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. In this paper, we will discuss different methods such as soft(probabilistic) assignment of orientations, and a modification of the standard method to speed up the algorithm.

1 Introduction

1.1 The Problem

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 may use the Projection-Slice theorem. (See [3, 4] for Fourier Volume Rendering, which implements this 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 models to the current models and reconstructs a new model from the aligned models. 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. Although techniques such as soft assignment have previously been considered, they are applied only to the equivalent problem in 2D, which does not capture most of the difficulties of 3D, leaving all the more important 3D problem as an open one.

1.2 Scientific Gap

We believe that the different approaches and variants that we have proposed can best be described under the following three topics:

- Use of Projection-Slice Theorem
- A Modification of the Standard Method
- Soft-assignment method

All of these items are the results of the attempts towards either a faster algorithm or an algorithm with better results. We note that some of these items are not "new" in the strict sense. A detailed discussion of these items is provided in section 2.2.

1.3 Approach Summary

We needed to continously modify our approach due to the lack of computational resources. As a result, we proposed some methods to avoid local minima while some others to speed up the algorithm. Our initial innovation is making use of the soft-assignment techniques to end up with healthier results. For a faster algorithm, we also used Fourier-based reconstruction and we proposed a modification of the standard hard-assignment method, which can be incorporated into our soft-assignment approach. Summarily, our approach is to solve the problem of avoiding local minima by utilizing a slightly modified soft-assigment technique on a Fourier-domain reconstructor. A detailed discussion will be provided in the following pages.

1.4 Key Results

We managed to program a Fourier-based reconstructor and obtained extremely satisfactory results from the reconstructor using 200-300 slices.(See, for instance, figures 1 and 2.) Most of the other simulations that we would like to run were severely limited by the available compuational resources. However, we managed to obtain some intermediate results by making certain simplifying assumptions. These assumptions have negative effects on the performance, but are nonetheless necessary to obtain meaningful results. We believe that an important result of this study is the fact that our modified hard-assignment algorithm seems to be working. The modification that we have done has nothing to do with avoiding local minima, but renders the overall algorithm faster. Moreover, we believe that this modification can be incorporated into the soft-assignment method. A detailed discussion can be found in section 2.2. In section 3, we have provided certain simulation results.

2 Approach

2.1 Background Tutorial and Basic Tools

2.1.1 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.

2.1.2 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 most of the difficulties of the real 3D version. For a brief discussion of the underlying reasons, please refer to our interim report.

2.1.3 Convergence of the soft-assignment 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.

2.1.4 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. To perform uniform sampling of the sphere, we sample uniformly on $z \in [-1;1]$ and $\phi \in [0;2\pi]$ in cylindrical coordinates. For an illustration of the routine that we have implemented for this purpose, see figure 4. Notice also the sparsity of the planes.

2.1.5 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 in the 2-D case.

2.1.6 The Projection-Slice Theorem

Before moving any further, we believe that a short introduction to the Projection-Slice Theorem will be beneficial.

The power of the Projection-Slice 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 [5]. 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.

2.2 Technical Innovation

As mentioned during introduction, we believe that the different approaches that we have proposed can best be described under the following three items:

- **Use of Projection-Slice Theorem:** This part is not an innovation over the existing literature in the strict sense. It is already known that a Fourier-domain based implementation beats the back propagation algorithm in speed. However, paraphrasing from Dr. Marshall Bern of PARC, the standard software for our cryo-EM single-particle reconstruction problem do not use the Projection-Slice Thorem due to various difficulties in implementation, and hence using the 3-D version of the theorem, which is a lot more complex than its 2-D counterpart, makes a difference.

- **A Modification of the Standard Method:** The standard method computes various reference models at each iteration, and compares the current guess to those reference models.(For a more detailed discussion of the standard method in 2-D, see [2].) When the problem is posed in 3-D as in our case, this method becomes very costly. As far as we know, companies working on this problem have dedicated computer clusters, whereas we only have an extremely limited computational source.(a moderate general purpose laptop PC)

  Therefore, motivated by our limited resources, we decided to modify the standard approach to make it more amenable from a computational requirements perspective: Instead of making the comaprisons in the 3-D model level, we proposed to make the comparisons at the 2-D slice level. This way, we get rid of the problem of constructing various 3-D models at each iteration, which is extremely costly, and we only compute one 3-D model per iteration using the comparisons performed on the 2-D slices. Of course, one expects that a degradation in performance may occur after such a modification on the standard method. Actually, our computational power was not adequate to perform full tests even in this case. Therefore, we made certain simplifications, which we discuss in sectionr̃efimplement.

After the simplifications, we were able to perform a few simulations. Although the results we obtained are, of course, different from the model we are trying to reconstruct,(The algorithm is not allowed to converge. See section 3.2) they are still encouraging in that they start to look like the target model through increasing iterations. For some of these intermediate results, please see section 3.

- **Soft-assignment method:** The standard method uses hard assignments to make a guess at each iteration. Bern *et al.* [1, 2] showed that using a soft-assignment technique yields better results for the equivalent 2-D problem. Inspired from this result, we(together with Marshall Bern) proposed that using the soft-assignment technique may help to avoid local minima. The soft-assignment technique assigns probabilities to each reference model instead of choosing reference i. One bad news about this technique is that it worsens the already existing computational problems. On the other hand, a better result is guaranteed. To see this, if we assign the probabilities as

$$p = (0, \ldots, 0, 1, 0, \ldots, 0), \tag{5}$$

where the ith position is 1 and all others are 0, this method boils down to the hard-assignment method and hence the results of the hard-assignment method are always achievable.

Another point is that, we believe that this approach can be incorporated into our modified standard method approach: Instead of choosing the best slice, one can assign probabilities to the slices, and proceed from that point on. Due to pratical limitations, one an choose to retain only N of the highest probability assignments.

3 Results

3.1 Main Questions

The main points that are being investigated in our experiments were

- making the Fourier-domain reconstructor work and getting rid of the complications it poses in an efficient manner
- avoiding computational problems by means of carefully chosen data sets and meaningfully modified algorithms
- reconstructing the particle by using the modified standard approach and/or soft-assignment approach.

In our opinion, the most important results of this work are yet to be obtained; the computational problems that we faced limited our ability to run the most demanding (and, of course, most important) simulations. Nevertheless, we have implemented the routines such that almost all of them are ready to be run for such ultimate simulations given enough computational resources. In the current situation, we believe that the most important result that we have obtained is the figure, which looks like a rectangular prism/ellipsoid mixture.(See figure 3.) This model is the result of 20-30 iterations, where we start with an ellipsoid as the initial guess. The input slices belong to a rectangular prism.This result is encouraging since it eliminates the need to reconstruct numerous 3-D models at *each* iteration. There may be a slight inefficiency imposed by this modification(A performance analysis may be needed to accurately judge the situation.), nevertheless, the gain in speed has been much more valuable for our situation.

3.2 Implementation Details

We first itemize the simplifications that we have mentioned in section 2.2:

- We always started with an ellipsoid as our initial guess. The performance will necessarily increase when more educated initial guesses (depending on the particle type) are made.
- We kept our models as small as possible to avoid memory problems. Hence, details are omitted due to subsampling of the particles. Better performance may be expected from finer models.

- We did not take more than 300 slices per object to avoid memory problems. Better performance may be expected when more slices are used.

- We allowed at most 30 iterations due to computational limitations. Better performance may be expected when more iterations are allowed or when the algorithm is allowed to converge.

As one of the goals of the projects, we have created certain data sets and problematic models. The data sets that we have used were mainly cubes, rectangular prisms, and ellipsoids. We have also used certain variants such as a cube whose one corner is cut. We have chosen this particular variant after our discussion with Marshall Bern; he told us that such models are excellent examples of local minima. Evidently, one of the reasons for choosing such data sets is to avoid complex structures since they necessarily require higher resolution models. Some of our results are provided at the end.

At this point, we think that providing a coarse outline of our main algorithm is beneficial:

Algorithm

1. Inputs are the Fourier-domain 2-D slices of the object to be reconstructed.(or the projections of the model, additional 2-D FFT's will be needed.) Start with an ellipsoid as the initial guess to capture some eccentricity.

2. Obtain Fourier-domain slices of the current guess at random orientations.

3. Compare the data slices with the guess slices and choose the best ones for each data slice.(Best in the sense of correlation based comparators. For more information, see [6].) If the soft-assignment approach is used, choose more than one slices together with their occurence probabilities.(For more information on probability assignments and the soft-assignment method in general in 2-D, see [2].)

4. Obtain the new guess using the slices kept, and their corresponding orientations.

We would also like to mention that we have made extensive use of the function sliceomatic.m(Author: Eric Ludham, Copyright: The Mathworks Inc.) during our simulations. This function comes in handy for various visualization purposes: rotation, zooming, level setting,...

3.3 Further Comments and Case Studies

3.3.1 Computation time issues - Issues related to the algorithms

Even the "hard-assignment" version of the algortihm can be very long to execute. For example, if N_{images} is the number of 2D images (thus equal to the number of 2D Fourier transforms), and $N_{orientations}$ is the number of oritentations we consider and S is the size in pixel of the images we consider, we will have to compute $N_{images} * N_{orientations} * S$ correlation results, just to have a guess for each image's orientation. Then, the reconstruction time is proportionnal to $N_{images} * S$.

Real-life experiments usually imply thousands of images (N_{images} between 20,000 and 50,000 according to [2]). These images can be of a very good resolution (for example $300 * 300$, ie S 10^5). To get fine enough results, $N_{orientations} = 500$ looks like a reasonable value. It is easily understood that such values lead to very long experiments. Bern *et al.* precises that such computations are run on clusters and usually take tens of hours to complete. This computation power and these computation times were out of reach for our experiments.

We had to reduce all values in order to get expreiments that could be reproduced and analyzed on basic computers. Typical values become : $N_{images} = 100$, $N_{orientations} = 100$, $S = 50 * 50 = 2500$. The really problematic value is the number of orientations. If it is too low (and 100 values is a quite sparse sampling of the suface of the sphere), even the best algorithm can guess real orientations with a big error.

3.3.2 Computation time issues - Issues related to the use of Matlab

All the code used in this project is implemented in Matlab.

The obvious advantage of using Matlab is that all current mathematical objects and functions are available. It proved helpful in this project, as it makes an intensive use of 3D objects, represented as 3D matrix, and direct and inverse Fourier transforms.

The drawback is the slowness and the memory demand of Matlab code. Even with previous values of parameters, a classic computer can quickly run out of memory.

3.3.3 Effect of noise on 2D reconstruction

Bern *et al.* [2] show very good results for reconstruction in 2D with very low signal-to-noise (SNR) ratios.

For example, with a synthetic data set designed to contain a "strong" local optima, the value $SNR = 0.07$ allows them to reconstruct correctly the model while the value $SNR = 0.05$ makes the reconstruction process fall into the local optimum.

3.4 Method for the simulation in the 3D case

In 3D, our own experiments fail to recover such good results with low level of SNR. For simplicity and computation time's sake, we did not conduct these experiments with the whole "hard-assignment" algorithm but by providing the reconstruction process with the original orientations, that is to say the ones from where the "picture" of the object was actually taken. We conduct our experiments with the corner-cut cube, seen in figure 5. The result of the reconstruction process can be roughly separated in 3 cases : a corner-cut cube can be recognized ; a simple cube can be recognized ; no particular shape can be recognized. We used 20 random orientations, and consequently 20 images. The Matlab code we use is quite straightforward :

- choose N_{images} random orientations
- take the 3D Fourier transform of the cube
- take the N_{images} slices of the 3D Fourier transform corresponding to the orientations previously computed. According to the slice theorem, these images are 2D Fourier transforms of images corresponding to the average of the density of the cube along lines perpendicular to the plane of projection.
- add noise to the images. This noise is a Gaussian white noise, such that $\sigma_{noise} = \sigma_{image}/SNR$.
- recontruct the object from these noisy 2D Fourier transforms. For this, we start from a void volume, and subsequently add each 2D Fourier image, as a slice with the right (known) orientation
- take the inverse Fourier transform of this 3D object
- hopefully, the original cube is recovered !

For visual results of the algorithm, see figures 7.

3.4.1 Influence of the number of images used in the reconstruction

The number of images (equal to 20) used in our experiment is quite low, and it could be blamed as the reason of these "bad" results. This is not the case, as it can be seen from figure 6; where we used 50 and 100 images for reconstruction with $SNR = 0.8$. The bigger number of images does not undoubtlessly improve the quality of the reconstructed object.

3.4.2 Effect of wrong hard assignment - The local minimum effect

The interest of the cut cube model is that it is strongly "local optimum" prone : if an image of it is taken such that the cut corner is visible, it can be correlated to another orientation, that is a rotation of the initial one by one of the rotations that would maje a true cube invariant. It is easily understood that especially in presence of noise, such an image can be mistaken. In the case of the cut cube, it would be reconstructed as a complete cube.

3.4.3 Simulation

Since it is very tedious to launch a complete "hard-assignment" reconstruction process, we tried to simulate the result of this reconstruction in order to make the local optimum effect lookable.

With the cut cube, it can be done by modifying some of the know orientations so that they become orientations that would be similar for a complete cube, as described above. Figure 8 shows the result of these experiments for various number of modified orientations during a 20 images reconstruction. It shows that the reconstruction process is already confused with 8 modified orientations (ie 2/5 of the images) : the cut corner is "filled". Increasing the number of mismatched images doesn't look to have a strong impact on the result of the reconstruction.

4 Related Work

Most of the work in this area uses the standard approach. On the one hand, the algorithm is not perfected yet (and the results are sometimes far from satisfactory) and on the other hand, the immense computational issues limit the use of more complicated algorithm. Recently, Bern *et al.* [2] have proposed a probabilistic approach at the expense of increasing computational requirements to avoid local minima, which is our ultimate goal. This paper, however, deals with the equivalent problem in 2-D; the 3-D problem is a lot more complicated and much more demanding. Nevertheless, this work provides us the motivation to use soft-assignment algorithms in the 3-D case since a substantial improvement in avoiding local minima is observed for the equivalent 2-D problem.

5 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. 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.

In this work, we have shown that the reconstructor implementation based on the Projection-Slice Theorem should be prefered over the back-propagation algorithm; we have obtained satisfactory results from our reconstructor.

In our opinion, one of the key results of this paper is that moving the comparisons to the 2-D slice level does not completely destroy the chances of a healthy reconstruction, and makes the algorithm faster. If we can verify this using better computational resources, it means that some of the resources can be freed for more complex algorithms towards avoiding local minima. We also attempted to provide a theoretical justification as to why a soft-assignment method should perform the same as the hard-assignment model in he worst case.

Other than the reconstruction experiments, we also provided case studies on noisy-environment reconstruction and computation time issues. We obtained both satisfactory and not so satisfactory results, which are limited by our computational power. Of course, these experiments are big simplifications of what would happen in a real life experiment, where results could only be worse. That is why one shouldn't be confident in the quality of the result of a reconstruction with cryo-EM images, whose signal-to-noise ratio is apparently much less than 1.

6 Figures

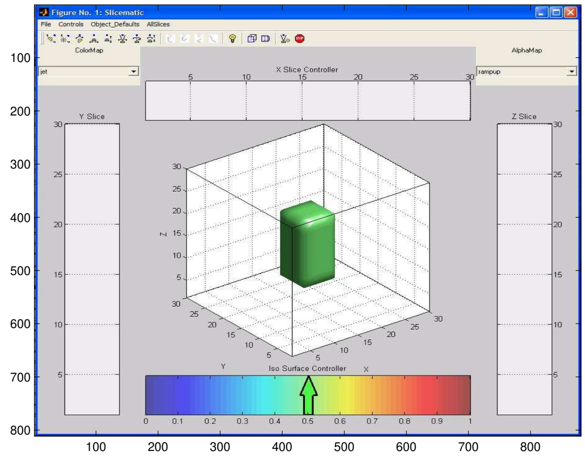

Figure 1: Rectangular prism is reconstructed by providing the orientations.

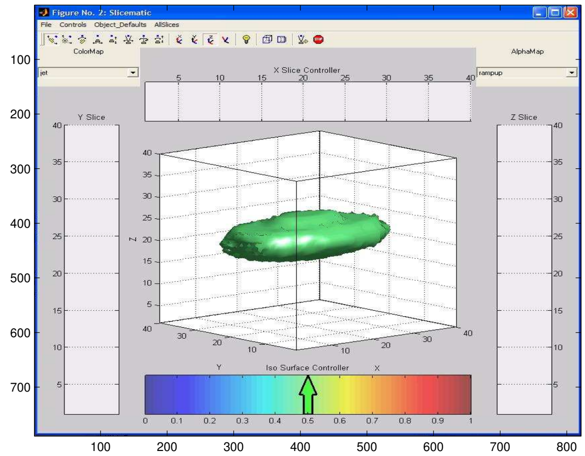

Figure 2: Ellipsoid is reconstructed by providing the orientations. When 100 more slices are used, almost perfect reconstruction is achieved.

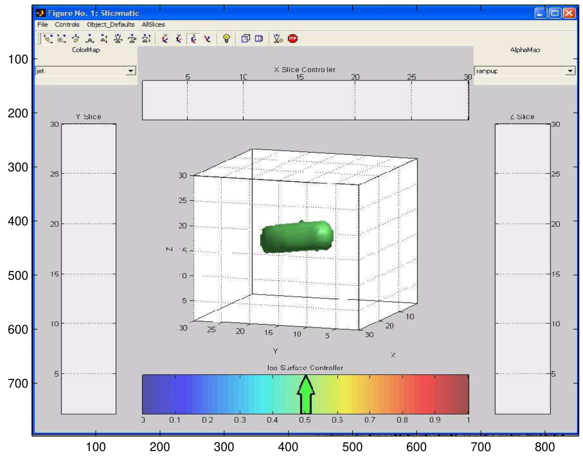

Figure 3: Intermediate result illustrating the reconstruction of a rectangular prism from an ellipsoid

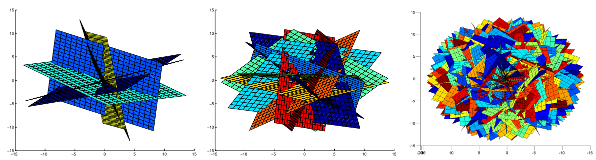

Figure 4: Sparsity of the sampling of the Fourier domain by 5, 10 and 100 planes passing through the DC component

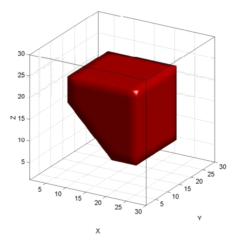

Figure 5: Perspective view of the cut cube used in some experiments

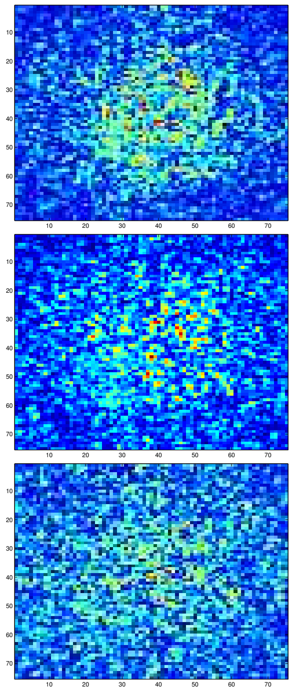

Figure 6: Comparaison of the reconstruction result with $SNR = 1$ and a variable number of images : from top to bottom, the number of images is 20, 50 and 100

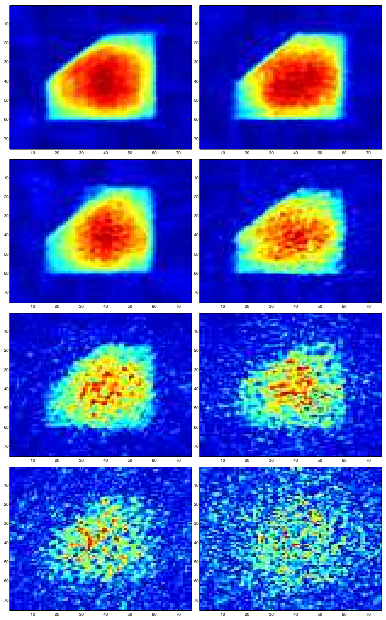

Figure 7: A slice of the reconstructed cut cube for different values of SNR. From left to right, from top to bottom, valus of SNR are : 50, 30, 20, 10, 5, 3, 2, 1

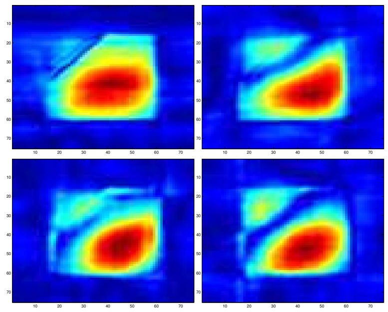

Figure 8: Result of a reconstruction process with known orientations, changing a variable number of orientations to orientations that would give a close result. From left to right, from top to bottom, the proportion of modified orientations is 1/5, 2/5, 3/5 and 4/5

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] M. Levoy, “Volume rendering using the fourier projection-slice theorem.” In Proceedings of Graphics Interface ‘92, pages 61–69. Canadian Information Processing Society, 1992

[4] T. Malzbender, “Fourier volume rendering.” ACM Transactions on Graphics, 12(3):233– 250, July 1993.

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

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