Avoiding Local Minima in Single-Particle Reconstruction

Mathieu Brédif
bredif@stanford.edu

Pierre-Alexandre Pont
ppont@stanford.edu

Uygar Sümbül
uygar@stanford.edu

Abstract

Techniques such as Cryo-EM provide many 2D images of the same particle, with unknown orientations. Existing algorithms to reconstruct a 3D model of the original particle sometimes fail and output a model corresponding to a local minimum. Different methods such as soft(probabilistic) assignment of orientations will be discussed.

1 Introduction

Single-particle cryo-EM is a technique for reconstructing a 3D structure from a large number of randomly oriented two-dimensional projections. Just like in computed tomography, the reconstruction process uses the famous Projection-Slice theorem. Unlike in computed tomography, however, the process must estimate the unknown orientations at the same time that it estimates the 3D structure. Hence, the main problem in single particle reconstruction is the chicken-and-egg problem of determining image orientations without knowing the particle shape, and vice versa [2].

The solution used today iteratively aligns images to the current shape and reconstructs a new shape from the aligned images. Here, the alignment is typically determined by maximum correlation. While this iterative approach is successful in most cases, in some cases, the algorithm converges to a completely incorrect local minimum. Such spurious results limit the use of the technique by experts from a larger biology community, and are the reasons for the problem to demand new approaches. (See [2] and the references therein.)

The standard approach uses hard assignments of orientations. We believe that using soft assignments(probabilistic assignments) can very well improve the performance of the standard approach. Another point is that using multiple initial models can be an improvement on the algorithm, at the expense of increased complexity. Although techniques such as soft assignment have previously been considered, they are applied only to the equivalent problem in 2D, which does not capture all the difficulties of 3D, leaving all the more important 3D problem as an open one.

In this paper, we will present our work so far on the problem, addressing the future work towards a more successful and robust algorithm. Also, we will discuss some of the intermediate decisions at different stages of the algorithm that we have dealt so far. Lastly, we will briefly touch on an alternative method called "tunneling," based on [4], which can be considered as a strategy for optimization problems in general. We will discuss appropriateness of such a method for our problem.

2 Background

2.1 Available data

For each particle, the only available data are many 2D images of the 3D particle, taken from unknown orientations. The assumption is that these images are orthographic projections of the object's density along a particular orientation. Moreover, these images are very noisy, with a typical Signal-to-Noise ratio (SNR) around 0.05. With such values, the human eye cannot see anything but noise. Real data sets include several 10,000s of images for one particle.

2.2 Goal

The goal is to reconstruct a 3D model of the particle. Thus, the intermediate goal is to determine each 2D image's orientation.

2.3 Equivalent problem in 2D

Bern *et al.* [1, 2] have implemented a reconstruction algorithm in the equivalent 2D problem : The data are noisy images of a 2D model that is rotated and translated differently for each image. An improvement (in addition to soft (or hybrid) assignment

of orientations) employed was to low-pass filter each image before determining orientation probabilities and adding it into the reconstruction, which in effect smoothens out the optimization and remove local optima. The algorithm fails with some problematic particles when the SNR is too low (less than 5%).
However, the 2D problem does not capture the whole difficulty of the real 3D version due to the fact that it does not use any projections. Moreover, there is an increase in the unknowns, both in the number of orientations and the model to be reconstructed. Lastly, the sensitivity to the initial guess increases in the 3D case [1].

3 Our approach

3.1 3D standard method

The standard method is iterative : Each step of the algorithm provides us with a 3D model and an orientation for each 2D image. In each iteration:

- For each 2D image, find the orientation that best correlates it to the corresponding projection of the current 3D model.
- Construct a new 3D model with new 2D images' orientations.

Many intermediate decisions should be made to clarify these implicit steps. For example:

- What is the 3D model for the first iteration of the algorithm?
- Does the algorithm converge? How many iterations does it need?
- How to determine the best orientation for each 2D image: How to compute the correlation between the image and a projection of the 3D model? Which orientations are we going to test, given that we can only compute a finite number of correlations?

In the following sections, we will discuss some of these issues along with some other methods towards a better algorithm as well as more efficient testing of the algorithm.

3.1.1 Reconstruction in Fourier domain

A common part to all the algorithms is that they all make use of the Projection-Slice Theorem to reconstruct the 3D model in the Fourier domain. The power of the theorem arises from the fact that it is possible to reconstruct the model just by using 2D projections. In words, the theorem says that the 2D Fourier transform of a projection of the 3D function on a plane equals a central slice, orientation of which is determined by the orientation of the projection plane, of the Fourier transform of the 3D function. Though it is almost certain that the proof of the theorem for the general 3D case appears in the literature, we could not come across the proof in our initial search. For a special case proof, see [3]. In the following lines, we will provide a proof for the general case by extending the general proof for the 2D counterpart of the theorem.

Let P be the plane the 3D model is projected. Let $\mathbf{n}$, $\mathbf{n}^{\perp}$ be *a*ny two unit vectors, perpendicular to each other, forming a plane P' passing through the origin and parallel to the plane P. Then,

$$\delta(\mathbf{n}\cdot\mathbf{x}-\rho_1), \delta(\mathbf{n}^{\perp}\cdot\mathbf{x}-\rho_2)$$

represent two plane deltas, perpendicular to P, whose intersection gives a line delta perpendicular to P. The projection of the 3D function $f(\mathbf{x})$ on P can be expressed as

$$g((\rho)) = \int_{R^3} f(\mathbf{x})\delta(\mathbf{n}\cdot\mathbf{x}-\rho_1)\delta(\mathbf{n}^{\perp}\cdot\mathbf{x}-\rho_2)\,d\mathbf{x}, \quad (1)$$

which is a 2D function of $\rho = (\rho_1, \rho_2)$ with an angle parameter specifying the orientation of $\mathbf{n}, \mathbf{n}^{\perp}$ in P'. Then, taking the 2D Fourier transform of this function yields

$$\begin{aligned} G(\mathbf{r}) &= \int_{R^2}\int_{R^3} f(\mathbf{x})\delta(\mathbf{n}\cdot\mathbf{x}-\rho_1)\delta(\mathbf{n}^{\perp}\cdot\mathbf{x}-\rho_2)\,d\mathbf{x} \\ &\qquad \times e^{-i2\pi\mathbf{r}\cdot\rho}\,d\rho \quad (2) \\ &= \int_{R^3} f(\mathbf{x})\int_{R^2}\delta(\mathbf{n}\cdot\mathbf{x}-\rho_1)\delta(\mathbf{n}^{\perp}\cdot\mathbf{x}-\rho_2) \\ &\qquad \times e^{-i2\pi(r_1\rho_1+r_2\rho_2)}\,d\rho\,d\mathbf{x} \quad (3) \\ &= \int_{R^3} f(\mathbf{x})\,e^{-i2\pi(r_1\mathbf{n}+r_2\mathbf{n}^{\perp})\cdot\mathbf{x}}\,d\mathbf{x}, \quad (4) \end{aligned}$$

which we recognize as

$$F(r_1\mathbf{n}+r_2\mathbf{n}^{\perp}),$$

a central slice of f, which completes the proof.

As a result, the Fourier transform of the function can be reconstructed using 2D projections, and taking the inverse Fourier transform gives the underlying 3D function f back.

At this point, one immediately recognizes a few points that need attention. First, the theorem is proved for a continuous world and assumes that an infinite number of projections are available for ideal reconstruction. An implementation has to take a finite number of projections as input, which certainly degrades the performance. Moreover, since the theorem uses central slices as the building blocks of reconstruction, the information available around the origin will be much higher than that available far from the origin. Another point is that there will be a mismatch between the grids during reconstruction due to rectangular versus spherical gridding. Hence, some kind of an interpolation step is necessary while applying the Projection-Slice theorem. All of these warrant the usual trade-off between performance and execution time. An optimal solution will surely depend on the specific problem requirements at hand.

3.1.2 Convergence of the algorithm

According to [2], the convergence is assured for the algorithm in the 2D case with continuous values of the orientation parameters. But with a sampling of these parameters' space, the algorithm can reach a limit cycle. We assume that a similar situation is encountered in the 3D case.

3.1.3 Correlation function

As mentioned in section 3.1, one of the central issues in the reconstruction process is to determine the best orientation for each image. For this purpose, one needs a criterion to evaluate the similarity between the images.(Some of the approaches correspond to a distance metric in the mathematical sense, while some do not.)

One approach is to use the cross-correlation between the images. One way to find the cross-correlation is to compute the inner product of the images, treated as vectors. However, one has to normalize the test images before taking the inner product to render the algorithm insensitive to additive noise and the energy difference between the images. The reference image need not be normalized since it is the same in all the comparisons.

Another approach is to compute the mean-squared error between the images, again treated as vectors. That is, one computes the sum of the squared differences between each pixel for each image pair. One nice thing about this approach is that it corresponds to computing the Euclidean distance in the vector space. For a more detailed discussion and comparison of these two aproaches, see [5].

For our problem, we have implemented both approaches. It may be necessary to perform a few trial-and-error steps to understand which one serves better to our purposes.

3.1.4 Orientations of the images

We want to compute the correlation of each image for only a finite number of orientations and we want these orientations to be equally distributed on the surface of the sphere. For example, if we use spherical coordinates and use a regular sampling of both angles' intervals : $\theta \in [0; \pi]$ and $\phi \in [0; 2\pi]$, the sphere will be oversampled close to the poles.

3.2 Expectation Maximization algorithm

The major drawback of the standard approach is the hard assignment method. We believe that using a likelihood function representing the probability that observed image comes from a specific alignment is a promising way to ameliorate this problem.

To utilize this approach, one needs a way to maximize the function at hand. The expectation maximization algorithm is the standard way to maximize likelihood expressions of the kind we have.(For a discussion, see [2].) What changes the performance of the algorithm is the prediction of the probabilistic model of image formation. An initial model, appearing in the literature(see the references in [2]) is to assume that there is independent Gaussian noise in the image pixels. However, this assumption of independent pixels is clearly not a very accurate one, and more complicated assignments may be needed.

Please refer to [2] for the expectation maximization algorithm applied to single particle reconstruction.

4 Current results

4.1 Generation of synthetic data sets

To test the algorithm, we first use synthetic data sets, because we can make them as simple or as difficult as we want and we know the results we want to get. That is why we have implemented a simple Matlab function that takes a 3D voxel grid and an orientation (two angles representing latitude and longitude) as input and outputs the corresponding orthographic projection of the density. To do so, the inverse rotation is applied to the rectangular box containing the

3D model, each point of the 3D model is then mapped to its 8 closest neighbours with integer coordinates and finally the projection is computed by taking the sum of the density along the dimension of the orientation (which is now collinear to one of the 3 axis, thanks to the first rotation).
To simulate real data sets, we have also implemented a function that adds gaussian noise to an image with the desired SNR.

4.2 3D Standard method

We have implemented every piece of code demanded by this algorithm : correlation computation between a 2D image and an orientation of the 3D model, interface between regular space and Fourier domain, reconstruction of a 3D model from 2D images' orientations, main loop... For unknown reasons, we didn't yet get any good results. This should be done very soon.

5 Next steps

In this section, we will provide a plan concerning the work in progress and the future work. We believe that the following branching scheme clarifies where the project is heading to, and the specific problems that must be solved towards reaching the project goal:

- **Polishing the 3D reconstruction program:** This program is central to the project, dictated by the problem specifications. – The inputs are 2D projections of the 3D particle at hand. – Therefore, all the algorithms considered make use of this program, necessitating that we use a bug-free and efficient 3D reconstructor.

- **Implementing the standard method:** We find the implementation of the standard method a necessary step because of the following reasons:
  - The approach that we will employ, in its basic form, is actually an extension of the standard method. Hence, implementing the new ideas on this currently existing version is convenient.
  - The standard method is the one currently used in commercial software packages. Therefore, it is plausible to consider it as the reference method while testing the performance of the newly implemented methods. Since our approach will be an improvement over the standard method, in the worst case, the performance should be equal to that of the standard method.

- **Soft assignment of orientations:** This part will most probably be at the heart of all the improvements. Soft assignment procedure uses probabilistic arguments to decide on the orientations. This means that a way of estimating the probabilities will be needed. At this point, models used for the 2D problem may be adapted to the 3D problem at hand. If time permits, we will try to come up with a better model.(The models designed for the 2D problem are not used for their accuracy, rather for the ease of implementation, and short execution times they provide.) A failure in this step corresponds to obtaining the same results for the standard method, since for a specific choice of probabilities, the soft assignment approach converges to the standard method. We believe that a healthy code using soft assignments will perform significantly better. We will also try the following ideas whether this step succeeds or not, since improvements can only be cumulative for our case.

- **Multiple initial models and simulated annealing:** A disadvantage of these two approaches is that they will make the resulting software even slower. Therefore, if the results of the soft assignment method come out to be satisfactory, we may not include these two approaches in the final software.

  The standard cryo-EM software makes a best guess for the initial model. Though execution time may increase significantly, using multiple initial models looks like a plausible way to avoid certain local minima. Simulated annealing can be loosely defined as bouncing around the space of reconstructions for a better solution. For the simulated annealing part, we will need to interface with Dr. Marshall Bern of PARC.

- **Creating synthetic problematic data sets:** This is an essential part of the project.(We have already written the code to generate these sets.) We need sets with incorrect local optima that are hard to avoid, to assess the performance of the various approaches we will be experimenting with.

In the following subsection, we will present a different approach. We consider this part of the project as optional, and we will work on this part if time permits.

5.1 Tunneling

The techniques that we employ in this work are aimed to get out of local minima within reasonable complexity. While we expect these techniques to yield satisfactory results, and solve the problem of single particle reconstruction to serve the needs of structural biology community, yet another alternative comes from considering the problem within the general framework of global optimization problems. We will try this point of view if we cannot obtain the correct 3D models through the previously mentioned models, or if we have enough time after the implementations, since this framework presents a noble way of dealing with such problems. On the other hand, one has to keep in mind the fact that such an approach is likely to be more complex in terms of execution.

"Tunneling" is a deterministic method for solving the following global optimization problem [4]:

$$min\, f(x) \qquad subject\, to \quad x_{min} \leq x \leq x_{max}$$

where

$$x \in \Re^n, \quad f : \Re^n \rightarrow \Re\, and\, f \in C^2.$$

After converging to a local minimum, the method tries to "tunnel" from one "valley" to another and converge to another local minimum there. Let $(x^*, f(x^*))$ be a local minimum. Then, the method solves the inequality problem $T(x) \leq 0$, where

$$T(x) = \frac{f(x) - f(x^*)}{||x - x^*||^{\lambda^*}} \leq 0,$$

is the tunneling function which also creates a pole to destroy the local minimum. (Other tunneling functions are also used in the literature.) This way, the method comes up with a set of sorted local minima, instead of just one local minimum.

We believe that, being a chicken-and-egg problem, our problem may not be appropriate for this method since both the alignments(the inputs) and the 3D structure are estimated on the go. However, this is not a mathematical statement and (after a few modifications) the "tunneling" method may well be the right method to apply.

6 Summary and conclusion

Cryo-Electron Microscopy (cryo-EM) is a technique for obtaining three-dimensional structure of biological molecules from a transmission electron microscope: First, 2D projections of a specimen are acquired using the microscope. Then, the 3D density map is reconstructed from the 2D projections using the Projection-Slice Theorem. In single-particle cryo-EM, unlike in cryo-EM, the reconstruction process must estimate the unknown orientations of the particles at the same time that it estimates the 3D structure. The central issue in single particle reconstruction is, hence, the chicken-and-egg problem of determining the shape of the particle without knowing the orientations and vice versa. The technique that major software packages use starts from an initial guess of the shape and iteratively aligns images to the current shape using hard assignment orientations, and reconstructs the new shape from those aligned images.

Bern *et al.* [1, 2] showed that soft assignment of orientations yields better results on the 2D problem. However, the 2D problem does not capture all the difficulties of the real 3D problem, since it does not involve projections to begin with. We believe that, by the time the project ends, we will be able to demonstrate that soft assigment of orientations (or a hybrid method) yields better results than the standard approach on the real 3D problem; soft assignment model takes each situation into account with a corresponding probability while hard assignment model chooses the best orientation in that iteration and dismisses other possibilities.

One complication readily arises in the soft assignment case; the probabilities should be estimated by the algorithm. Previous work on this subject(see [2] and the references therein) assumed independent Gaussian noise on each pixel, which is more likely due to practical limitations instead of the dictation of an accurate model since neighboring pixels are in general highly correlated. Nevertheless, a good initial step will be to use this assumption for the 3D case.

Another point that deserves attention is that the standard method makes a best guess for the initial model instead of considering more than one initial models. Therefore, an improvement may be obtained by using multiple initial models at the expense of a slower algorithm, the usual trade-off.

Lastly, an important part of the problem is to make the algoritm robust while attempting to add the expectation maximization algorithm. Robustness here refers to the insensitivity of the algorithm to incorrect probabilistic models. Bern *et al.* [1, 2] mentions that the pure likelihood model is not robust; if the true data deviates from the probabilistic model, the reconstruction can fail quite dramatically.

References

[1] M. Bern, J.D. Chen, H. Chi Wong, “Avoiding Local Optima in Single Particle Reconstruction”

[2] M. Bern, J.D. Chen, H. Chi Wong, “A Study of Maximum-Likelihood Methods for Single Particle Reconstruction”

[3] S. R. Deans, “The Radon Transform and some of its applications,” New York, Wiley, 1983.

[4] http://uxdea4.iimas.unam.mx/disca/Optima /TunelMeth.htm

[5] E. Trucco, A. Verri, “Introductory Techniques for 3D Computer Vision,” Upper Saddle River, Prentice Hall, 1998.