_\theta - 1\}$  and  $\xi \in \{0, \dots, n_r - 1\}$ . Then we define

$$\rho_{ij}(\theta, \phi) = \Re \langle v_{i,\theta}, v_{i,\phi} \rangle_{\mathbb{C}^{n_r}}$$

The real part of the (already normalized) cross correlations between  $v_{i,\theta}, v_{i,\phi}$ . Now if we denote the indices of the common lines between  $Q_r, Q_s$  as  $\left\{\left(\alpha_{Q_r, Q_s}^{k,1}, \alpha_{Q_r, Q_s}^{k,2}\right)\right\}$ , we define the score function  $\pi_{ij}$  as:

$$\pi_{ij}(Q_r, Q_s) := \prod_{1 \leq k \leq |G|} \rho_{ij} \left( \alpha_{Q_r, Q_s}^{k,1}, \alpha_{Q_r, Q_s}^{k,2} \right)$$

We use a product as each term is less than or equal to 1 and so by maximizing the product we enforce all the correlations are large simultaneously which will be effective in reducing error due to noise.

We also incorporate the self-common-lines that we calculated for each pair  $Q_r, Q_s \in \mathrm{SO}_G(3)$  into the score function in a similar manner. So our updated score function is

$$\pi_{ij}(Q_r, Q_s) := \prod_{1 \leq k \leq |G|} \rho_{ij} \left( \alpha_{Q_r, Q_s}^{k,1}, \alpha_{Q_r, Q_s}^{k,2} \right) \prod_{k \in \text{scl\_inds}} \rho_{ii} \left( \alpha_{Q_r, Q_r}^{k,1}, \alpha_{Q_r, Q_r}^{k,2} \right) \rho_{jj} \left( \alpha_{Q_s, Q_s}^{k,1}, \alpha_{Q_s, Q_s}^{k,2} \right)$$

In order to handle shifts in the projection-images (in practice the projections aren't perfectly aligned) we also consider the presence of unknown shifts. A shift in the projection-image corresponds to a change in phase in the Fourier transform and so when calculating the cross correlations we also maximize over a finite set of possible shifts (multiplying by the different possible phases).

## Handedness Synchronization

Due to the handedness ambiguity and the nature of the symmetry groups, the set  $\left\{ (JR_i J)^T g^{(k)} JR_j J \right\}_{k=1}^n$  also maximizes  $\pi_{ij}$  and as a result, each pair  $(\tilde{R}_{ij}, \tilde{R}_{ji})$  estimates either the pair  $(R_{ij}, R_{ji})$  or the pair  $(JR_{ij}J, JR_{ji}J)$ , and each pair could estimate either of the two independently of the other pairs. To handle this, we apply the handedness synchronization procedure in order to split the set  $\left\{ (\tilde{R}_{ij}, \tilde{R}_{ji}) \right\}$  into two subsets that estimate with or without the  $J$  conjugation:

$$A = \left\{ (\tilde{R}_{ij}, \tilde{R}_{ji}) \mid (\tilde{R}_{ij}, \tilde{R}_{ji}) \text{ estimates } (R_{ij}, R_{ji}) \right\}$$

$$B = \left\{ (\tilde{R}_{ij}, \tilde{R}_{ji}) \mid (\tilde{R}_{ij}, \tilde{R}_{ji}) \text{ estimates } (JR_{ij}J, JR_{ji}J) \right\}$$

Then we choose one of the subsets and replace each element  $(\tilde{R}_{ij}, \tilde{R}_{ji})$  with  $(J\tilde{R}_{ij}J, J\tilde{R}_{ji}J)$  making all the estimates consistent with the same handedness.

## Estimating the Rotations $R_i$

We begin by stating a corollary proved in the paper to be used in this step. Proof of this corollary involves showing the normalizer of the groups  $\mathbb{O}$  and  $\mathbb{T}$  equal  $\mathbb{O}$ .

**Corollary 3.** Let  $R_{ij}, R_{ji}$  and  $R_i, R_j$  be two pairs of rotation matrices satisfying  $\left\{ R_{ij}^T g^{(k)} R_{ji} \right\}_{k=1}^n = \left\{ R_i^T g^{(k)} R_j \right\}_{k=1}^n$ .

- For  $G = \mathbb{T}$  it holds that

$$R_{ij} = h_{ij}g_{ij}R_i, \quad R_{ji} = h_{ij}g_{ji}R_j, \quad g_{ij}, g_{ji} \in \mathbb{T}, \quad h_{ij} \in \mathbb{O}$$

- For  $G = \mathbb{O}$  it holds that

$$R_{ij} = g_{ij}R_i, \quad R_{ji} = g_{ji}R_j, \quad g_{ij}, g_{ji} \in \mathbb{T}$$

This step also relies on a specific representation of the rotation matrices in the groups (or equivalently, the axes through which we perform the rotations). This choice of axes allows us to represent each group element as a matrix with exactly one non-zero entry (either 1 or -1) in each row and each column. Consequentially another property of these matrices is that they can each be represented uniquely using addition and subtraction of single entry matrices.

**Definition 4 (single-entry matrix):** A single-entry matrix, denoted by  $e_{ij} \in \mathbb{R}^{3 \times 3}$  is a matrix whose  $(i, j)$  element is one and the rest of its elements are zero.

**Definition 5 (one-line notation):** Given a symmetry group element  $g$  from either  $\mathbb{T}$  or  $\mathbb{O}$ , we define its one-line notation as the vector  $\sigma = (\sigma(1), \sigma(2), \sigma(3))$  given by

$$\sigma = g \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

**Lemma 6.** Each group element from either  $\mathbb{T}$  or  $\mathbb{O}$  can be represented uniquely by the sum

$$g = e_{1\sigma(1)} + e_{2\sigma(2)} + e_{3\sigma(3)}$$

Due to the symmetry ambiguity all orientations assignments of the form  $\{g_i R^i\}_{i=1}^N$ , where  $g_i \in G$  is some arbitrary symmetry group element, are consistent with the provided set of projection-images. This means there are  $n^N$  valid assignments, and a single one will suffice.

Now we denote the rows of  $R_i$  by  $v_i^{(m)}$  i.e.

$$R_i = \begin{pmatrix} - & v_i^{(1)} & - \\ - & v_i^{(2)} & - \\ - & v_i^{(3)} & - \end{pmatrix}$$

For some assignment  $\{g_i R^i\}_{i=1}^N$ ,  $g_i \in G$  we denote by  $\sigma_i$  the one line notation of  $g_i$  and  $\sigma = (\sigma_1, \dots, \sigma_N)$ . We also define the  $3N$  length concatenated vector:

$$v_{\sigma, \tau(m)} = (v_1^{\sigma_1(\tau(m))}, \dots, v_N^{\sigma_N(\tau(m))})$$

Essentially a vectorized concatenation of the  $\tau(m)$ th rows of all  $g_i R_i$ . Finally we define

$$H_{\sigma, m} = v_{\sigma, \tau(m)}^T v_{\sigma, \tau(m)}, \quad m = 1, 2, 3$$

Through these  $3N \times 3N$  rank 1 block matrices, by factorizing them using the SVD decomposition, we can extract the rows of all  $g_i R^i$  up to multiplication by some rotation matrix (initially an orthogonal matrix but this can be changed to an  $SO(3)$  matrix) which represents the degree of freedom in the orientation assignment problem, having all the rotations aligned with one another.

In this step we construct the matrices  $H_{\sigma, m}$  block by block, using the previous estimations calculated  $(\tilde{R}_{ij}, \tilde{R}_{ji})$ . Further details are discussed in the paper.

# Results for simulated data

The file 'pipeline\_T\_abinitio.py' demonstrates the flow of the algorithm for simulated data of the EMD-10835 with the options to set the simulation parameters:

img\_size – Desired resolution to downsample to.  
num\_imgs – Number of projection-images to use.  
noise\_variance – Control the noise added to the simulated projections.

The file 'pipeline\_O\_abinitio.py' behaves similarly for the 'O' symmetric molecule EMD-4905.

Below is an example of results from running the file 'pipeline\_T\_abinitio.py' on a simulation with 200 projection-images, a downsampled resolution of 189, a noise variance of  $5 \cdot 10^{-7}$  and induced CTF effect. As well as the results of running the file 'pipeline\_O\_abinitio.py' on a simulation with 150 projection-images, a downsampled resolution of 169, a noise variance of  $5 \cdot 10^{-7}$  and induced CTF effect.

We can observe the similarities between the reconstructed volumes as well as the similarities between the simulated downsampled projection images and the projections of the output volume on the estimated rotations which indicate accurate estimations of these rotations.,

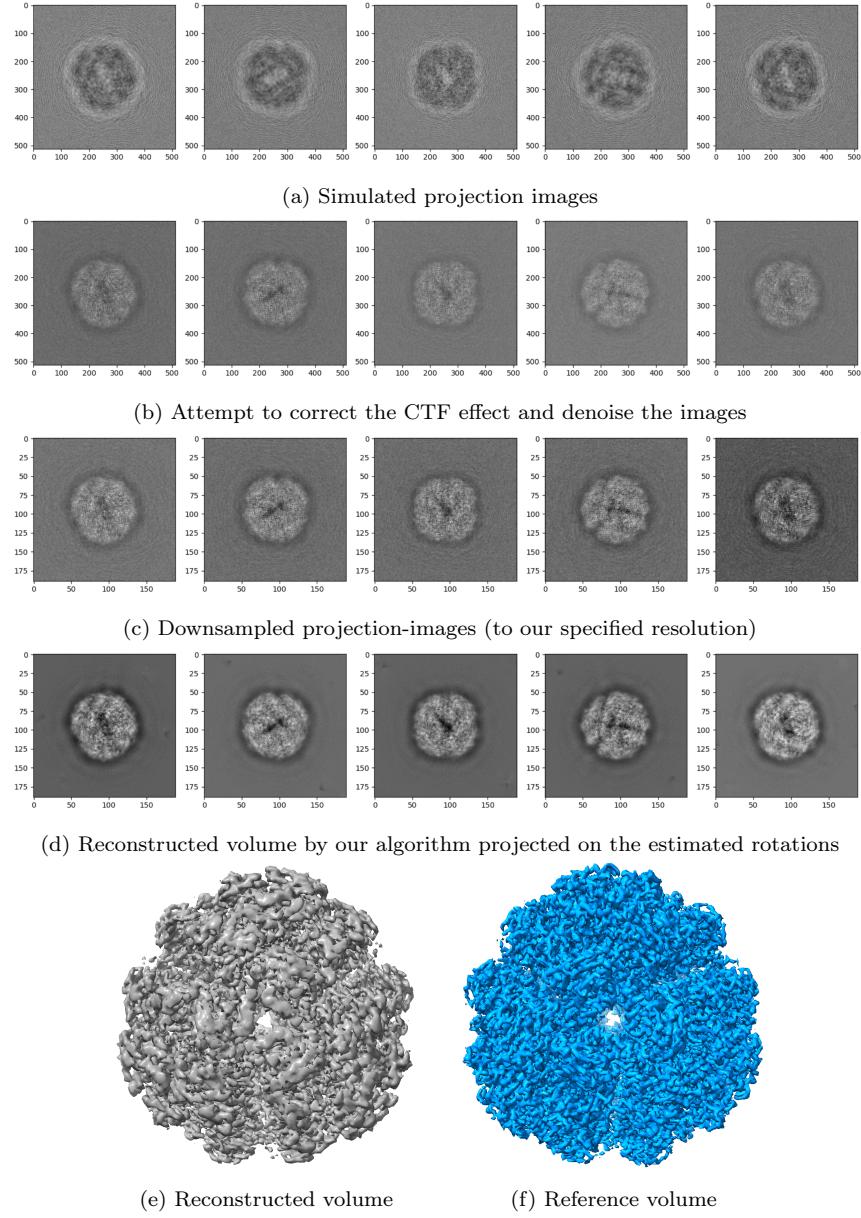


Figure 5: Algorithm pipeline results on the 'T' symmetric EMD-10835 using 200 simulated projections downsampled to resolution 189x189

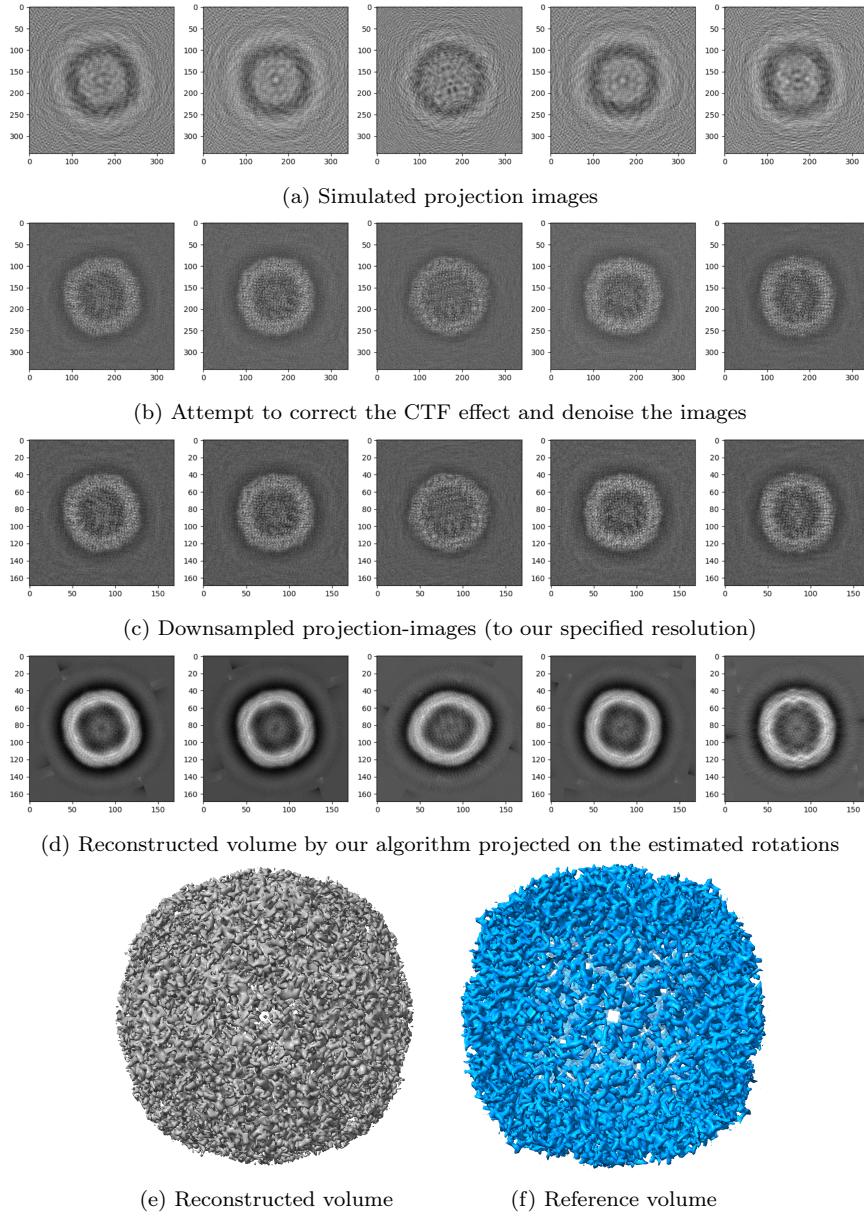


Figure 6: Algorithm pipeline results on the 'O' symmetric EMD-4905 using 150 simulated projections downsampled to resolution 169x169