

3D Reconstruction in Cryogenic Electron Microscopy with Translation and Rotational Variance

An Undergraduate Thesis Report 2 Submitted by

Maitrey Gramopadhye
Roll No. 160050049

Supervisor
Prof. Ajit Rajwade

Department of Computer Science and Technology
Indian Institute of Technology Bombay
Mumbai 400076 (India)

21 June 2020

Contents

1	Introduction	5
1.1	Tomography	5
1.1.1	Introduction	5
1.1.2	Radon Transform	5
1.1.3	Fourier Slice Theorem	6
1.1.4	Filtered Back Projection	7
1.1.5	Reconstruction with Compressed Sensing	8
1.1.6	Applications	8
1.2	Cryogenic Electron Microscopy	9
1.2.1	Tomography under unknown observation parameters	10
1.2.2	Moment-Based Approach	10
1.2.3	Ordering-Based Approach	11
1.3	Basic Cryo-EM Pipeline	12
1.3.1	Particle Picking	12
1.3.2	CTF Correction	13
1.3.3	Clustering	13
1.3.4	Angle Assignment	13
1.3.5	3D reconstruction	14
1.4	Literature Survey	14
1.4.1	3D structure determination from Common Lines - Amit Singer, Y. Shkolnisky	14
1.4.2	Khursheed Ali's M.Tech Project	15
1.4.3	RELION	16
1.4.4	CryoSPARC	17
2	Our Method	17
2.1	Introduction and Significance	17
2.2	Center of Mass Method	18
2.3	Stochastic batchwise Expectation Maximisation	21
2.4	Stochastic Average Gradient Descent	22
2.4.1	Stochastic Average Gradients	23
2.4.2	Importance Sampling	24
3	Compare and contrast with current packages	25
4	Experiments and Results	26
4.1	Datasets Used	26
4.1.1	Simulated Datasets	26
4.1.2	Real Datasets	29
4.2	Results	33

4.2.1	Simulated Datasets	33
4.2.2	Real Datasets	49
5	Future Work	53
5.1	CTF correction	53
5.2	Branch and Bound	53
5.3	Multiple Particle Case	55
6	Acknowledgements	55

Abstract

Cryogenic Electron Tomography (Cryo-EM) is an imaging technique used to produce high-resolution (1-4 nm) 3D views of samples. An electron microscope is used to record several 2D images of samples held at cryogenic temperatures. These 2D images are then aligned to yeild a 3D (tomographic) reconstruction of the sample.

While reconstructing structures of particles such as proteins or microbes, the electron microscopes are used to image slides prepared from the particle solutions at cryogenic temperatures. The output from the electron microscope is a large 2D image (micrograph) often containing images of several particles in various alignments, often in very low resolutions and having a poor signal to noise ratio. These particles are then marked and cropped out from the micrograph and denoised before reconstruction.

As the particles are randomly aligned in the micrographs their angles of projections are unknown. Also, while cropping out the particles from the micrograph, the particles need not be at the center of the crop, thus this introduces a translational shift in the projections. The projection angle (3 degrees of freedom) and shifts (2 degrees of freedom) need to be estimated first for a high quality 3D reconstruction.

We build upon the existing work for estimating projection angles and present a robust method to estimate the projection shifts. We divide the estimation of shifts in two steps, a course step and a fine tuning step. We present results on simulated and real data and propose ideas to improve them further.

1 Introduction

1.1 Tomography

1.1.1 Introduction

As [1], [2] have explained tomography is the process of reconstructing an image (object) from its projections in a lower dimension, usually 2D image from 1D projections or 3D image from 2D projections. In case of CT scans, it is also used to generate internal representations of the objects. A tomographic projection is defined as the radon transform of the image in a particular direction.

1.1.2 Radon Transform

As in [1], [3] radon transform can be defined as the projection of an image. Where every bin (or pixel) on the transform is the summation of image pixels lying on a line perpendicular to the projection and passing through the bin. Conversely, it can be thought as if a line was drawn through the image at some angle α and the image pixel values lying along the line were integrated to form the value of one bin. Every bin on the projection is formed by integrating along parallel lines with different offsets.

Mathematically, for a 2D image radon transform can be calculated as in eq. (1). Where R_θ is a radon transform of f along direction θ and ρ and δ is derac delta function. One projection is obtained by fixing θ and varying ρ which is the offset.

$$R_\theta(f) = g(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy \quad (1)$$

It can be observed that the radon transform would have one dimension less than the image. Fig. 1 shows a visual representation of how radon transforms of 2D images are taken.

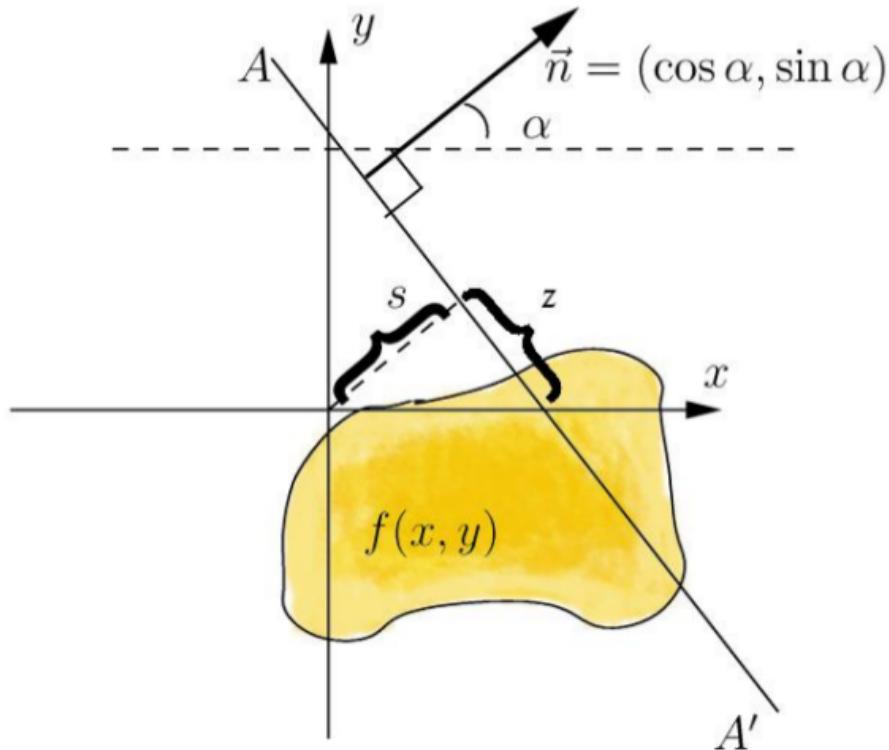


Figure 1: 2D to 1D Radon transform [3]

1.1.3 Fourier Slice Theorem

As stated in [1], [4] The Fourier slice theorem provides a very useful relation between the 2D Fourier transform of the original function and the 1D Fourier transform of its Radon transform. The theorem states that the Fourier transform of a projection of the 2D object along some direction θ (i.e. $G(\mu, \theta)$) is equal to a slice of the 2D Fourier transform of the object along the same direction θ (in the frequency plane), passing through the origin.

If $g(\rho, \theta)$ is the radon transform of f mathematically it is proved in eq. (2).

$$\begin{aligned}
G(\mu, \theta) &= \int_{-\infty}^{\infty} g(\rho, \theta) \exp(-j2\pi\mu\rho) d\rho \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) \exp(-j2\pi\mu\rho) dx dy d\rho \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \left[\int_{-\infty}^{\infty} \delta(x \cos \theta + y \sin \theta - \rho) \exp(-j2\pi\mu\rho) d\rho \right] dx dy \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-j2\pi\mu(x \cos \theta + y \sin \theta)) dx dy \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-j2\pi(xu + yv)) dx dy \quad (2)
\end{aligned}$$

Where $u = \mu \cos \theta$ and $v = \mu \sin \theta$

$$\therefore G(\mu, \theta) = [F(u, v)]_{u=\mu \cos \theta, v=\mu \sin \theta} = F(\mu \cos \theta, \mu \sin \theta) \quad (3)$$

The RHS of eq. (3) represents a slice of the 2D Fourier transform of $f(x, y)$, i.e. $F(u, v)$, along the angle θ in the frequency plane, and passing through the origin.

The fourier slice provides us a very easy way for tomographic reconstruction of 2D images known as the Filtered Back Projection.

1.1.4 Filtered Back Projection

Given tomographic projections we hope to reconstruct the original image by performing back projection along all the directions of projections and adding up the results. One method for that is the filtered back projection. [1]

Using radon transform and the fourier slice theorem we can reconstruct the 2D image by using the filtered back projection. Consider the image $f(x, y)$, its fourier transform $F(u, v)$, the projection $g(\rho, \theta)$ and its fourier transform $G(\mu, \theta)$.

By fourier slice theorem we have eq. (4).

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta \quad (4)$$

After simplifying eq. (4) can be written as eq. (5).

$$f(x, y) = \int_0^{\pi} \int_{-\infty}^{\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) |\mu| d\mu d\theta \quad (5)$$

In eq. (5), the inner integral is the 1D inverse fourier transform with an added term $|\mu|$. But this function is not integrable as $|\mu|$ grows unboundedly. To solve this problem, instead of $|\mu|$, the Ram-Lak filter [1] is used which is a clipped ramp filter.

Thus, using Ram-Lak filter Filtered Back Projection is defined as in eq. (6).

$$f(x, y) = \int_0^\pi \int_{-\infty}^{\infty} |\mu| rect(\mu D) G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) d\mu d\theta \quad (6)$$

1.1.5 Reconstruction with Compressed Sensing

Another method for tomographic reconstruction is reconstruction using compressed sensing (CS). [1] In most tomographic applications the number of angles of projection is limited due to cost, energy and health considerations. So the problem can be consider a type of "angle starved" case. Hence, tomography can be considered a CS problem. CS based tomographic reconstruction performs better than filtered back projection when the number of angles is less.

CS exploits one important property of images - their sparsity or compressibility in standard bases such as DCT. The cost function for CS based tomographic reconstruction is given in eq. (7). This problem can be solved using CS optimization algorithms like ISTA.

$$E(\beta) = \|y - RU\beta\|^2 + \lambda \|\beta\|_1 \quad (7)$$

$$f = U\beta \quad (8)$$

Where y vector is the concatenation of all tomographic projections. U is the basis matrix in which the image f is sparse. β is the representation of the image in U basis. R is the radon transform martix/operator.

1.1.6 Applications

The applications of tomography are in medical imaging such as CT scans as shown in fig. 2.[5] It is used in industrial applications, such as fault detection in machines or fault detection in buildings while construction.

It also has sensing applications, like observation of plant roots or remote sensing of underground objects or phenomenon.

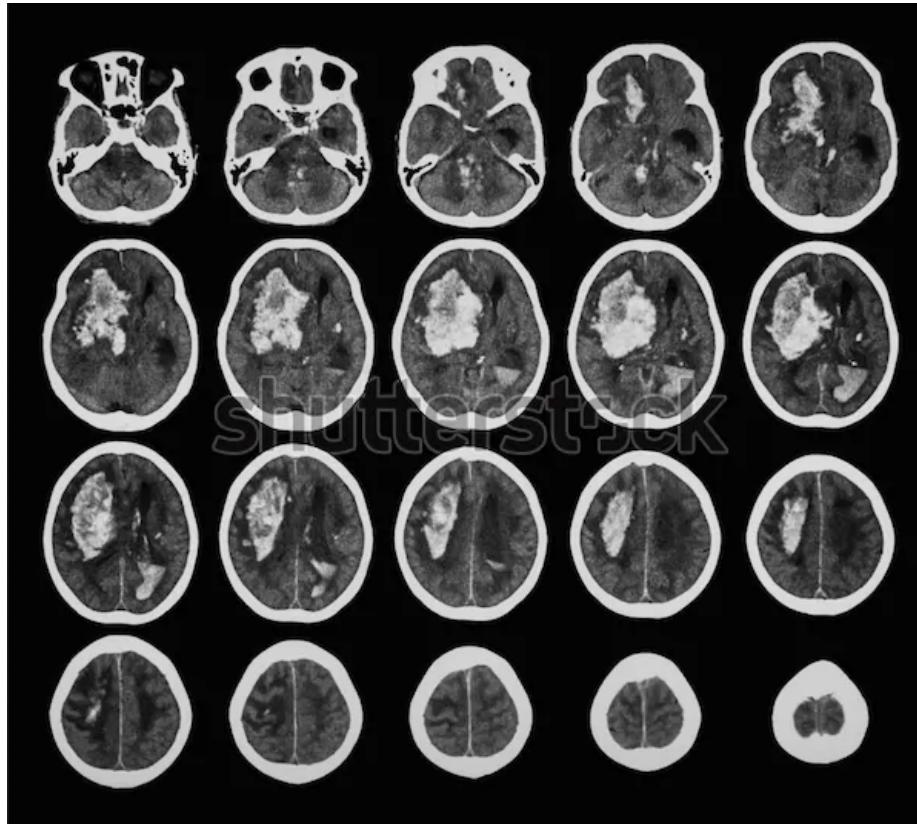


Figure 2: CT scans of brain *Source: shutterstock*

1.2 Cryogenic Electron Microscopy

As given in [1], [6] Cryogenic Electron Microscopy, or Cryo-EM, is used for imaging and reconstructing the structure of microscopic particles, such as virus, bacteria or proteins. Instead of the CT scans which uses X-rays and deals with imaging macroscopic particles, Cryo-EM uses an electron beam for capturing projections. Thus, the resolution of Cryo-EM projections is much better. However, in Cryo-EM the particles being observed are fragile and thus could be destroyed if a high intensity electron beam is used, which puts a limit on the resolution that can be achieved.

The intensity of the electron beam is adjusted according to the particle to get maximum possible resolution in the projections. For preparing the particles to be imaged, a slice of the sample is taken and frozen in vitrified water (water mixed with ethane) to a temperature of -150°C . Cryo-EM gets its name because the process is carried out in these cryogenic conditions and an electron microscope is used for imaging.

While imaging a slide of the sample is used, which contains several of

the particles of interest. These particles can be aligned randomly and thus in Cryo-EM the angle at which projections are taken is unknown. The problem is made tougher due to the presence of high levels of noise in projections, noise levels often reaching 100-200% of the signal. This is due to the limit on the maximum intensity of the electron beam that can be used.

1.2.1 Tomography under unknown observation parameters

In certain applications it is infeasible to know the angles at which the tomographic projections are taken. These cases fall under the case of Tomography under unknown observation parameters. In these cases, certain methods are used to estimate the angles before the reconstruction step. Some examples of said applications are as follows.

1. Patient motion during CT scanning
2. Moving insect tomography
3. Cryo-electron tomography

1.2.2 Moment-Based Approach

For understanding, consider the case of 2D images and 1D tomographic projections although the theory is extensible to 3D images (and their 2D projections).

The moment of order (p, q) of an image $f(x, y)$ is defined in eq. (9).

$$M_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) x^p y^q dx dy \quad (9)$$

The order n moment of a tomographic projection at angle θ is defined in eq. (10).

$$M_{\theta}^{(n)} = \int_{-\infty}^{\infty} P_{\theta}(s) s^n ds \quad (10)$$

Where $P_{\theta}(s)$ is the projection at angle θ .

$$P_{\theta}(s) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - s) dx dy \quad (11)$$

By substituting the value of $P_{\theta}(s)$ in the eq. (11) and simplifying, we get eq. (12).

$$M_{\theta}^{(n)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) (x \cos \theta + y \sin \theta)^n dx dy \quad (12)$$

The Helgason-Ludwig consistency conditions (HLCC), given in [1] give relations between image and projection moments, eq. (13).

$$M_{\theta}^{(n)} = \sum_{l=0}^n f(x, y) C_n^{n-l} (\cos \theta)^{n-l} (\sin \theta)^l M_{n-l, l} \quad (13)$$

Since we know the projections of $f(x, y)$, we know its moments $M_{\theta}^{(n)}$. In the above equation, the angles can be computed by iteratively solving for angles. Then the earlier mentioned methods could be used for reconstructing the image.

By this method, in case of order- n moments $M_{\theta}^{(n)}$. The original image can be exactly reconstructed, if number of unique angles of projections is at least $n + 1$, accurate upto a global rotation ambiguity. This method is easy to implement, however it is very sensitive to the noise in projections.

1.2.3 Ordering-Based Approach

In this approach, explained in [1], the tomographic projections are “sorted” - i.e. arranged in order of increasing angles. The method assumes availability of a large number of tomographic projections - under unknown angles - but sampled independently from a uniform distribution on a unit circle. The problem essentially reduces to projection association - i.e. matching each projection to one of the angles, sampled evenly from the unit circle.

It is solved iteratively, by first selecting a projection and assigning it a random angle. Then selecting the closest projection to it (based on some simple heuristic such as nearest neighbour search based on L2 distance) and assigning it the next angle. The process is repeated till a full mapping is completed.

An extra step can also be used in Ordering-Based approach to reduce the computational complexity of the algorithm. Instead of doing pairwise searches, a dimensionality reduction step is applied to the projections.

The projections are projected onto lower dimensional spaces in a manner that keeps their relative position (based on the heuristic chosen) the same, i.e. the projections far away from each other are still away in the lower dimensional space. The lower dimensional projections are aligned and angles are assigned to them. The original projections are also assigned the same angles and reconstruction is done by any of the earlier mentioned methods.

1.3 Basic Cryo-EM Pipeline

Detailed below is the entire Cryo-EM pipeline. We follow a pipeline similar to the one in [9]

1.3.1 Particle Picking

In Cryo-EM, we get huge micrographs of size for example 7000x7000. A few real datasets are [7], [8]. This micrograph contains many particles of interest, as shown for example in fig. 3, located at random positions and at random orientations. Thus, they need to be cropped out before the 3D structure of the particle can be estimated.

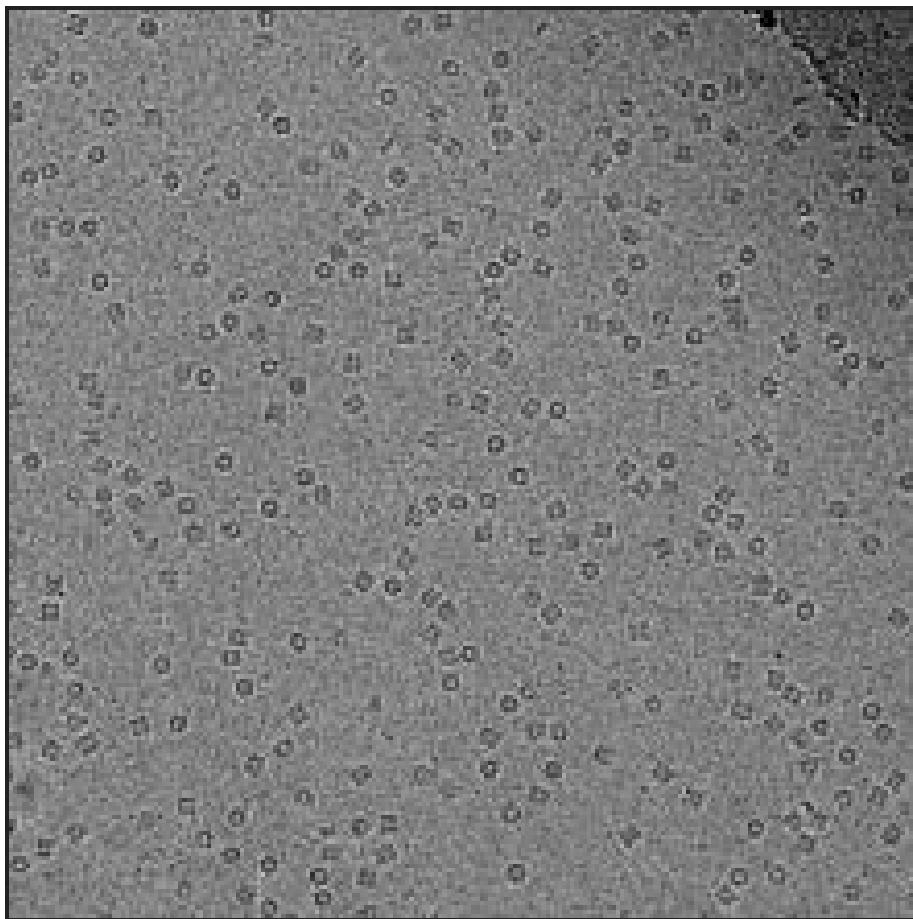


Figure 3: Cryo-EM Micrograph of GroEL [8]

Manual picking is one method to do this, where the biologists mark out the particles themselves or crowd source the task. Certain supervised and

semi-supervised methods also exist that mark out the particles. More on this is given in Section 2.2.1 of [9]

1.3.2 CTF Correction

Image formation in an electron microscope for biological samples taken under cryogenic environment by freezing the sample in vitrified water can mathematically be described as Contrast Transfer Function (CTF). CTF depends on many parameters, changing some of which improve resolution but changes the shape or other properties of particles. If we know the relations between the changes in these parameters, a CTF correction step can be applied to estimate the correct structure of the particles. More on this is given in Section 2.2.2 of [9]

1.3.3 Clustering

Since the levels of noise in Cryo-EM micrographs are high. In some algorithms, similar particle projections are clustered and averaged to reduce noise. Since an abundance of projections is available in Cryo-EM, it doesn't adversely affect the quality of reconstruction. The clustering is done in a way that projections with similar angles of projection lie in the same cluster. This can be done using simple heuristics like nearest neighbours based on L2 difference. More on this is given in Section 2.2.3 of [9]

1.3.4 Angle Assignment

As tomographic reconstruction requires angles at which projections were taken, an angle assignment step is performed using methods like moment-based or ordering-based approach described earlier.

However, these methods assume that the number of projections are uniformly distributed over the entire angle space. In case of certain particles, due to their structure, it is possible that their alignment in the slide is not random, this gives rise to a distribution of angles that is not uniform. In such cases multiple micrographs are collected by tilting the slides to collect a more uniform distribution of angles of projection. However, this makes the task of CTF correction more difficult. More on this is given in Section 2.2.4 of [9]

1.3.5 3D reconstruction

After assigning angles, the 3D structure is reconstructed by methods like FBP or CS described above. More on this is given in Section 2.2.5 of [9]

Since Cryo-EM micrographs have high levels of noise, the best results are not obtained by the methods described so far. Algorithms that achieve better results are described below.

1.4 Literature Survey

1.4.1 3D structure determination from Common Lines - Amit Singer, Y. Shkolnisky

The algorithms for angle estimation discussed so far rely on distance between the projections to sort them and assign them an angle of projection. However, in case of Cryo-EM the number of projections is very large and the SNR is low in projections. Due to this the angle assignment process becomes computationally heavy and error prone.

An extra information present in case of 3D images and 2D projections is the presence of common lines in projections. The common lines can be used to estimate the relative angle between projections. Thus giving at least a good initialization to the projection angles.

Per Fourier slice theorem, 2D Fourier transform of a 2D projection in direction d is equal to a slice through the 3D Fourier transform of f , that passes through the origin. Consider two projections of f . Both their fourier transforms will be equal to a centeral slice through the 3D Fourier transform of f . If these slices are non-coplannar, they will intersect in a line passing through the origin. This line is called the common line.

Consider two projections at angles R_i and R_j intersecting in a common line. Let c_{ij} be defined as the angle made by the common line in the local coordinate space of P_i (Projection i). Similarly for P_j . Being the common line of intersection, the mapping of c_{ij} by R_i must coincide with

the mapping of c_{ji} by R_j , given by eq. (14)

$$\begin{aligned} R_i c_{ij} &= R_j c_{ji} \\ \text{Where,} \\ c_{ij} &= (x_{i,j}, y_{i,j}, 0)^T \\ c_{ji} &= (x_{j,i}, y_{j,i}, 0)^T \end{aligned} \tag{14}$$

Unfortunately due to high noise value the exact equivalence is infeasible. So eq. (15) is used instead.

$$\max_{R_1, \dots, R_n} \sum_{i \neq j} R_i c_{ij} \cdot R_j c_{ji} \tag{15}$$

Constrained by eq. (16) to remove the trivial solution.

$$R_i R_i^T = I, \text{ for all } i \tag{16}$$

Defining S as eq. (17)

$$\begin{aligned} S &= \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \\ \text{where,} \\ S_{11} &= x_{ij} x_{ji} \\ S_{12} &= x_{ij} y_{ji} \\ S_{21} &= y_{ij} x_{ji} \\ S_{22} &= y_{ij} y_{ji} \end{aligned} \tag{17}$$

It can be shown that the solution for the direction of the common line corresponds to the 3 largest eigen values of S in eq. (17). This gives us a good initial estimate for the relative angle between the projections. More detailed description of the method is in [11]

1.4.2 Khursheed Ali's M.Tech Project

The work we propose is built upon [9]. This M.Tech project gives a supervised algorithm for particle picking from micrographs. It then goes on to describe a method to reconstruct the 3D structure of the particles.

First the initial estimates of the angles of projections are computed from the common lines in projections as described in the earlier section. Then after initializing the angles the angles of projections and 3D reconstruction are both iteratively optimized. The iterative optimisation tries to optimise using eq. (18) in every iteration.

$$\begin{aligned}
(\theta_i^1, \theta_i^2, \theta_i^3, G) = \operatorname{argmax}_{\theta_i^1, \theta_i^2, \theta_i^3, G} NCor(p_i^{true}, \operatorname{Radon}(G, [\alpha + \theta_i^1, \beta + \theta_i^2, \gamma + \theta_i^3])) \\
\text{where,} \\
- k \leq \theta^1 \leq k \\
- k \leq \theta^2 \leq k \\
- k \leq \theta^3 \leq k \\
\forall i = 1, 2, \dots, N \\
\alpha : \text{angle with } X \text{ axis} \\
\beta : \text{angle with } Y \text{ axis} \\
\gamma : \text{angle with } Z \text{ axis} \\
k : \text{angle search area parameter} \\
N : \text{number of projections}
\end{aligned} \tag{18}$$

$NCor(x, y)$ will finds out the normalized correlation between the image x and y . $\operatorname{Radon}(G, ang)$ will find out the 3D radon transform of the 3D object G at the given angle ang .

In the iterative loop, the 3D reconstruction is computed with the current angle estimation. Then the projections of this reconstruction are taken with the same angles of projection, but with some jitter added. Then these new projections are compared with the original projections to generate a normalised correlation value. The angle estimate is updated to the angle of the new projection with most normalised correlation. Several values of jitter are used in each iterative step and the projection angle estimations are updated to the (projection angles + jitter) value that gave the best correlation value. This loop is run till convergence.

1.4.3 RELION

RELION, for REgularized LIkelihood OptimizatioN, (described in detail in [12]) is an open-source computer program for the refinement of macromolecular structures by single-particle analysis of electron cryo-microscopy (cryo-EM) data. Whereas alternative approaches often rely on user expertise for the tuning of parameters, RELION uses a Bayesian approach to infer parameters of a statistical model from the data.

RELION proposed good methods that reduce the computational costs for Cryo-EM reconstruction and gave new insights into the accuracy with

which the relative orientations of individual particles may be iteratively optimised.

1.4.4 CryoSPARC

Cryo-EM single-particle ab initio reconstruction and classification (cryoSPARC) [13] uses a stochastic gradient descent (SGD) approach to iteratively refine the final 3D reconstruction, instead of optimising the orientation and translation parameters. After the SGD has terminated, they also apply a branch-and-bound maximum likelihood optimization to refine the structure to high resolution.

Furthermore, SGD with Bayesian marginalization allows ab initio (random initialisation) 3D reconstruction.

2 Our Method

2.1 Introduction and Significance

The work we propose builds upon the M.Tech work of Khursheed Ali. We propose improvements by the estimation and correction of translation error along with the estimation of angles of projections. We also improve upon the completion time of the algorithm, while maintaining comparable results, by employing stochasticity and gradient based approaches in the iterative optimization loop.

The input for Cryo-EM reconstruction is a large micrograph with many projections of the particles of interest. A particle picking step is necessary before proceeding with the 3D reconstruction. Now, during the particle picking step a single projection is extracted by enclosing it in a 2D bounding box. Irrespective of the method used for constructing this box, whether manual picking, supervised or semi-supervised algorithm, the particle enclosed need not be at the center of the box. This gives rise to a shift or translation error. We define this error as $E_t = (T_x, T_y)$. It arises due to imperfections in the algorithm used, human error etc and it is very infeasible to ensure $E_t = (0, 0)$.

Now, E_t poses a problem in 3D reconstruction step. As, 3D reconstruction works essentially by back-projecting the projections along their directions of projection and taking a sum over all back-projections. A key assumption here is that all the projections are aligned i.e all their centers

coincide. Also the same assumption is taken while computation common lines for projections. Here, if the centers of projections are not aligned, then for any two projections the common line won't pass through the center of both the projections, which is important for Fourier slice theorem.

In the works proposed in the literature survey, none initialise the projections to be aligned before their method is applied. [11] gives a good initial estimate for the angles of projection. However, their work relies on estimating common lines in projections and if the projections are misaligned then the majority of common lines detected will be false detections which will make the initial estimation useless.

This is the area where we mainly contribute by proposing a method to estimate and correct majority of the translation error before common line estimation. Our work divides the correction of translation error into two steps, a coarse estimation step and a fine tuning step. We present the complete algorithm for coarse estimation and fine tuning steps and give promising results on simulated and real data for the same.

The coarse step is performed using Center of Mass estimation of the projections. It is done before common line estimation and reduces the translation error to at max 4 pixels (tested in case of simulated data with gaussian noise with mean = 0 and sigma = 100% of mean of signal). The fine tuning step then iteratively refines the estimates to reduce translation error along with rotation error.

2.2 Center of Mass Method

This section describes in detail the Center of Mass Method method for coarse Estimation and Correction of Translation error. We first prove that the center of mass of the image should be projected to the center of mass of the projection and then describe our method.

For simplicity assume the case of 2D image $f(x, y)$ with a 1D radon projection $g_\theta(\rho)$ at angle θ . Consider the center of mass of f be (\bar{x}, \bar{y}) and of g be $\bar{\rho}$. By the equation for center of mass we have eq. (19), eq. (20), eq. (21) on \bar{x} , \bar{y} , and $\bar{\rho}$ respectively.

$$\bar{x} = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x f(x, y) dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy} \quad (19)$$

$$\bar{y} = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y f(x, y) dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy} \quad (20)$$

$$\bar{\rho} = \frac{\int_{-\infty}^{\infty} \rho g_{\theta}(\rho) d\rho}{\int_{-\infty}^{\infty} g_{\theta}(\rho) d\rho} \quad (21)$$

From eq. (1) of radon transform we have eq. (22)

$$\begin{aligned} g_{\theta}(\rho) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy \\ \int_{-\infty}^{\infty} \rho g_{\theta}(\rho) d\rho &= \int_{-\infty}^{\infty} \rho \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy d\rho \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \int_{-\infty}^{\infty} \rho \delta(x \cos \theta + y \sin \theta - \rho) d\rho dx dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) (x \cos \theta + y \sin \theta) dx dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y \sin \theta f(x, y) dx dy + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x \cos \theta f(x, y) dx dy \end{aligned} \quad (22)$$

Similarly, from eq. (1) we have eq. (23)

$$\begin{aligned} \int_{-\infty}^{\infty} g_{\theta}(\rho) d\rho &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy d\rho \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy \end{aligned} \quad (23)$$

Therefore,

$$\begin{aligned} \bar{\rho} &= \frac{\int_{-\infty}^{\infty} \rho g_{\theta}(\rho) d\rho}{\int_{-\infty}^{\infty} g_{\theta}(\rho) d\rho} \\ &= \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y \sin \theta f(x, y) dx dy + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x \cos \theta f(x, y) dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy} \\ \bar{\rho} &= \bar{x} \cos \theta + \bar{y} \sin \theta \end{aligned} \quad (24)$$

Therefore, from eq. (24) we can say that $\bar{\rho}$ lies on the line $\bar{x} \cos \theta + \bar{y} \sin \theta - \bar{\rho} = 0$, i.e (\bar{x}, \bar{y}) and $\bar{\rho}$ lie on the same line and the line is in the direction of projection. Therefore, (\bar{x}, \bar{y}) is projected on $\bar{\rho}$ for any θ . This can be generalised for any dimensions.

In our Center of Mass method, we shift the center of mass of every projection to the center of the projection to form shifted projections. We

define the center of projection as the center of the projection image, i.e if the projection is of size $N \times N$ the coordinates of the center of projection would be $(N/2, N/2)$ in the projection image. In this way, theoretically, the center of the image is now projected to the center of the shifted projections. Thus, the shifted projections are aligned with $E_t = 0$. However this is not the case due to the presence of noise. Never the less, this method provides a good initial alignment for the projections.

For every projection we compute it's center of mass and shift the projection such that the center of mass lies at the center of the projection.

```

function [ rProj , shifts ] = rmvTranslationWithCOG( projections )
    [ l,m,n]=size( projections );
    rProj=zeros( l,m,n );
    shifts=zeros( 2 ,n );
    for i=1:n
        p=projections (:,:,i );
        [ sumImg_x ,sumImg_y ,sumImg_0]=centroid_est(p);

        x_ = sumImg_x/sumImg_0;
        y_ = sumImg_y/sumImg_0;

        shift_x=round( x_-(m+1)/2);
        shift_y=round( y_-(l+1)/2);
        shifts (1,i)=shift_x ; shifts (2,i)=shift_y ;
        [P] = translations(p,[ - shift_x -shift_y ]);
        rProj (:,:,i)=P;
    end
end

function [ sum_x,sum_y,sum_0 ] = centroid_est(img)
    %% Find a centroid of a 2d projection
    [a,b] = size(img);
    x = 1:a;
    y = 1:b;

    [X , Y] = meshgrid(x,y);

    sum_0 = sum(double(img) , 'all ');
    sum_x = sum(( double(img) .* X ) , 'all ');
    sum_y = sum(( double(img) .* Y ) , 'all ');
end

```

```

function [img] = translations(p, shift)
    P = circshift(p, shift(1), 2);
    img = circshift(P, shift(2), 1);
end

```

2.3 Stochastic batchwise Expectation Maximisation

One of the methods we propose for iterative refinement of the orientations and translations (henceforth referred as parameters) of projections is Stochastic batchwise Expectation Maximisation. This method draws motivation from RELION [12] and is an extension of the method used in [9].

We use the common line estimation approach from [11] to get an initial estimate of the parameters of projections and use it as a starting point for our refinement. As an input, we have an initial estimate of the 3D structure and initial orientations for all picked particles (original projections); we initialise all the shifts to $(0,0)$, as the center of mass method performs a coarse translation correction on the original projections.

In each iteration of the Stochastic batchwise Expectation Maximisation algorithm, we randomly (uniformly) select a batch of projections. For each projection in the batch, we randomly select a subset of orientations and a subset of translations. We select the parameters from a distribution, defining how relevant each parameter is to the 3D reconstruction.

We take projections of the current estimate of 3D structure for all the parameters in this subset and compare these projections to the picked particles from the micrograph. We use minimum L2 error with the picked particles to estimate the best possible parameters in this subset and assign them as the current orientation and shift for the projection. To ensure that the estimates of the parameters don't get worse in any iteration, we also include the current estimates in the subsets.

To define the relevance of an orientation or translation for a projection we maintain a distribution on them. For the orientations for each projection, we use an uniform distribution, on the orientations in the vicinity of the current orientation. With each iteration we reduce the size of this vicinity, thus making the orientations close to current orientation more relevant as we improve the 3D reconstruction. For the shifts, we use a Gaussian centered at the current shift value, and in every iteration we reduce the

variance of this gaussian, making the shifts closer to the current shift more relevant as we improve the 3D structure.

By taking the projections and parameters in batches and not considering all the possibilities, we exploit the redundancy in the data and see a great improvement upon the execution time. As the projections are selected uniformly randomly, we assume that after several iterations all of them would be considered, while having the advantage of updating the 3D reconstruction more frequently. Also, by defining and updating distributions on the orientations and shifts, we avoid considering many irrelevant options, thus improving the execution time further.

2.4 Stochastic Average Gradient Descent

In Stochastic Average Gradient Descent, we propose iteratively refining the 3D structure itself, instead of refining the orientations and shifts of projections. This method draws motivation from [13] and [14].

We initialise the 3D structure with the common line estimation approach from [11] and iteratively update it to minimise the posterior probability given by eq. (25).

$$p(V|\theta) \propto p(V) \prod_{i=1}^K p(I_i|\theta_i) \quad (25)$$

Where,

$$p(I_i|\theta_i) = \sum_{j=1}^{M_r} \sum_{k=1}^{M_t} w^R w^t p(I_i|\theta_i, R_j, t_k) p(R_j) p(t_k) \quad (26)$$

Where V is the 3D structure, I_i is a projection, θ is the CTF parameter, $p(V)$ is a prior over 3D molecular densities and K is the number of picked particles. As suggested by [14], we have taken the prior to be $p(V) = \prod_{i=1}^{D^3} \lambda e^{-\lambda V_i}$ where V_i is the density of the i th voxel, D is the size of the picked particle images and λ is an adjustable constant.

M_r is the total orientations possible and M_t is the total shifts possible. R_j is a possible orientation, i.e it is a rotation/reflection matrix of size 3x3. R_j can be completely characterised by an axis of rotation, which is its direction of tomographic projection and an angle which denotes its rotation in the plane perpendicular to this axis. t_k is a possible shift, i.e it denotes

the shift of the center of the particle from the center of the projection image, in the 2D projection plane.

We assume that the initial distribution on orientation parameters is uniform and on shift parameters is a gaussian. Thus, $w^R = M_r^{-1}$ and w^t is a gaussian centered at (0,0) with a variance of 3. As described in section 2.1, the maximum translation error we saw on simulated data after the coarse shift correction step was 4 pixels from (0,0). Thus, we have kept the variance 3 making an assumption that the translation error will be small.

Optimising the structure corresponds to finding V that maximizes the posterior. We choose our objective function as in eq. (27)

$$f(V) = -\log p(V) - \sum_{i=1}^K \log p(I_i | \theta_i) \quad (27)$$

However, optimising this function directly is costly due to the large number of particles and large number of possible parameters. Therefore, we employ stochastic average gradients and importance sampling.

We calculate $p(I_i | \theta_i, R_j, t_k)$ by calculating the L2 error, eq. (28), between the projection of the current estimate of the 3D structure, with the parameters R_j and t_k , and the original picked particle I_i

$$\begin{aligned} p(I | \theta, r, t) &= \frac{1}{Z} \exp\left(\sum_l \frac{-1}{2\sigma^2} |Y_l(r) - S_l(t)I_l|^2\right) \\ Y_l(r) &= \Theta_l(r)V \end{aligned} \quad (28)$$

Where l represents a pixel in the projection I , σ is the noise standard deviation, $S_l(t)I_l$ is the l^{th} pixel of projection I after applying a shift of t , $Y_l(r)$ is the projection of V at orientation r and Z is a normalisation constant.

2.4.1 Stochastic Average Gradients

SAGD explicitly produces an estimate of the full gradient over the entire dataset as described in [14]. Our goal, as stated above is to minimize the negative log posterior given by eq. (29).

$$f(V) = \sum_{i=1}^K \left[-\frac{1}{K} \log p(V) - \log p(I_i | \theta_i) \right] \quad (29)$$

$$f(V) = \sum_{i=1}^K [f_i(V)] \quad (30)$$

At each iteration τ , SAGD selects a random particle image, indexed by i_τ , the corresponding objective for which is the log likelihood, denoted, $f_{i_\tau}(V)$. Also, let the gradient of the objective with respect to the 3D structure be $g_{i_\tau}(V) \equiv \nabla_V f_{i_\tau}(V)$. SAGD then computes an update given by eq. (31).

$$V_{\tau+1} = V_\tau - \frac{\epsilon}{L} \sum_{i=1}^K [dV_i^\tau] \quad (31)$$

where ϵ is a base learning rate, L is a Lipschitz constant of the gradient $g_{i_\tau}(V)$ and dV_i^τ is given by eq. (32).

$$\begin{aligned} dV_i^\tau &= g_{i_\tau}(V) \text{ if } i == i_\tau \\ &= dV_i^{\tau-1} \text{ otherwise} \end{aligned} \quad (32)$$

Further, rather than selecting a single data point at each iteration, we select a subset of data points (batches) and compute the gradient for the sum of the objective over the entire batch.

L is estimated using a line search algorithm where an initial value of L is increased until the instantiated Lipschitz condition $f(V) - f(V - L^{-1}dV) < \frac{\|dV\|^2}{2L}$ is met. In between line searches, L is also gradually decreased to try to take larger steps.

2.4.2 Importance Sampling

The cost of computing the required gradient remains high even after implementing a stochastic batch method over picked particles due to marginalization over 3D orientations and 2D shifts in eq. (26). Intuitively, one could consider randomly selecting a subset of the parameters and using them to calculate an approximation. We do just that and draw random subsets of particles from their importance distributions.

We select the subset of parameters from importance distributions defined on the parameter space. We initialise different distributions for the orientations and shifts of each projection as described in section 2.4. The initial distribution on orientation parameters, ψ_R , is uniform in the vicinity of the initial orientation, being zero everywhere else. The initial shift distribution, ψ_t , is a gaussian centered at (0,0) with a variance of 3.

We update the distributions for each projection by eq. (33).

$$q_j = (1 - \alpha)Z^{-1} \sum_{i \in \tau} \phi_i^{1/T} K_{i,j} + \alpha \psi \quad (33)$$

Where,

$$\begin{aligned} \phi_j^R &= \sum_{k \in \tau^t} \frac{w_k^t p(I_i | \theta_i, R_j, t_k)}{N_t q_k^t} \\ \phi_k^t &= \sum_{j \in \tau^R} \frac{w_j^R p(I_i | \theta_i, R_j, t_k)}{N_R q_j^R} \end{aligned}$$

Where, q_j is the importance distribution for the j^{th} parameter, q being a distribution on either orientations or translations. ψ is the initial distribution. α is the mixing parameter with the prior distribution and T is an annealing parameter. Kernel $K_{i,j}$ is used to diffuse probability around parameters as neighbouring parameters are more likely to be useful. ϕ_i can be thought as a random variable, that when selected according to the importance distribution q would be in expectation equal to the inner sum in the posterior. j spans over all parameter space and i spans over the subset of parameters selected in the iteration.

N_t is the total number of shifts possible and N_R is the total number of orientations possible. $K_{i,j}^R$ is proportional to $\cos(a_{i,j})$, where $a_{i,j}$ is the angle between orientation directions of R_i and R_j . $K_{i,j}^t$ is proportional to the distance between the shifts t_i and t_j .

We perform this update to the importance distribution in every iteration.

3 Compare and contrast with current packages

Even though we take motivation from earlier works, we suggest and have implemented various modifications to them.

Comparing our Stochastic batchwise Expectation Maximisation method with RELION [12], we take the motivation for initialising the orientations from them but go an extra step and use our Center of mass method to initialise the translations as well. RELION also proposes a method to select a few parameters instead of all of them to update the optimal parameter. However, our methods of selecting the parameters differ. RELION

proposes a hierarchical grid on the parameter space. A distribution on the parameters is first computed on the parameters using a coarse grid, and the parameter ranges with maximum values are selected. Then a finer grid is used on these ranges to select the parameters. This is done for both orientations and shifts, in each iteration. RELION doesn't maintain a distribution on the parameter space like we propose. Our method of maintaining a distribution produces reasonable results and also avoids heavy computation in every iteration, thus increasing efficiency.

For our Stochastic Average Gradient Descent, we draw motivation from CryoSPARC package [13]. However, there are several key differences in our implementation. The key difference in algorithm being that CryoSPARC uses SGD whereas we use SAGD, which is described in [14]. We maintain a gradient for all particles and not just the ones selected in the iteration, while maintaining similar time complexity. This helps us generalise over the data better and leads to faster reconstruction. Another difference is that we provide a decent initialisation as a starting point, whereas CryoSPARC uses a random initialisation. The motivation for a random initialisation can be understood as - without a prior bias the algorithm can detect other underlying structures in case of multi-conformer data. However, in our case of single particle reconstruction, providing an initialisation helps achieve better reconstruction time. We also propose importance distributions to better select relevant orientations and shifts, which is not included in the CryoSPARC paper. CryoSPARC however does propose an additional Branch and Bound algorithm to be applied after SGD converges to further increase resolution of the reconstruction.

4 Experiments and Results

4.1 Datasets Used

We use 2 types of datasets to test our algorithms. Simulated datasets and real datasets

4.1.1 Simulated Datasets

The simulated datasets used for our experiments are taken from the "Electronic Microscopic Data Bank" [15]. Our algorithm was tested on the EMDB-8647 (fig. 4) and EMDB-5689 (fig. 5).

EMD-8647 (fig. 4) is a ribosome from *Mycobacterium smegmatis*, i.e., the human pathogen *Mycobacterium tuberculosis*. It's a ribosome of voxel dimension $2.5 * 2.5 * 2.5 \text{ \AA}^\circ$ and map dimensions as $128 * 128 * 128$, but for experiments purpose, it's being downsampled by 2. So, throughout all the experiments, the map dimension will be $64 * 64 * 64$.

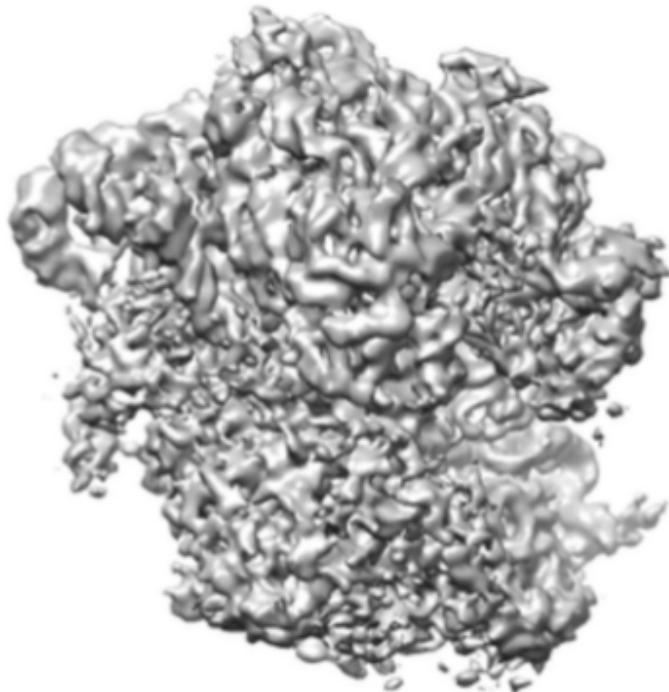


Figure 4: EMDB-8647 Source: [15]

EMD-5689 (fig. 5) is the Enterobacteria phage T7, a bacteriophage, i.e a virus composed of DNA. Its voxel dimension are $16 * 16 * 16 \text{ \AA}^\circ$ and map dimensions as $160 * 160 * 160$, but for experiments purpose, it's being downsampled by 2. So, throughout all the experiments, the map dimension will be $80 * 80 * 80$.

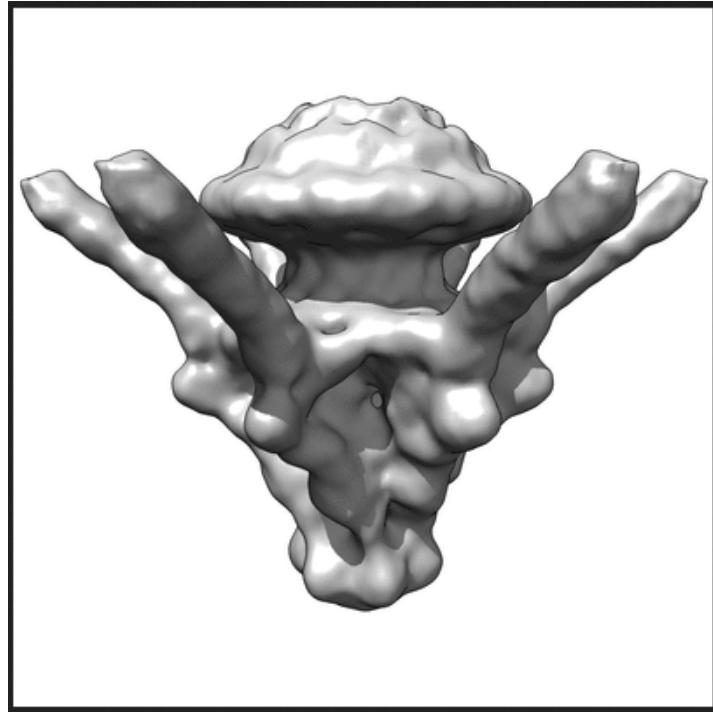


Figure 5: EMDB-5689 Source: [15]

We generate simulated data by taking 2D projections of these particles in random orientations and then adding gaussian noise and random shifts to them (fig. 6).

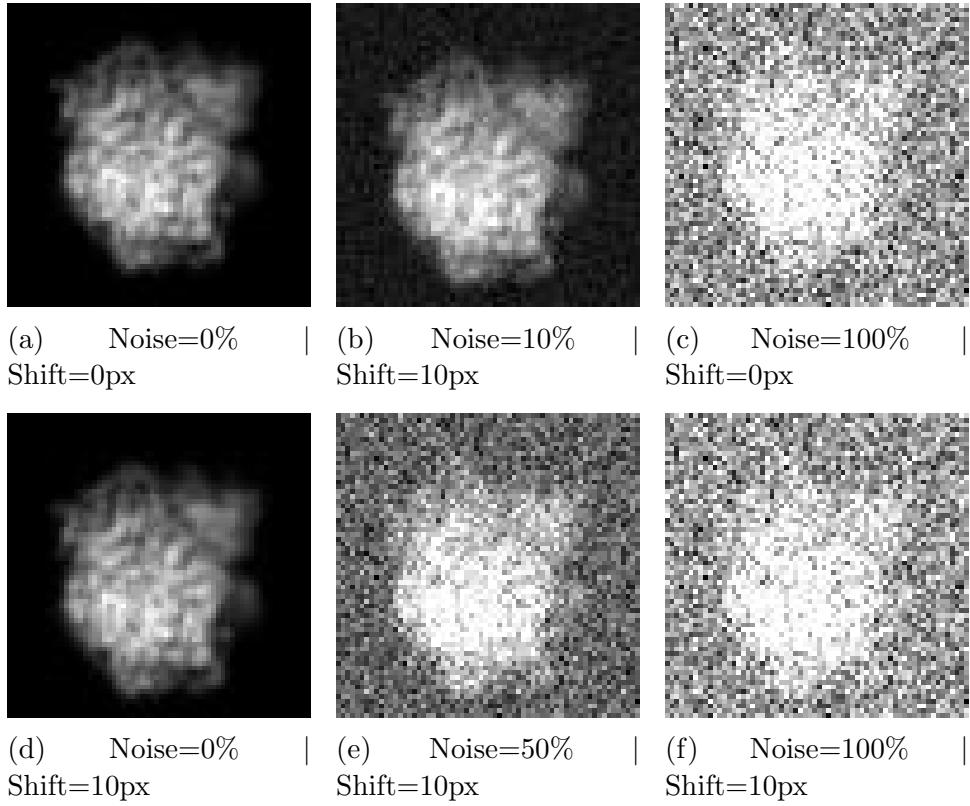


Figure 6: Projections of EMDDB-8647 [15] at varying levels of gaussian noise (in percentage - %) and shift (in pixels - px)

4.1.2 Real Datasets

The real datasets used for our experiments are taken from the "Electron Microscopy Public Image Archive (EMPIAR)" [16]. Our algorithm was tested on the EMPIAR-10295 [7] (fig. 4) and EMPIAR-10029 [8] (fig. 5).

EMPIAR-10295 (fig. 7) is the Single particle cryo-EM dataset of clathrin cages. Its map dimensions are $750 * 750 * 750$, but for experiments purpose, it's being downsampled by 10. So, throughout all the experiments, the map dimension will be $75 * 75 * 75$.



Figure 7: EMPIAR-10295 Source: [7]

EMPIAR-10029 (fig. 8) is a simulated cryoEM data set of GroEL particles. Its map dimensions are $200 * 200 * 200$, but for experiments purpose, it's being downsampled by 4. So, throughout all the experiments, the map dimension will be $50 * 50 * 50$.

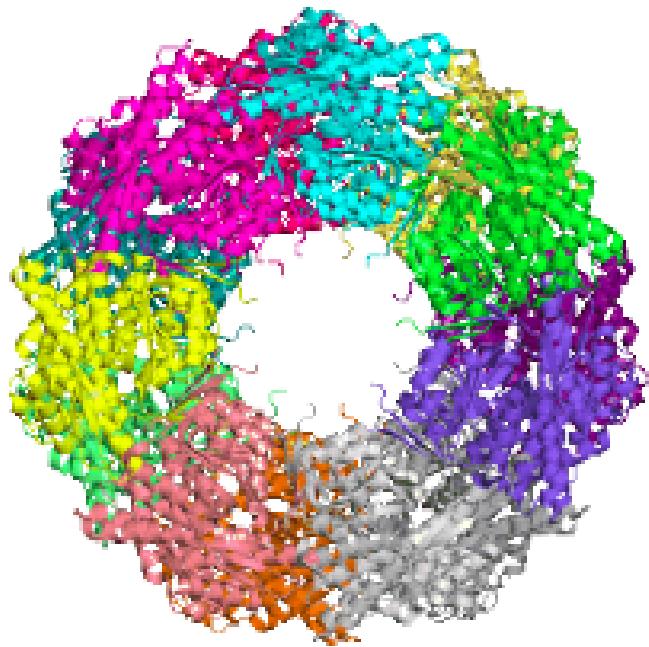


Figure 8: EMPIAR-10029 Source: [8]

From EMPIAR, we get stacks of picked particles for some 3D structures. The structures we perform experiments on have their picked particles available (fig. 9), (fig. 10). This allows us to skip the particle picking step. In a final package, a micrograph would be given as input and the particle picking step would need to be applied, [9] is a good implementation of a particle picker.

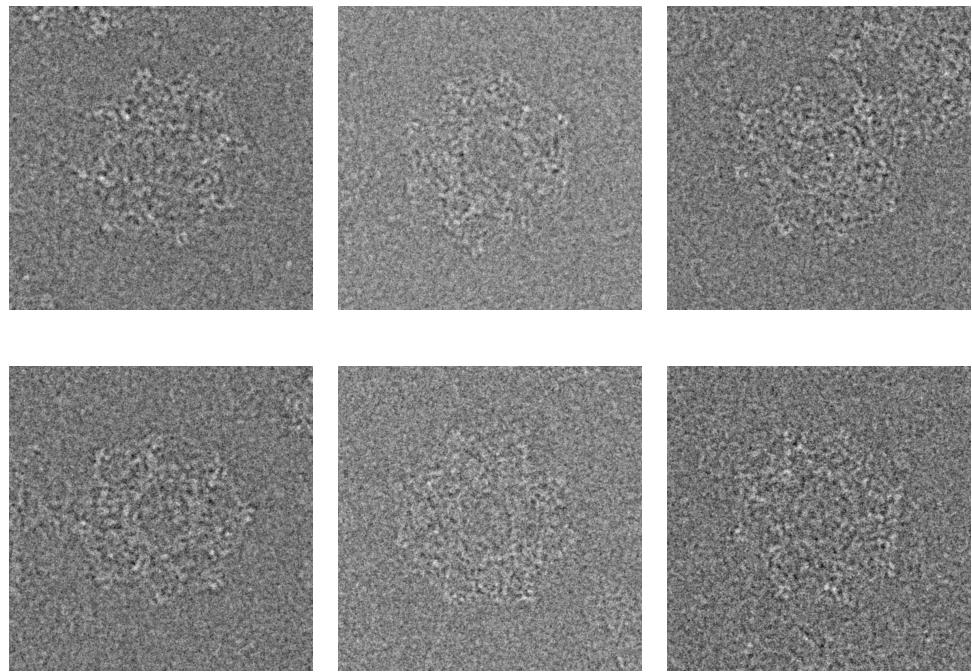


Figure 9: Picked Particles of EMPIAR-10295 [7]

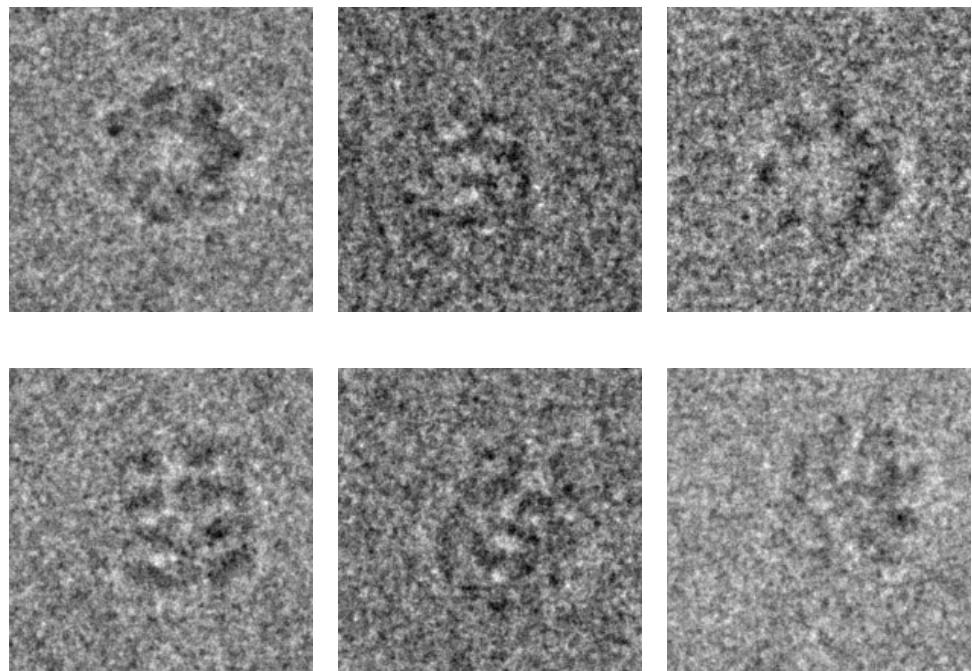


Figure 10: Picked Particles of EMPIAR-10029 [8]

4.2 Results

This section details the results from our experiments. We conduct experiments on simulated datasets and real datasets both. We present results obtained from the pipeline we described. We use our own method for coarse estimation (section 2.2) and correction of shifts before the iterative refinement. We describe results from both our proposed methods for iterative refinement - Stochastic batchwise Expectation Maximisation (section 2.3) and Stochastic Average Gradient Descent (section 2.4). All the other parameters used for reconstruction are presented with the results. The reconstruction videos can be found at <https://drive.google.com/open?id=1eGZZlh0Z1q1ERNyEI20y0eIX1pIBg659>.

4.2.1 Simulated Datasets

Simulated datasets [15].

Stochastic batchwise Expectation Maximisation

The cross-sections of reconstructed 3D structures are displayed below. In the results,

total projections = 2000

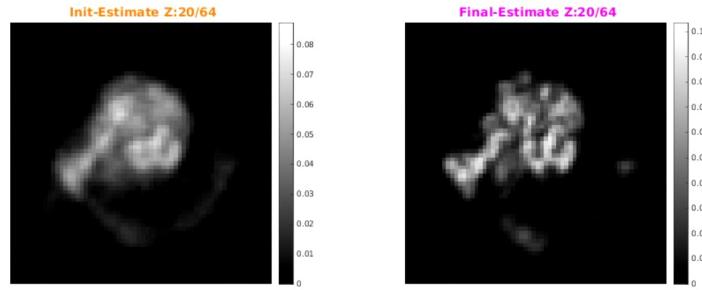
batch size of projection for each iteration = 200

max iterations = 50

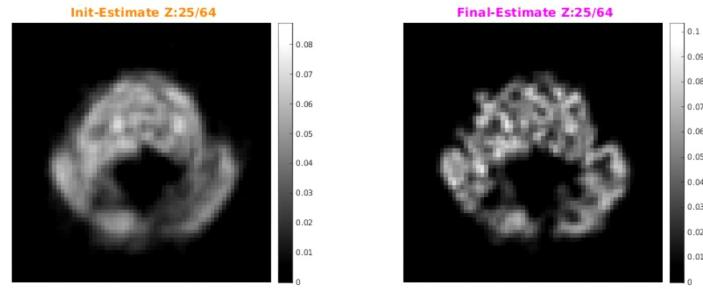
batch size of orientation parameters considered for each iteration = 1000

batch size of shift parameters considered for each iteration = 25

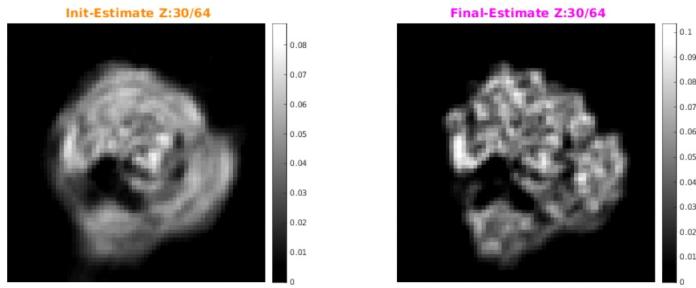
Compared to our previous report [10], these results are comparable in resolution but were generated much faster. In [10], results with similar resolution would take approximately 100 hours to generate, whereas this work generates them in 6-7 hours.



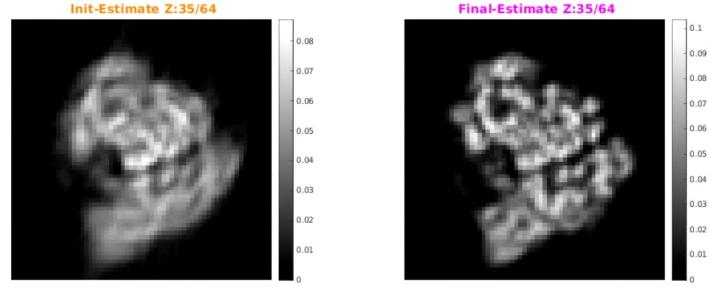
(a) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=20/64



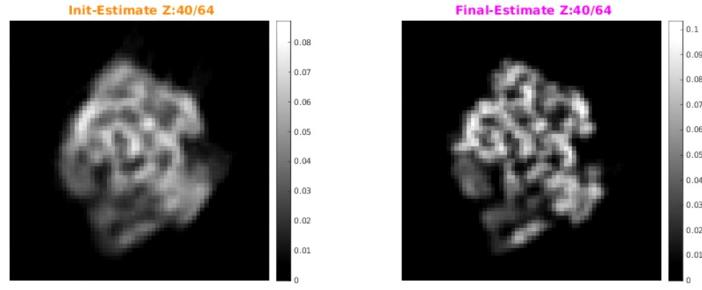
(b) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=25/64



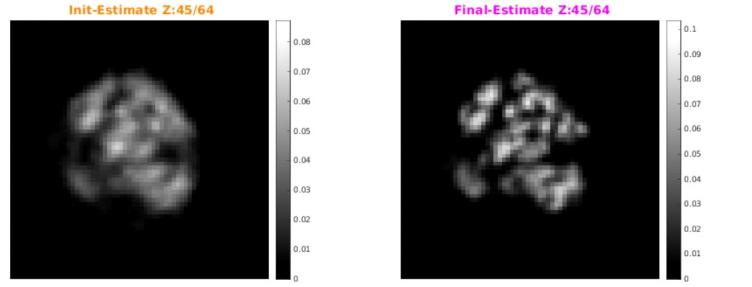
(a) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=30/64



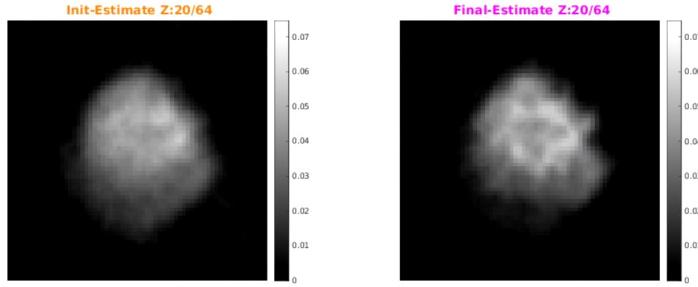
(b) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=35/64



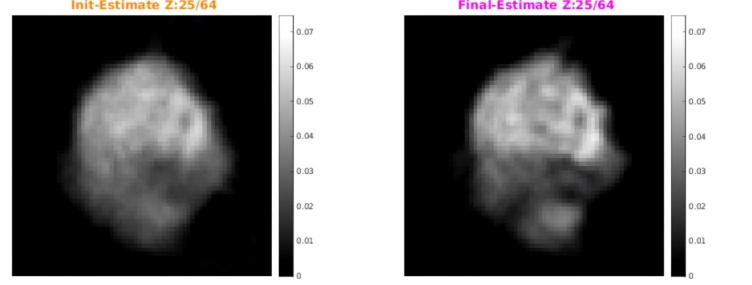
(a) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=40/64



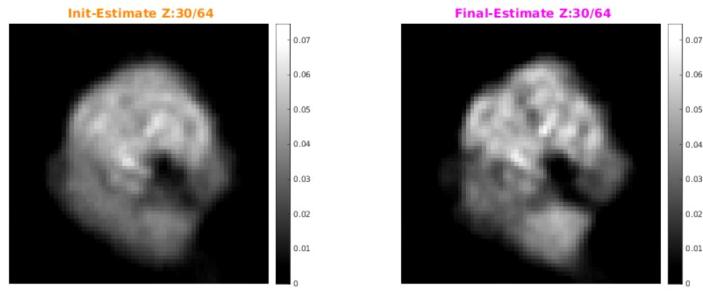
(b) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=45/64



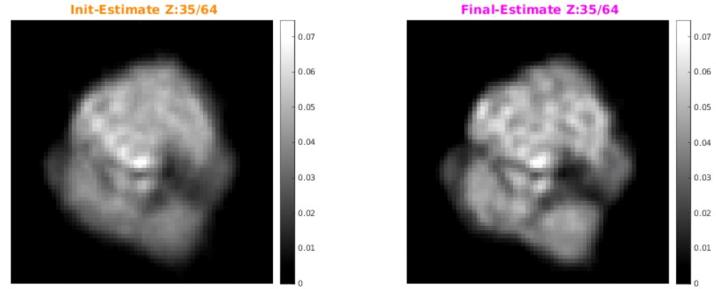
(a) 3D reconstruction of EMDB-8647 [15] | Noise=30%
| Shift=10 pixel | Cross-section at Z=20/64



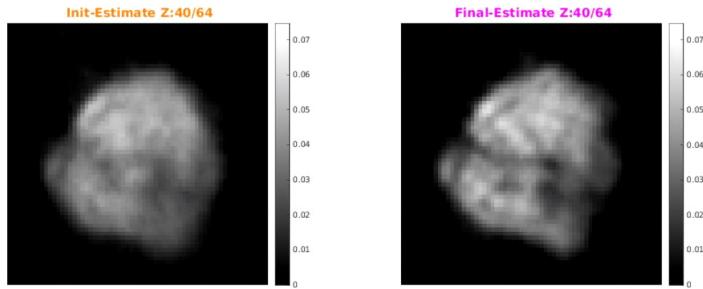
(b) 3D reconstruction of EMDB-8647 [15] | Noise=30%
| Shift=10 pixel | Cross-section at Z=25/64



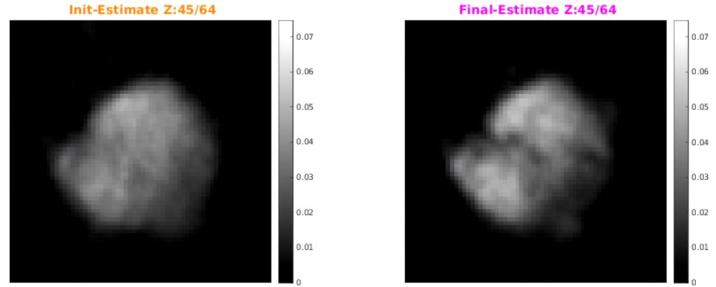
(a) 3D reconstruction of EMDB-8647 [15] | Noise=30%
| Shift=10 pixel | Cross-section at Z=30/64



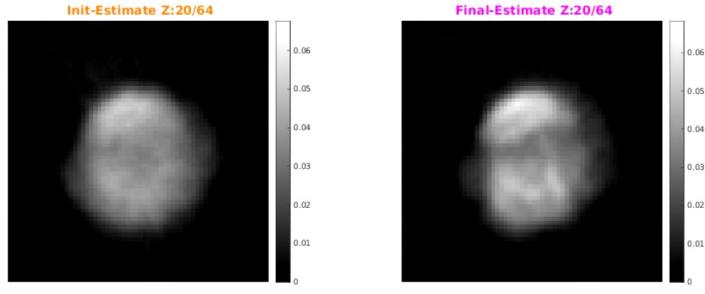
(b) 3D reconstruction of EMDB-8647 [15] | Noise=30%
| Shift=10 pixel | Cross-section at Z=35/64



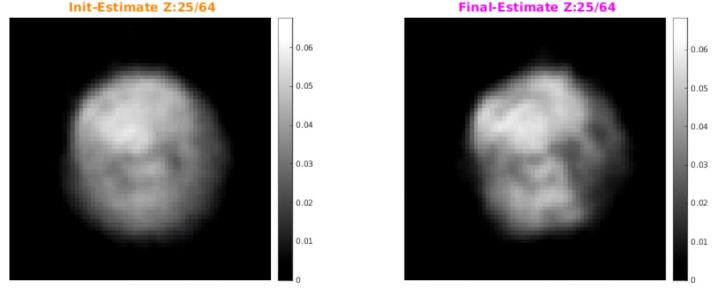
(a) 3D reconstruction of EMDB-8647 [15] | Noise=30%
| Shift=10 pixel | Cross-section at Z=40/64



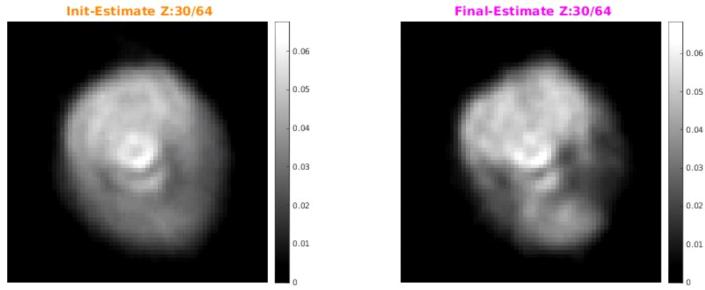
(b) 3D reconstruction of EMDB-8647 [15] | Noise=30%
| Shift=10 pixel | Cross-section at Z=45/64



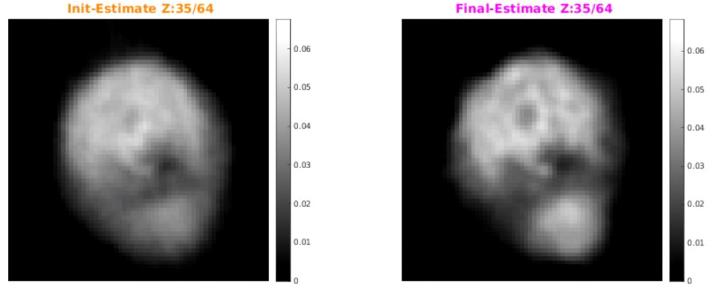
(a) 3D reconstruction of EMDB-8647 [15] | Noise=50%
| Shift=15 pixel | Cross-section at Z=20/64



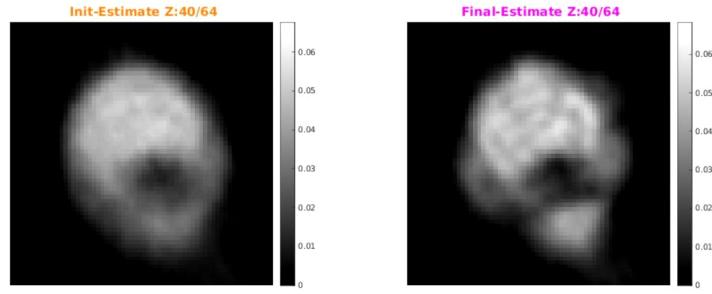
(b) 3D reconstruction of EMDB-8647 [15] | Noise=50%
| Shift=15 pixel | Cross-section at Z=25/64



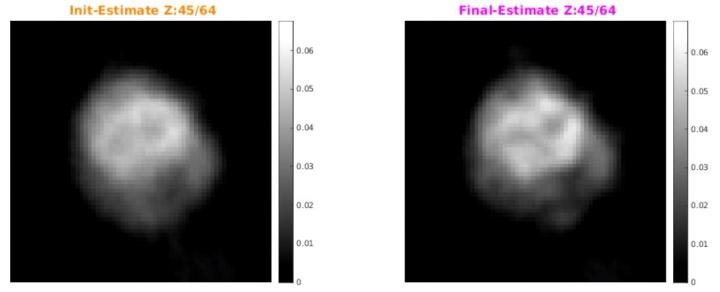
(a) 3D reconstruction of EMDB-8647 [15] | Noise=50%
| Shift=15 pixel | Cross-section at Z=30/64



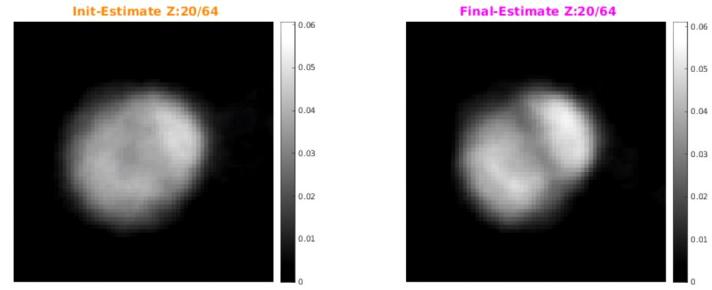
(b) 3D reconstruction of EMDB-8647 [15] | Noise=50%
| Shift=15 pixel | Cross-section at Z=35/64



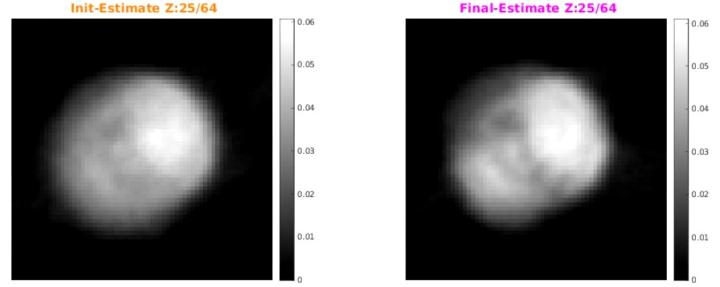
(a) 3D reconstruction of EMDB-8647 [15] | Noise=50%
| Shift=15 pixel | Cross-section at Z=40/64



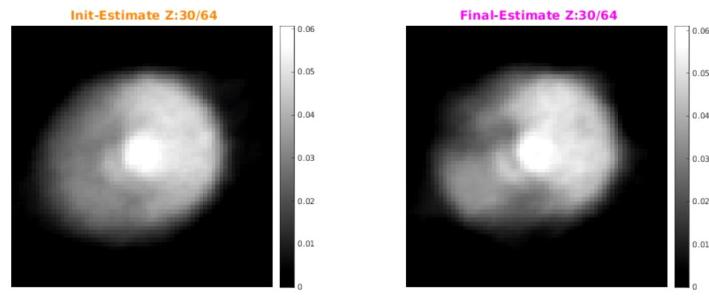
(b) 3D reconstruction of EMDB-8647 [15] | Noise=50%
| Shift=15 pixel | Cross-section at Z=45/64



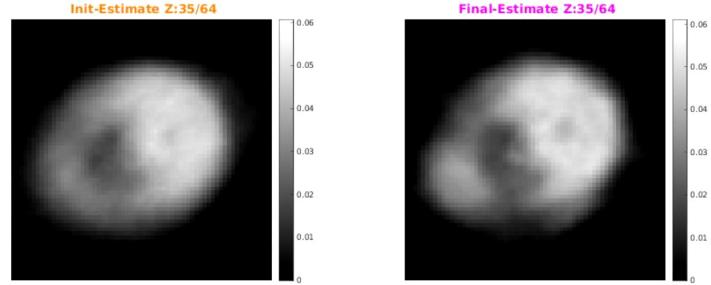
(a) 3D reconstruction of EMDB-8647 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=20/64



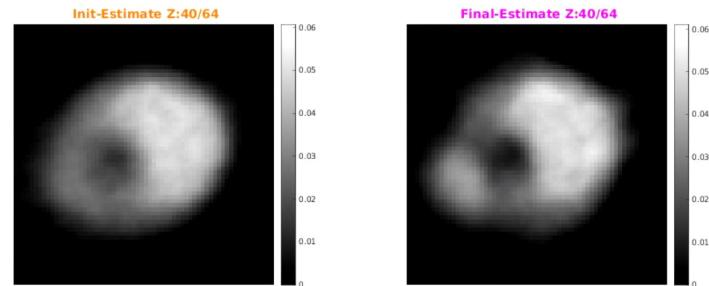
(b) 3D reconstruction of EMDB-8647 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=25/64



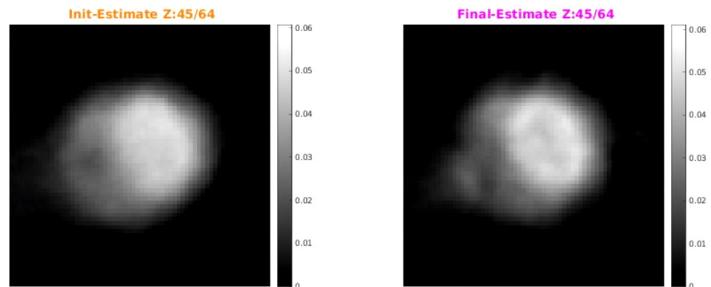
(a) 3D reconstruction of EMDB-8647 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=30/64



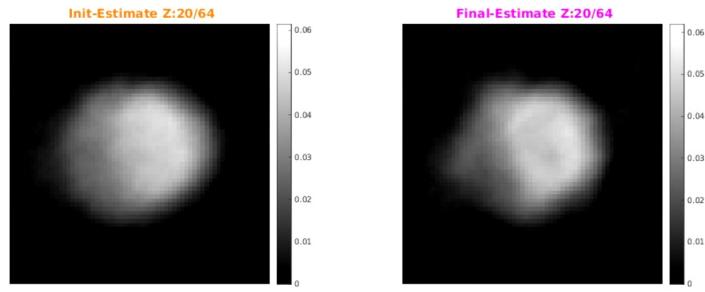
(b) 3D reconstruction of EMDB-8647 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=35/64



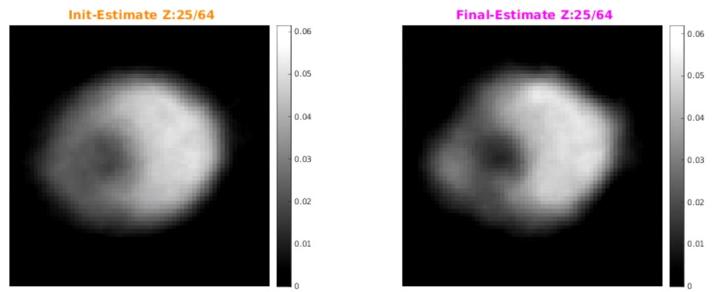
(a) 3D reconstruction of EMDB-8647 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=40/64



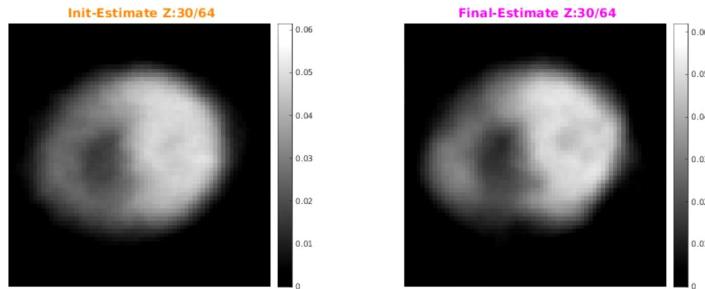
(b) 3D reconstruction of EMDB-8647 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=45/64



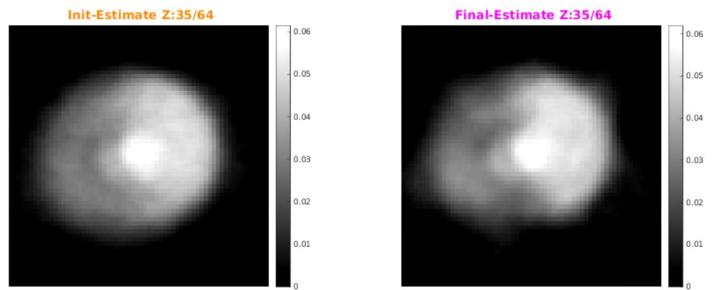
(a) 3D reconstruction of EMDB-8647 [15] | Noise=100%
| Shift=15 pixel | Cross-section at Z=20/64



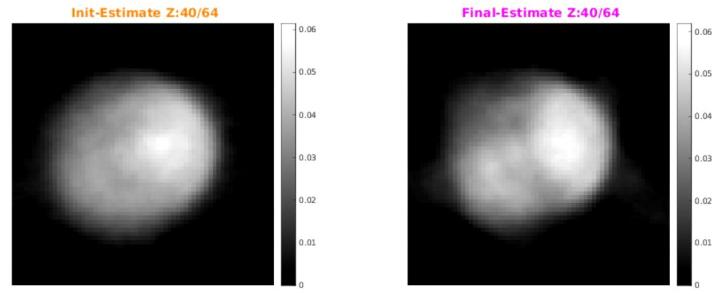
(b) 3D reconstruction of EMDB-8647 [15] | Noise=100%
| Shift=15 pixel | Cross-section at Z=25/64



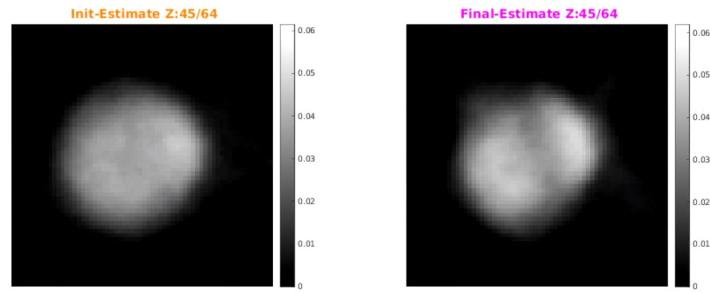
(a) 3D reconstruction of EMDB-8647 [15] | Noise=100%
| Shift=15 pixel | Cross-section at Z=30/64



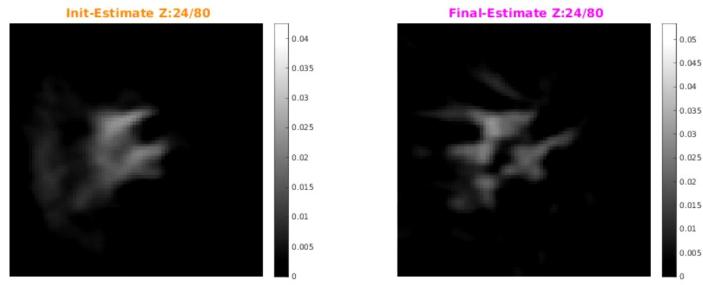
(b) 3D reconstruction of EMDB-8647 [15] | Noise=100%
| Shift=15 pixel | Cross-section at Z=35/64



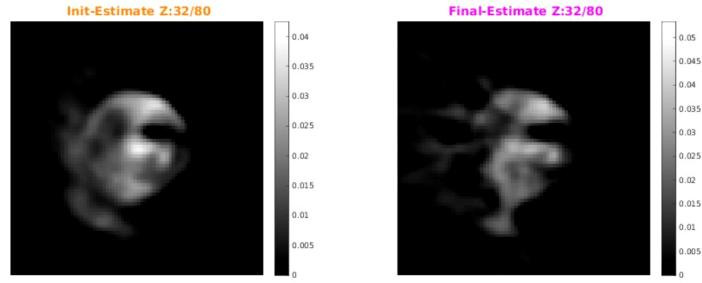
(a) 3D reconstruction of EMDB-8647 [15] | Noise=100%
| Shift=15 pixel | Cross-section at Z=40/64



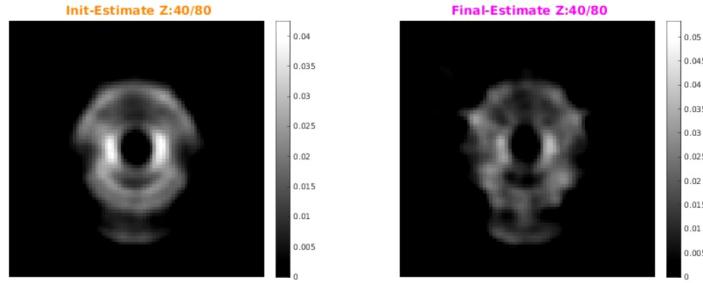
(b) 3D reconstruction of EMDB-8647 [15] | Noise=100%
| Shift=15 pixel | Cross-section at Z=45/64



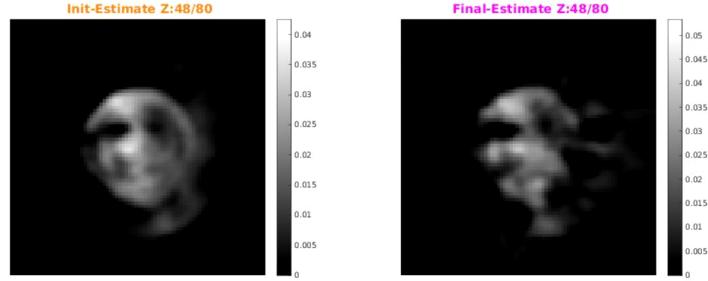
(a) 3D reconstruction of EMDB-5689 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=24/80



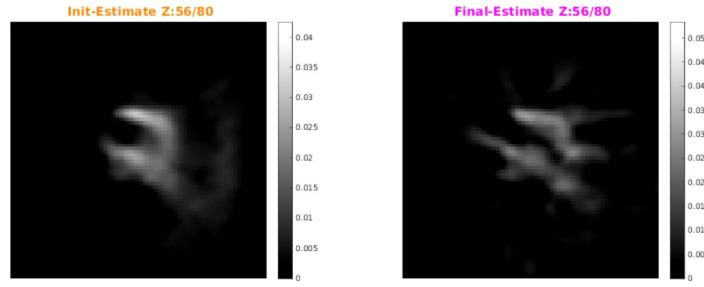
(b) 3D reconstruction of EMDB-5689 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=32/80



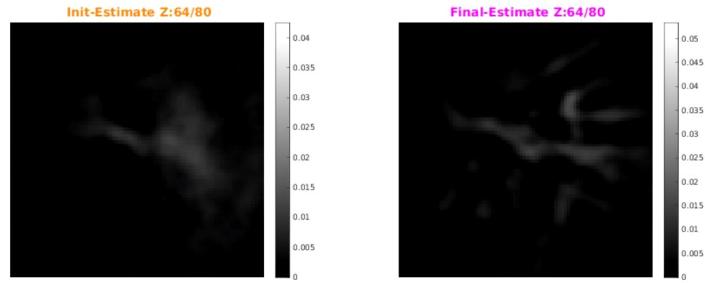
(a) 3D reconstruction of EMDB-5689 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=40/80



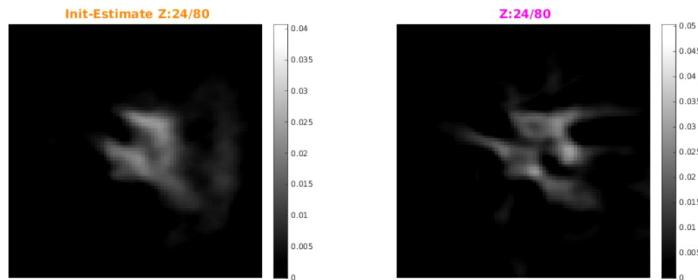
(b) 3D reconstruction of EMDB-5689 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=48/80



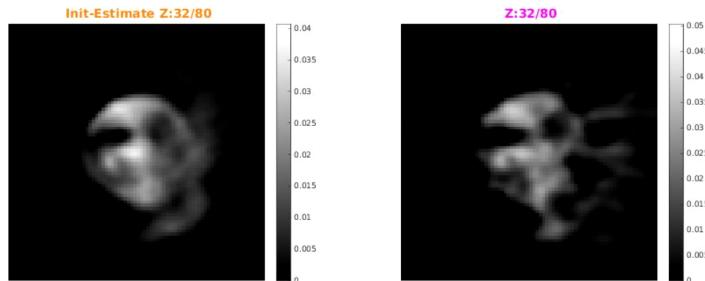
(a) 3D reconstruction of EMDB-5689 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=56/80



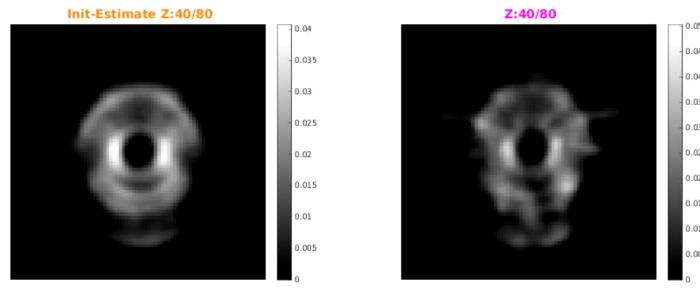
(b) 3D reconstruction of EMDB-5689 [15] | Noise=10%
| Shift=3 pixel | Cross-section at Z=64/80



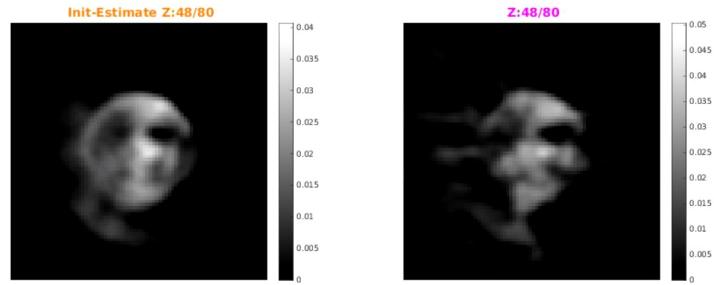
(a) 3D reconstruction of EMDB-5689 [15] | Noise=30%
| Shift=15 pixel | Cross-section at Z=24/80



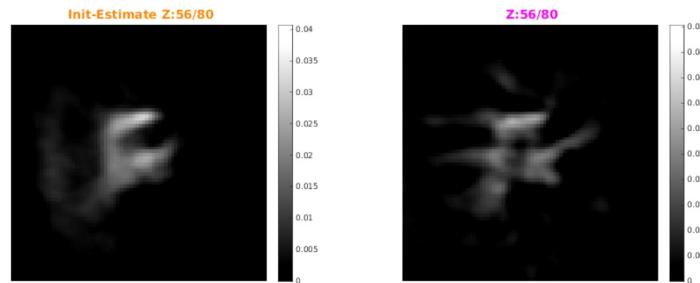
(b) 3D reconstruction of EMDB-5689 [15] | Noise=30%
| Shift=15 pixel | Cross-section at Z=32/80



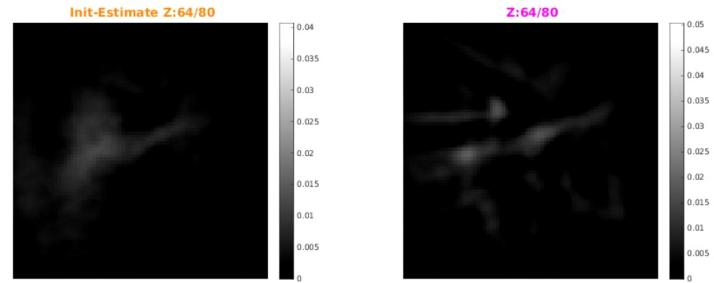
(a) 3D reconstruction of EMDB-5689 [15] | Noise=30%
| Shift=15 pixel | Cross-section at Z=40/80



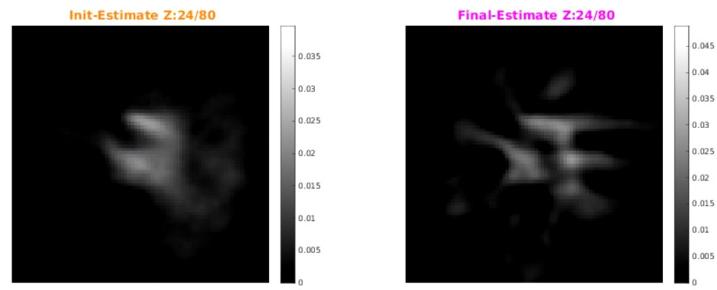
(b) 3D reconstruction of EMDB-5689 [15] | Noise=30%
| Shift=15 pixel | Cross-section at Z=48/80



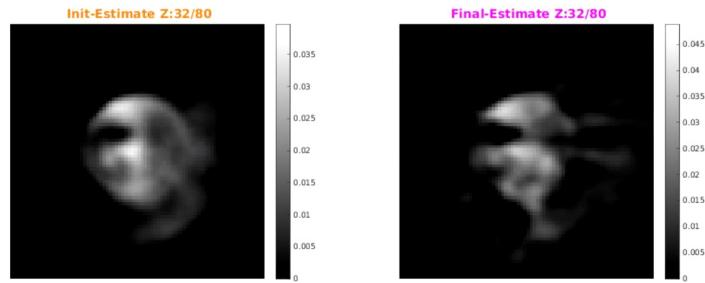
(a) 3D reconstruction of EMDB-5689 [15] | Noise=30%
| Shift=15 pixel | Cross-section at Z=56/80



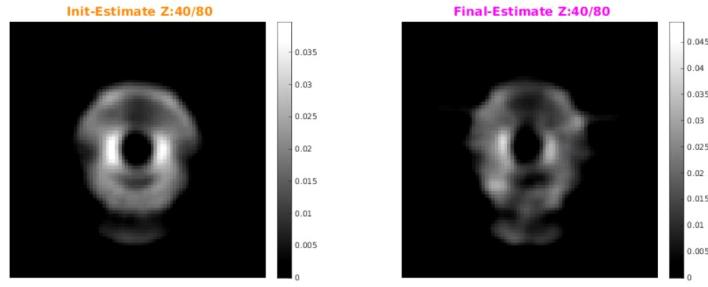
(b) 3D reconstruction of EMDB-5689 [15] | Noise=30%
| Shift=15 pixel | Cross-section at Z=64/80



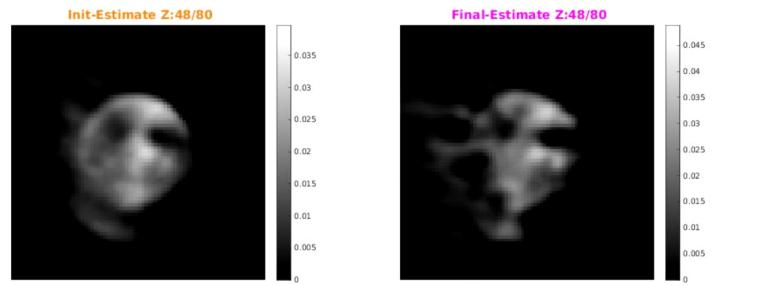
(a) 3D reconstruction of EMDB-5689 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=24/80



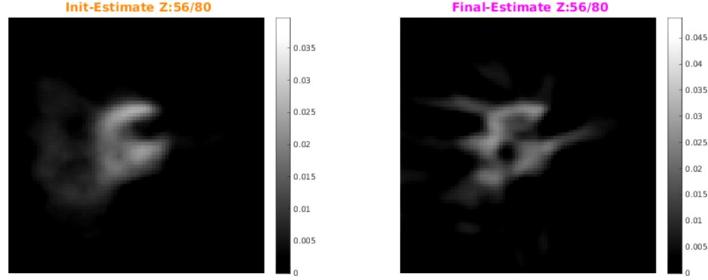
(b) 3D reconstruction of EMDB-5689 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=32/80



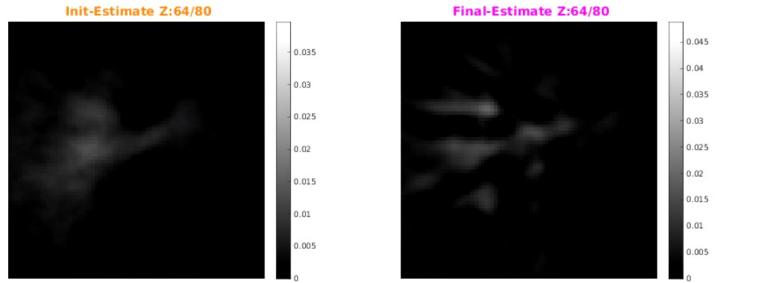
(a) 3D reconstruction of EMDB-5689 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=40/80



(b) 3D reconstruction of EMDB-5689 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=48/80



(a) 3D reconstruction of EMDB-5689 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=56/80



(b) 3D reconstruction of EMDB-5689 [15] | Noise=80%
| Shift=15 pixel | Cross-section at Z=64/80

Stochastic Average Gradient Descent

The cross-sections of reconstructed 3D structures are displayed below. In the results,

total projections = 2000

batch size of projection for each iteration = 200

max iterations = 50

batch size of orientation parameters considered for each iteration = 1000

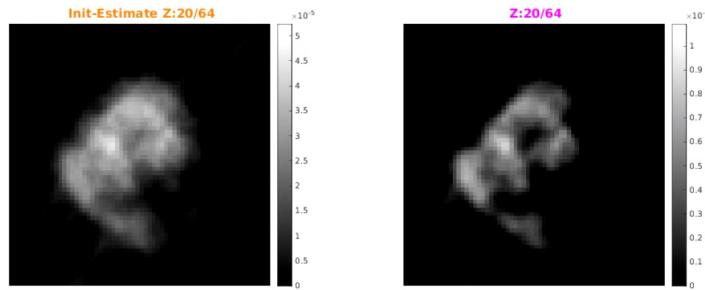
batch size of shift parameters considered for each iteration = 25

base learning rate $\epsilon=0.01$

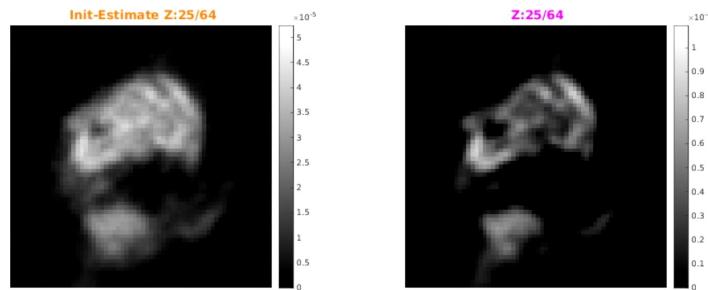
constant in prior $\lambda = 0.0005$

Lipschitz constant initialisation $L = 1$

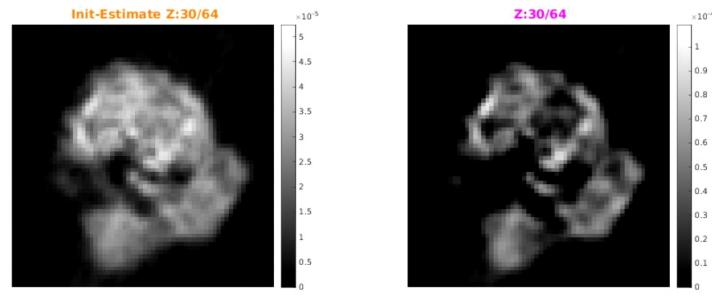
Compared to our previous report [10], these results are comparable in resolution but were generated much faster. In [10], results with similar resolution would take approximately 100 hours to generate, whereas this work generates them in 12-15 hours.



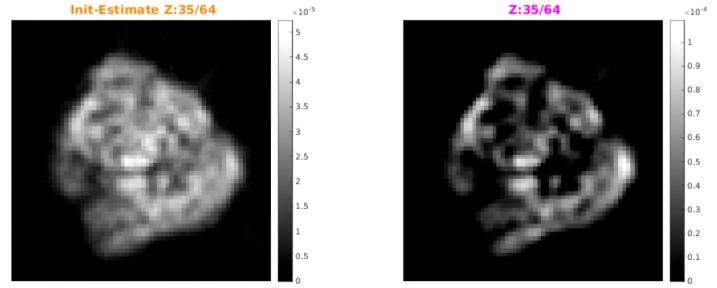
(a) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=10 pixel | Cross-section at Z=20/80



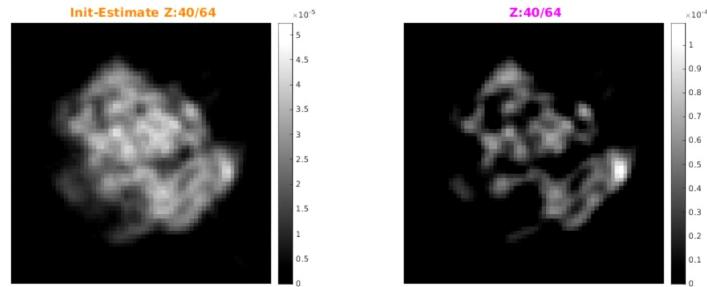
(b) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=10 pixel | Cross-section at Z=25/80



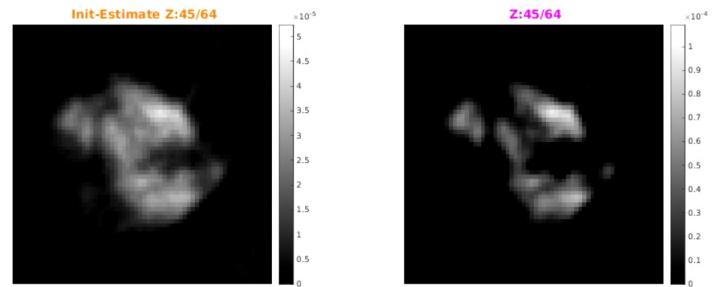
(a) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=10 pixel | Cross-section at Z=30/80



(b) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=10 pixel | Cross-section at Z=40/80



(a) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=10 pixel | Cross-section at Z=45/80



(b) 3D reconstruction of EMDB-8647 [15] | Noise=10%
| Shift=10 pixel | Cross-section at Z=45/80

4.2.2 Real Datasets

Real datasets [7], [8].

Stochastic batchwise Expectation Maximisation

The cross-sections of reconstructed 3D structures are displayed below. In the results,

total projections = 2500

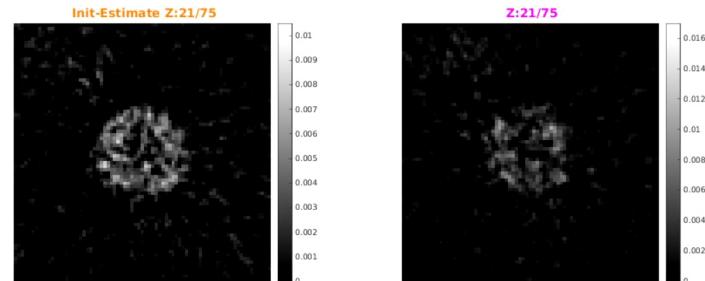
batch size of projection for each iteration = 200

max iterations = 50

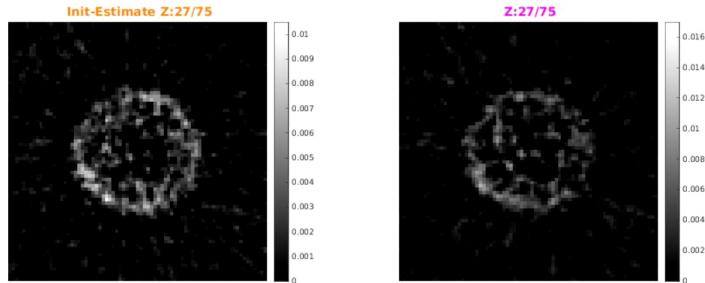
batch size of orientation parameters considered for each iteration = 1000

batch size of shift parameters considered for each iteration = 25

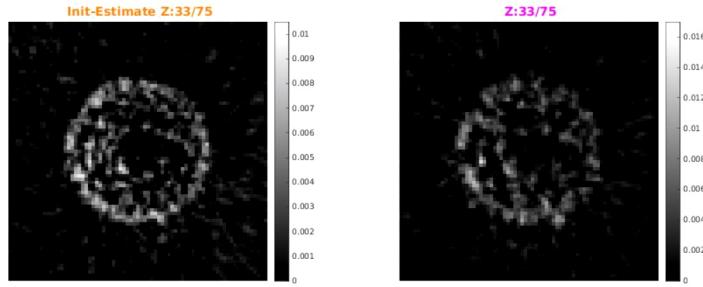
Compared to our previous report [10], these results are comparable in resolution but were generated much faster. In [10], results with similar resolution would take approximately 120 hours to generate, whereas this work generates them in 6-7 hours.



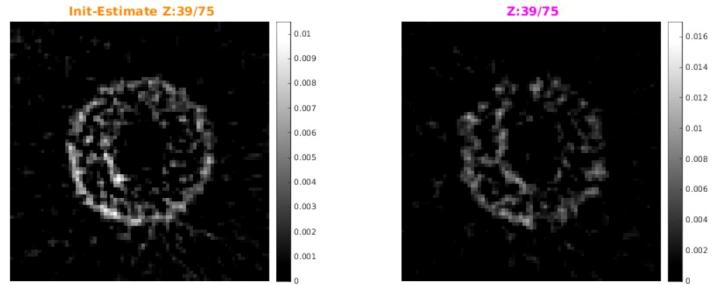
(a) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=21/75



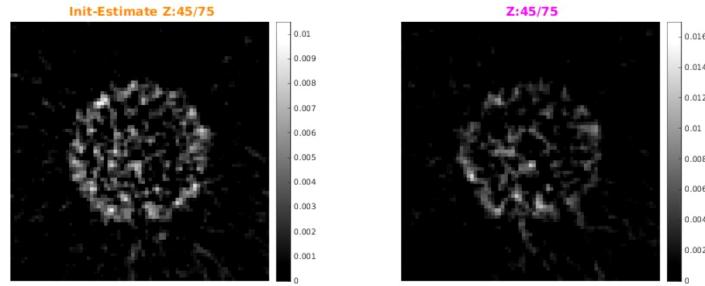
(b) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=27/75



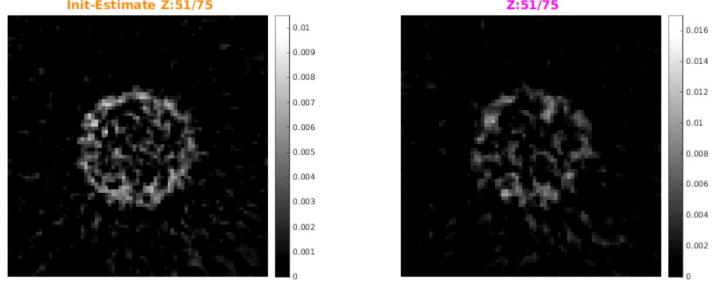
(a) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=33/75



(b) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=39/75



(a) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=45/75



(b) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=51/75

Stochastic Average Gradient Descent

The cross-sections of reconstructed 3D structures are displayed below. In the results,

total projections = 1000

batch size of projection for each iteration = 200

max iterations = 10

batch size of orientation parameters considered for each iteration = 1000

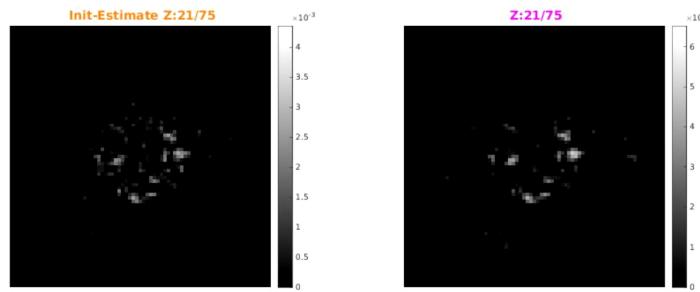
batch size of shift parameters considered for each iteration = 25

base learning rate $\epsilon=1$

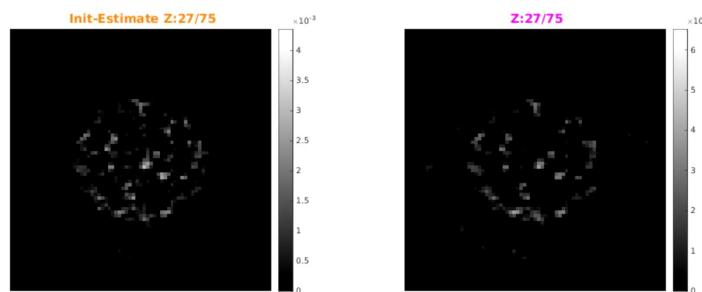
constant in prior $\lambda = 0.00001$

Lipschitz constant initialisation $L = 1$

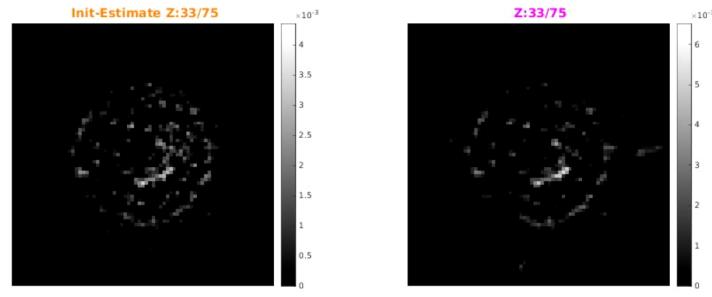
Compared to our previous report [10], these results are comparable in resolution but were generated much faster. In [10], results with similar resolution would take approximately 120 hours to generate, whereas this work generates them in 12-15 hours.



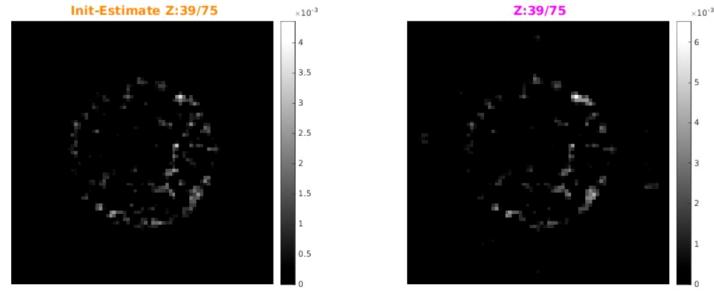
(a) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=21/75



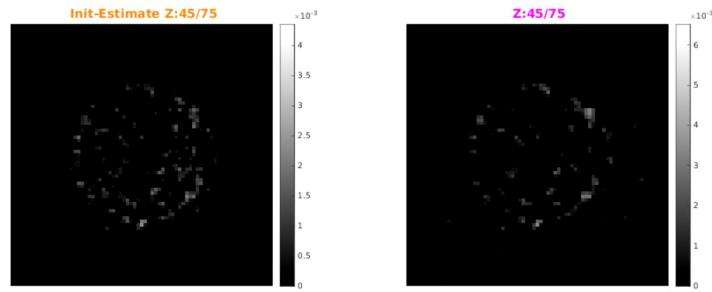
(b) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=27/75



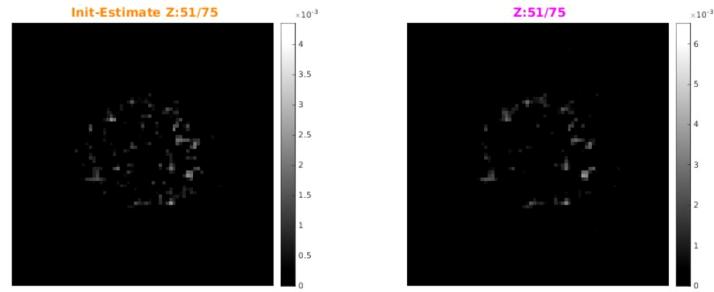
(a) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=33/75



(b) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=39/75



(a) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=45/75



(b) 3D reconstruction of EMPIAR-10295 [7] | Cross-section at Z=51/75

5 Future Work

5.1 CTF correction

During our work, we faced several difficulties with selecting datasets, and even still are not confident with the results on the datasets selected. There are several unknown parameters like CTF and value ranges of the datasets that make reconstruction using them very difficult.

In several of the datasets tried, in the final reconstruction white line like artifacts could be seen in the background which overshadow the 3D structure itself (fig. 44), this error is not due to our parameter assignment steps proposed and occurs even when the particles are assigned to their ideal orientation and shift (tested on simulated dataset). Another issue faced was with the noise removal step, we saw that the noise removal step, in several cases, instead of blurring the background, blurs the particle itself, which in turn causes an error with the coarse shift estimation step. We believe that these issues could be solved by applying CTF correction or scaling all micrographs used to a similar range.

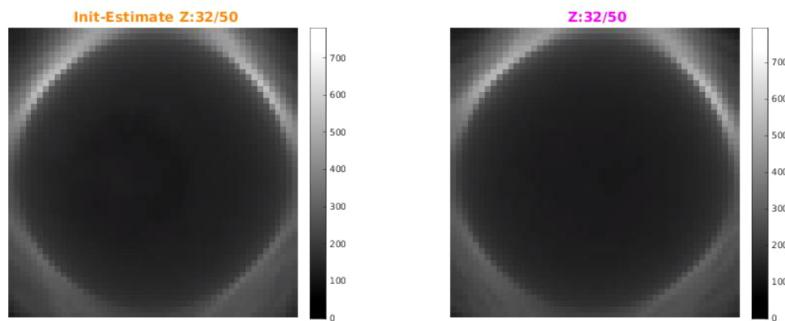


Figure 44: EMPIAR-10029 [8]. Reconstruction using Stochastic batchwise Expectation Maximisation

5.2 Branch and Bound

Branch and Bound is a method used in CryoSPARC for further refinement of the 3D structure to achieve a high resolution after the stochastic refine-

ment has converged. It works in a similar manner to iterative refinement of the orientations and shifts, but here all the projections are used in every iteration. It is more costly because it uses all the projections but note that it is run after stochastic refinement is converged and is only used to refine the structure to a high resolution, thus doesn't run for many iterations. Branch and bound method however, does propose on a method to sample the orientations and shifts so that not all need to be used and time required for the algorithm can be reduced. The optimal orientation or shift in every iteration is conventionally selected by taking a projection of the 3D structure at all possible orientations and shifts and picking the ones which correspond to the least L2 error with the original picked particle. However taking projections for all the parameters and calculating their L2 error is very time taking.

Branch and bound method works by calculating a lower bound of the L2 error for all the parameters. The lower bound is selected to be a function which can be calculated efficiently. After calculating all the lower bounds, the actual L2 error is calculated for the parameters with lowest lower bound. We now have one possibility for optimal parameters, and thus we can reject all the parameter ranges which have their lower bound more than this computed L2 error. Since the L2 error is always greater than or equal to its lower bound, it can never be less than the L2 error for the parameters selected.

The lower bound on L2 error is calculated by first taking the Fourier transform of the picked particles and the 3D structure. We only consider the few lower Fourier frequencies to calculate the L2 error here, i.e. we take the projections of the 3D structure for all the parameters and calculate the L2 error with the original picked particles, but in doing this only consider the lower few Fourier frequencies. This is possible since we are in Fourier domain and taking projection is simply taking a slice of the 3D structure (Fourier slice theorem). We term the L2 error so calculated as the lower bound.

This makes sense as the lower bound, since first, the actual L2 error will consider all the Fourier frequencies and will thus definitely be greater than this. However, it will not be very different as a reasonable assumption is that most of the information content is in the lower frequency range, supported by the inverse square law of frequencies.

In future works, branch and bound can be used to refine the 3D structure further.

5.3 Multiple Particle Case

So far, in our micrographs we only had a single 3D structure and didn't need to worry about clustering and separating particles belonging to different structures prior to reconstruction. Moving forward we could relax this assumption and consider a mixture of particles that would need to be separated before reconstructing. Our work would then be combined with the work of Rupesh, who is working on the clustering problem with Prof. Ajit Rajwade.

6 Acknowledgements

I would like to take this opportunity to thank Prof. Ajit Rajwade for giving me the opportunity to work under his guidance. The work on this project so far would not have been possible without his support and insight. I would also like to thank Khursheed Ali, for laying out the ground work for my work and also for answering questions and debugging code at odd hours. Also thanks to Rupesh for his support at key moments and giving valuable suggestions.

References

- [1] Ajit Rajwade (2019). Tomography slides from CS-754 AIP 2019. IIT Bombay
- [2] Tomography wiki. (2019). Wikipedia article on tomography. <https://en.wikipedia.org/wiki/Tomography>
- [3] Radon transform wiki. (2019). Wikipedia article on radon transform. https://en.wikipedia.org/wiki/Radon_transform
- [4] Fourier slice Theorem wiki. (2019). Wikipedia article on fourier slice theorem https://en.wikipedia.org/wiki/Projection-slice_theorem
- [5] CT Scan wiki. (2019). Wikipedia article on CT scan https://en.wikipedia.org/wiki/CT_scan
- [6] Cryo-EM wiki. (2019). Wikipedia article on Cryo-EM https://en.wikipedia.org/wiki/Cryogenic_electron_microscopy

- [7] EMPIAR, the Electron Microscopy Public Image Archive, EMPIAR-10295 Single particle cryo-EM dataset of clathrin cages <https://www.ebi.ac.uk/pdbe/emdb/empiar/entry/10295>
- [8] EMPIAR, the Electron Microscopy Public Image Archive, EMPIAR-10029 A simulated cryoEM data set of GroEL particles <https://www.ebi.ac.uk/pdbe/emdb/empiar/entry/10029/>
- [9] Khursheed Ali (2019). "Particle Picking and 3D Reconstruction in Cryogenic Electron Microscopy" (M.Tech Thesis). IIT Bombay
- [10] Maitrey Gramopadhye (2019). "3D Reconstruction in Cryogenic Electron Tomography with Translation and Rotational Variance" (B.Tech Thesis phase 1). IIT Bombay
- [11] A. Singer and Y. Shkolnisky (2009). "Three-Dimensional Structure Determination from Common Lines in Cryo-EM by Eigenvectors and Semidefinite Programming"
- [12] Sjors H.W. Scheres (2012). "RELION: Implementation of a Bayesian approach to cryo-EM structure determination". Journal of Structural Biology 180 (2012) 519–530. <https://www.sciencedirect.com/science/article/pii/S1047847712002481>
- [13] Punjani, A., Rubinstein, J., Fleet, D. et al. (2017). "cryoSPARC: algorithms for rapid unsupervised cryo-EM structure determination". Nat Methods 14, 290–296 (2017). <https://doi.org/10.1038/nmeth.4169>
- [14] A. Punjani, M. A. Brubaker and D. J. Fleet (2017). "Building Proteins in a Day: Efficient 3D Molecular Structure Estimation with Electron Cryomicroscopy," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 706-718, 1 April 2017, doi: 10.1109/TPAMI.2016.2627573. <https://ieeexplore.ieee.org/document/7740877>
- [15] The Electron Microscopy Data Bank (EMDB), Protein Data Bank in Europe(PDBe). <https://www.ebi.ac.uk/pdbe/emdb/>
- [16] EMPIAR, the Electron Microscopy Public Image Archive <https://www.ebi.ac.uk/pdbe/emdb/empiar/>