

A fast iterative convolution weighting approach for gridding-based direct Fourier three-dimensional reconstruction with correction for the contrast transfer function

V. Abrishami^a, J.R. Bilbao-Castro^c, J. Vargas^a, R. Marabini^{a,b}, J.M. Carazo^a, C.O.S. Sorzano^{a,d,*}

^a Biocomputing Unit, Centro Nacional de Biotecnología-CSIC, C/Darwin 3, Cantoblanco, 28049 Madrid, Spain

^b Escuela Politécnica Superior, Universidad Autónoma de Madrid, C/Francisco Tomás y Valiente, Cantoblanco, 28049 Madrid, Spain

^c Unidad Asociada al CSIC, 'Supercomputación y Algoritmos', Universidad Almería & Málaga, Spain

^d Bioengineering Lab, Escuela Politécnica Superior, University San Pablo CEU, Boadilla del Monte, 28668 Madrid, Spain

ARTICLE INFO

Article history:

Received 30 September 2014

Received in revised form

18 May 2015

Accepted 24 May 2015

Available online 12 June 2015

Keywords:

Cryo-electron microscopy

Three-dimensional reconstruction

CTF correction

Single Particle Analysis

ABSTRACT

We describe a fast and accurate method for the reconstruction of macromolecular complexes from a set of projections. Direct Fourier inversion (in which the Fourier Slice Theorem plays a central role) is a solution for dealing with this inverse problem. Unfortunately, the set of projections provides a non-equidistantly sampled version of the macromolecule Fourier transform in the single particle field (and, therefore, a direct Fourier inversion) may not be an optimal solution. In this paper, we introduce a gridding-based direct Fourier method for the three-dimensional reconstruction approach that uses a weighting technique to compute a uniform sampled Fourier transform. Moreover, the contrast transfer function of the microscope, which is a limiting factor in pursuing a high resolution reconstruction, is corrected by the algorithm. Parallelization of this algorithm, both on threads and on multiple CPU's, makes the process of three-dimensional reconstruction even faster. The experimental results show that our proposed gridding-based direct Fourier reconstruction is slightly more accurate than similar existing methods and presents a lower computational complexity both in terms of time and memory, thereby allowing its use on larger volumes. The algorithm is fully implemented in the open-source Xmipp package and is downloadable from <http://xmipp.cnb.csic.es>.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Single-Particle Analysis (SPA) is an Electron Microscopy (EM) method wherein the three-dimensional (3D) structure of a biological complex is determined from projections at random orientations of multiple instances of the specimen. Each projection is a two-dimensional (2D) projection of the 3D complex with a random spatial orientation that is additionally modulated by the Contrast Transfer Function (CTF) of the microscope. Upon determination of the orientation parameters, an inversion procedure yields a 3D volume that is compatible with the original projections. However, noisy imaging conditions, CTF effects, errors in orientation parameters, and a finite number of discrete projections not covering the whole spatial domain under study makes this

inversion problem nontrivial [1]. Many approaches have been proposed to solve this ill-posed inversion, which can be categorized into three classes: algebraic, Weighted Back-Projection (WBP), and direct Fourier methods.

Algebraic methods treat this inversion problem as a system of linear equations where well-established algebra methods are employed to find the solution. In other words, the problem is formulated as $\vec{p} = W\vec{v}$, where the 3D object is decomposed into a finite set of basis functions whose coefficients are lexicographically stored in a vector \vec{v} , \vec{p} is a vector with the values of all of the projections' pixels, and W encodes the weight of each of the basis functions onto each pixel. The Algebraic Reconstruction Technique (ART) [2] and Simultaneous Iterative Reconstruction Technique (SIRT) [3] are iterative approaches to solve this system of equations. The general idea behind these methods is to iteratively improve an initial volume by comparing each of the experimental projections with the projections from the current volume, thereby attempting to compensate for this difference. Although algebraic

* Corresponding author at: Biocomputing Unit, Centro Nacional de Biotecnología-CSIC, C/Darwin 3, Cantoblanco, 28049 Madrid, Spain. Fax: +34 91 585 4506.

E-mail address: cos@cnb.csic.es (C.O.S. Sorzano).

methods have the potential to be applied to many different types of reconstructions [1] and to incorporate a variety of constraints [4], they suffer from a high computational complexity. In the SPA field, the introduction of blobs as spherically symmetric basis functions by [5] was one of the most efficient efforts towards a fast algebraic reconstruction, but it is still more computationally expensive compared to the two other aforementioned groups.

WBP [6,7] is the most intuitive method used to reconstruct a 3D object from its 2D projections based on the concept of the Fourier slice theorem. For a set of projections with known orientations, each projection is back-projected across the objective volume from its position in the projection space defined by its orientation. The superposition of all of these back-projections provides an estimation of the original 3D object. To account for the angular distribution of projections, these methods use a weighting function in Fourier space. Weighted back-projection methods are faster than algebraic methods, but they perform poorly in cases where large angular gaps exist, and they have been shown to underperform algebraic methods in a number of cases [5,8].

Based on the central slice theorem, direct Fourier reconstruction (DFR) methods try to obtain the 3D Fourier transform of an object directly from the 2D Fourier transform of its projections, so that an estimation of the original 3D object can be quickly obtained through an inverse 3D Fast Fourier Transform (FFT). In practice, the irregularity of the spatial distribution of the frequency of samples in the set of projections in experimental SPA studies makes the direct use of the inverse FFT unfeasible. Thus, an additional interpolation step is required to obtain the 3D Fourier transform of the object on a regular grid. The so-called gridding algorithm is an alternative method introduced by Penczek et al. [9] into the SPA field; this method was originally developed by Jackson et al. [10] to efficiently estimate the 3D Fourier transform in a regular grid of points using irregularly distributed samples in Fourier space. This algorithm uses an interpolation kernel; in our case this kernel is a modified Kaiser-Bessel (MKB) window function (also known as a blob). The gridding-based direct Fourier method can yield resolutions higher than the algebraic methods (and clearly weighted back-projection methods) in a fraction of their computing time.

Frequency samples from different projections mainly concentrate at the center of the 3D frequency domain, and their sparsity increases as we move away from the center. If no weighting scheme were employed, samples close to the 3D Fourier origin would be over-represented with respect to points away from the origin. In general, the sample values must be corrected by a weight function before inverting the Fourier transform of the volume. An accepted method in SPA to perform this weighting task is to use the volume of the Voronoi region [9] around each Fourier sample. This region is a polyhedron associated with each sample such that the distance between this sample and any point in the polyhedron is shorter than the distance from these points to any of the remaining samples. However, computing Voronoi cells is time consuming (particularly for SPA, where the algorithm has to handle millions of Fourier samples).

An alternate algorithm for obtaining the weighting function was proposed by Matej and Lewitt [11] for Positron Electron Tomography (PET). This approach seeks appropriate weights at each Fourier sample so that it participates with the right weight during the interpolation of regular points. The algorithm begins with the initial weights of the samples and uses convolution with a kernel to iteratively refine these weights. Their proposed method is practical for PET where the geometry of data is known and the number of sampling points is not large, but not for SPA where such conditions are not met.

In this paper, we introduce a gridding-based direct Fourier three-dimensional reconstruction in SPA following the method

suggested by Matej and Lewitt [11] in PET. Our method follows the same iterative scheme for computing the weights, but we estimate the weights at each Fourier sample by evaluating a function instead of storing the collection of weights (which would become impractical in SPA). The proposed approach follows the general idea of Scheres [12], but differs in the way that the weights are calculated. Finally, our algorithm has an additional novel step to compensate for the trilinear interpolation of weights in Fourier space (described in Section 2.4), which improves the resolution of the final reconstruction. The CTF correction is applied during 3D reconstruction and is crucial for a high-resolution structure determination.

We compared the proposed reconstruction algorithm with the algorithms from the SPARX package [13] and the RELION package [12]. In the method by Zhang et al. [13], the projections are first padded to 2 times their original size (default value in the implementation), and then the Nearest Neighbor (NN) interpolation is used to calculate the target 3D Fourier volume. Finally, a weighting function using Bracewell's "local density" [14] is computed to correct the value for each voxel of the 3D Fourier volume.

The experiments showed that our approach is a superior method for 3D reconstruction in terms of accuracy, speed, and memory usage. The new algorithm is fully implemented in the open-source Xmipp package and is downloadable from <http://xmipp.cnb.csic.es>.

2. Material and methods

2.1. Preliminaries

The goal of the gridding-based direct Fourier method is to approximate frequency samples on a regular 3D Cartesian lattice $F_{3D}(\bar{R})$ from the measured samples of the 3D frequency domain $\hat{F}_{3D}(\bar{Q})$ as

$$F_{3D}(\bar{R}) = \int \text{CTF}^{-1}(\bar{Q}) \hat{F}_{3D}(\bar{Q}) K(\bar{R} - \bar{Q}) d\bar{Q}, \quad (1)$$

where \bar{R} is the frequency coordinate within the regular 3D grid and K is the kernel function by which the integration is accomplished. We recommend using a kernel function with some appealing attributes, such as finite size, bell-shaped decay, and differentiability at the borders. The modified Kaiser-Bessel (MKB or blob) is considered to be the best kernel for gridding interpolation by several authors [10,11,15,16]. Matej and Lewitt [16] generally assessed the optimal values for the parameters of the MKB to achieve a reconstruction with good quality. We use an MKB with the same parameter values suggested in their paper. CTF correction is incorporated during interpolation by dividing each irregular sample by $\text{CTF}^{-1}(\bar{Q})$, where $\text{CTF}(\bar{Q})$ is the value of the CTF at frequency \bar{Q} (as a practical implementation issue, this division is performed as long as the CTF is above a given threshold).

Under experimental conditions, a limited number of projections from the specimen are available. Therefore, the discrete form of the integral in Eq. (1) should be considered because measurements are only available at a finite set of frequencies $\bar{Q} \in \{\bar{R}_i\}$. To obtain a discrete form of Eq. (1), the integral is substituted by a summation as shown

$$F_{3D}(\bar{R}) = \sum_i \text{CTF}^{-1}(\bar{R}_i) \hat{F}_{3D}(\bar{R}_i) K(\bar{R} - \bar{R}_i) w(\bar{R}_i), \quad (2)$$

where $w(\bar{R}_i)$ is the weighting factor for the i -th irregular sample. It is important to note that the weighting function is a substitution for $d\bar{Q}$ within Eq. (1). In fact, the value of each irregular sample should be corrected by the weighting factor and the related CTF

value before participating in the interpolation of the points on the 3D regular grid because the distribution of the samples in Fourier space is non-uniform. If no weighting scheme was applied, the regular points close to the origin would contribute more to the summation and the points further away would contribute less, generating undesired artifacts and yielding the wrong reconstructions. In Penczek et al. [9], this weight is proportional to the volume of the Voronoi regions around each experimental Fourier sample.

In the following sections, we describe in detail our iterative algorithm for obtaining the weighting function.

2.2. Calculation of the weighting function

In this section, we propose a numerical iterative approach for computing the weighting function $w(\bar{R})$ based on the method introduced by Matej and Lewitt [11] for PET. In their method, the weighting coefficients are computed for each sample in the 3D frequency space. Because the PET scanner geometry is fixed, this calculation can be performed once and the result can be stored for subsequent reconstructions. Additionally, due to the scanner characteristics the number of Fourier samples is relatively low compared to the number of samples in 3DEM. However, this algorithm is not applicable to SPA, where hundreds of thousands of projections of several hundred pixels may exist per side with no prior knowledge about their angular distribution. Thus, the algorithm has to compute the weighting coefficients for a large number of samples (in the order of billions) for each new set of projections, which is not efficient in terms of both time and storage complexity. Our proposed method is adapted to SPA to overcome these issues.

The proposed approach begins with an initial weighting function that is initialized as a constant function of value 1 as

$$w^{(0)}(\bar{R}) = 1. \quad (3)$$

Then, this weighting function is refined through a number of iterations by evaluating Eq. (2) with $F_{3D}(\bar{R}_i) = 1$ at the locations specified by the projection directions until the reconstructed function $F_{3D}(\bar{R})$ is as close to 1 as desired (formally that the Chebyshev norm of the function $F_{3D}(\bar{R}) - 1$ is smaller than ε , $\|F_{3D}(\bar{R}) - 1\|_\infty < \varepsilon$). Then, the refined weighting function can efficiently remove the effect of non-uniformity by multiplying it by the estimated Fourier Transform of the 3D object.

To iteratively refine the weights, we first evaluate Eq. (2) to compute $c^{(n+1)}(\bar{R})$ at the position of each point on the 3D regular grid, as below

$$c^{(n+1)}(\bar{R}) = \sum_i w^{(n)}(\bar{R}_i) K(\bar{R} - \bar{R}_i), \quad (4)$$

where $w^{(n)}(\bar{R}_i)$ is the interpolated weight from $w^{(n)}(\bar{R})$ of the n -th iteration at the position of the i -th input sample. We use trilinear interpolation to interpolate $w^{(n)}(\bar{R}_i)$ from $w^{(n)}(\bar{R})$. $c^{(n+1)}(\bar{R})$ is the evaluation of Eq. (2) when all samples have a value of 1 and there is no CTF.

After computing Eq. (4) at the position of each frequency point on the 3D regular grid, the algorithm updates the weights from the previous step by dividing them by the obtained convolution in Eq. (4):

$$w^{(n+1)}(\bar{R}) = \frac{w^{(n)}(\bar{R})}{c^{(n+1)}(\bar{R})}. \quad (5)$$

After a number of iterations, the points in $c^{(n)}(\bar{R})$ converge to values close to one and the weighting function reaches stability. At this point, $w^{(n)}(\bar{R})$ can be used to correct the values of regular

samples to overcome the non-uniformity problem. Note that $w^{(1)}(\bar{R}) = \frac{1}{\sum_i K(\bar{R} - \bar{R}_i)}$. Substituting this value into Eq. (2) results in

$$F_{3D}(\bar{R}) = \frac{\sum_i F_{3D}(\bar{R}_i) K(\bar{R} - \bar{R}_i)}{\sum_i K(\bar{R} - \bar{R}_i)}$$

that is the well-known formula for a kernel interpolator. In this way, we see that our method goes beyond kernel interpolation by further guaranteeing that the different weights result in a constant interpolated function when the input samples are constant.

Note that Eq. (5) is similar to Eq. (A3) in Scheres [12]. However, there are some important differences: (1) the motivation of Eq. (5) and Eq. (A3) of Scheres [12] is different. In our algorithm the weights are constructed as a way of discretizing the convolution in Eq. (1), while in Eq. (A3) of Scheres [12] the weights are meant to represent the inverse of the denominator in Eq. (3) of Scheres [12]; (2) Eq. (A3) has a term Ω that is far from trivial and is not present in Eq. (5); (3) the starting values of both sets of weights ($w^{(0)}(\bar{R})$ in our algorithm and the corresponding starting point in Relion) are different, so that both sets of weights become increasingly dissimilar as the iterations increase (due to reason 2, the iterative step is also different); and (4) our iterative scheme normally converges in 2–3 iterations, while the scheme in Relion is normally run for 10 iterations, probably reflecting a larger distance between the starting and finishing sets of weights.

2.3. Compensation for the convolution kernel

Eq. (1) represents a convolution in Fourier space of a function given by its samples $F_{3D}(\bar{R}_i)$ and the kernel $K(\bar{R})$. This is equivalent to a multiplication in real space of the function $f_{3d}(\bar{r})$ and the inverse Fourier transform of the kernel $k(\bar{r})$. This means that we need to divide by $k(\bar{r})$ to recover the function representing the macromolecular complex. This is a well-known effect that is already handled by the standard gridding algorithm. In our implementation, the kernel function in Fourier space is given by the MKB function with parameters a (the width of the kernel), α (a smoothness parameter), m (the order of the Bessel function), and n (the dimension in which the kernel is defined, in our case $n=3$), whose definition is

$$K(R) = \begin{cases} \frac{\left(\sqrt{1 - \left(\frac{R}{a}\right)^2}\right)^m I_m\left(a\sqrt{1 - \left(\frac{R}{a}\right)^2}\right)}{I_m(\alpha)} & 0 \leq R \leq a \\ 0 & R > a \end{cases}$$

Note that R is the module of the frequency vector \bar{R} and I_m is the modified Bessel function of the 1st type and order m . The inverse Fourier transform of the kernel is

$$k(r) = \begin{cases} \frac{(2\pi)^{\frac{n}{2}} a^n I_{\frac{n}{2}+m}\left(\sqrt{a^2 - (2\pi ar)^2}\right)}{I_m(\alpha) \left(\sqrt{a^2 - (2\pi ar)^2}\right)^{\frac{n}{2}+m}} & 0 \leq 2\pi ar \leq a \\ \frac{(2\pi)^{\frac{n}{2}} a^n I_{\frac{n}{2}+m}\left(\sqrt{a^2 - (2\pi ar)^2}\right)}{I_m(\alpha) \left(\sqrt{a^2 - (2\pi ar)^2}\right)^{\frac{n}{2}+m}} & 2\pi ar > a \end{cases}$$

where J_m is the Bessel function of order m .

2.4. Compensation for the interpolation in Fourier space

It is important to note that we used trilinear interpolation to estimate $w(\bar{R}_i)$ from $w(\bar{R})$ in frequency space, which is equivalent to the convolution in 3D Fourier space by a trilinear kernel. Compensating for this interpolation kernel is extremely important because the 3D reconstruction is otherwise masked in real space

by the inverse Fourier transform of the interpolation kernel, resulting in a loss of intensity as the diameter of the macromolecule being reconstructed increases. None of the previously published works on Fourier gridding that performed any type of interpolation of the weights in Fourier space (i.e., Scheres [12]) mention this effect. To the best of our knowledge, the trilinear kernel does not have a known inverse Fourier transform, but can be approximated by a spherically symmetric kernel [17] whose shape in Fourier space is defined as

$$K_i(R) = \begin{cases} 1 - R & 0 \leq R \leq 1 \\ 0 & R > 1. \end{cases}$$

This convolution in Fourier space must be corrected in real space by dividing by the function

$$k_i(r) = \left(\frac{\sin(\pi r)}{\pi r} \right)^2.$$

2.5. Parallelization of the algorithm

The processing time of the algorithm depends on the number and size of the projections, which may amount to hundreds of thousands of projections with several hundred pixels per side for SPA. However, most of the processing steps can be accomplished independently on each projection image, which makes the algorithm naturally suitable for parallelization to achieve higher processing speeds. Parallelization of the algorithm has been performed at two levels: at the level of multiple processors distributed over several computing nodes using the Message Passing Interface (MPI) [18] and at the level of threads executing in the same node using the POSIX threads [19].

Three major computations are discernible in our proposed reconstruction algorithm: FT of projection images, FT of the 3D object on a regular grid, and calculation of the weighting function. The rest of this section explains our approach for parallelizing each of the aforementioned steps.

Considering a multicomputer and multithreaded scenario, our algorithm acts as follows. First, a master process is created at one of the cluster nodes (computer). This process (master) spawns other processes (slaves) at different nodes in the cluster using MPI tools. The master process is in charge of assigning tasks to the slaves and coordinating them. It is also responsible for gathering results from the slaves and building up the final, complete 3D reconstruction. Once the slaves are spawned, they initialize their local memory structures to hold 3D grids and FTs. Upon initialization, they spawn on their own a series of threads that will be used locally to make use of multiple processors and/or cores. Once these initialization steps are completed, the slave informs the master that it is ready to process. Then, the master communicates back to the slave the range of projections it will have to process. Finally, each slave will proceed independently to build a partial 3D regular grid based on this set of projections using threads to speed up the process.

The computation of the FT of each projection image can be performed independently without any interference, with an expected speedup for this step growing almost linearly with the number of available processor/cores. When executing in multithreaded mode, each thread computes the FT of a projection image and retains it in its memory. Note that at this step, the threads do not continue with another projection image, but instead share their local FTs with the other threads to jointly compute the effect of their local irregular samples on the final interpolation of the 3D regular grid. Thus, each thread is responsible for processing a number of rows of the image and uses the MKB window function to estimate the impact of each sample in these rows on each

regular sample. After finishing with the local FTs for all threads, new projection images are loaded by threads to continue with the same process until no more projection images are available. It is worth mentioning that the number of rows to process can be reduced by avoiding exploration at certain resolutions. To ensure the highest performance, row processing is dynamically balanced. Usually, many more rows than threads will exist, but not all rows will necessarily represent the same workload (see previous sentence). If the number of rows is equally distributed to the existing threads, some threads could finish their assigned tasks while others are still running. Thus, rows to be processed are dynamically assigned as threads finish processing previously assigned rows. This mechanism must be arbitrated to avoid the same rows being processed by different threads. To allow this, synchronization is accomplished by defining a shared bit array in which each bit shows whether a row has already been processed.

Finally, the master will ask for the interpolated values of the 3D grid estimated by each node and sum them up to obtain the final interpolated 3D grid. This grid must now be weighted before proceeding to perform the inverse FT and obtain the final 3D reconstruction.

To obtain weights, after computing the 3D regular grid we use the same parallelization strategy to compute the weights on a regular grid. The only difference here is that now all of the nodes should retain the total weights at the position of the 3D regular grid to compute the convolution in Eq. (4). For this mechanism, each node computes the weights at the position of the regular grid points by mean of its projection images and waits until the rest of the nodes have also completed this task. At this point, all nodes combine their values orchestrated by the master node. Then, in the next iteration of weight refinement these corrected weights are used by the processors to compute Eq. (5).

Finally, the master node carries both the interpolated 3D regular grid and weighting function. Then, the final 3D reconstruction is obtained by applying the weighting function and an inverse FFT. The schema of our parallel framework for computing the interpolated 3D regular grid is presented in Fig. 1.

3. Results

We compared the proposed reconstruction method with two other well-known methods: (1) *NN direct inversion implemented in SPARX* [13] and (2) *RELION DFR* [12]. These methods were compared in terms of accuracy, memory complexity, and the time required for execution. Two asymmetric test objects were used to achieve this goal: first, a 70S ribosomal subunit as an instance of a small complex and second, a DNA-origami object as a good representative of a large complex. Different data sets (set of projections) with specific properties (i.e., noise in projections, noise in angular assignment, and CTF-affected) were made for each test object to examine the robustness of each method.

To evaluate the accuracy of each reconstruction method, we used the Fourier Shell Correlation (FSC) criterion calculated with respect to the ground truth, which is the standard quality measurement in single particle analysis. All experiments were run on a cluster with 28 nodes, each with two Intel Xeon E5405 running at 2 GHz and with 16 GB of RAM memory.

3.1. 3D reconstruction of the 70S ribosome

In this experiment, we compared the reconstruction results of the proposed method with the results from RELION and SPARX using different data sets that were generated from simulated projections with specific characteristics corresponding to the 70S ribosomal subunit (PDB ID: 3V2D) represented on a cube of size

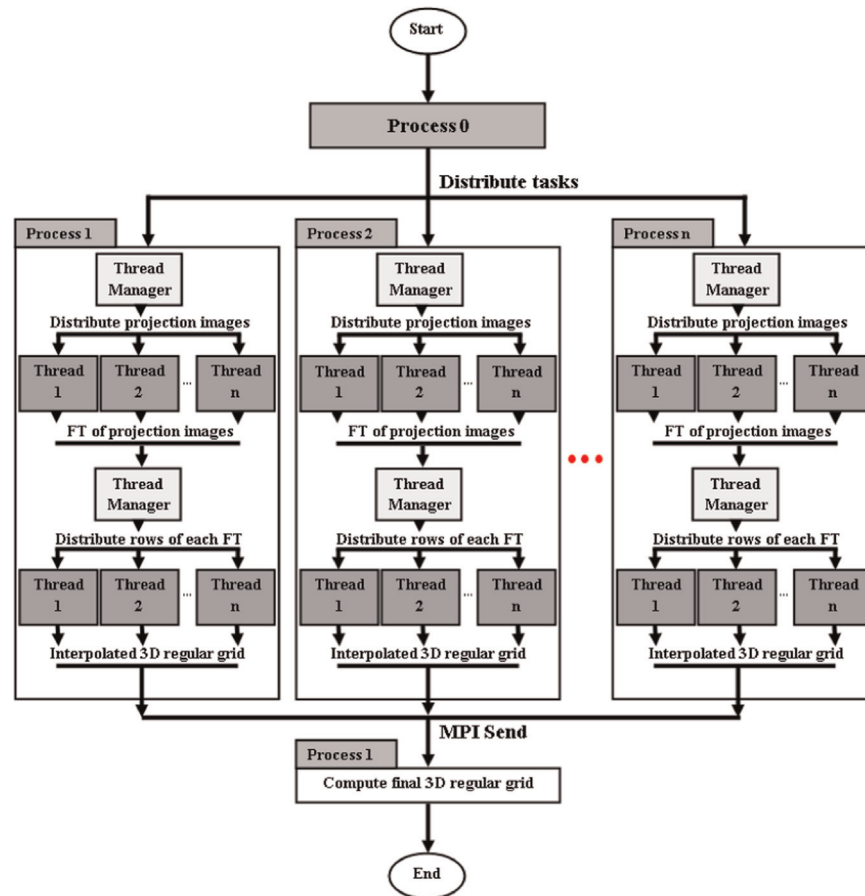


Fig. 1. Parallelization strategies. The two levels of parallelization of the proposed algorithm, including multiple threads and multiple processors are shown. The algorithm can be executed on several nodes of a cluster (multi-CPU parallelism) with several threads (multi-thread parallelism).

$174 \times 174 \times 174$ voxels (the sampling rate was $1.5 \text{ \AA}/\text{pixel}$).

We applied the three reconstruction algorithms (with the exception of the first experiment) to five sets of simulated projections. The first data set is a set of noise-free projections of the test object, the second and third sets comprise projections contaminated by noise in angles and noise in pixel values, respectively, the fourth set contains a set of projections for coarser angular sampling (i.e., when the Crowther frequency is lower than the Nyquist frequency), and the fifth set contains CTF-affected projections.

We used default parameters for each method in these experiments. All methods use a padding factor of 2, and in our tests we set $\varepsilon = 0.01$.

3.1.1. Projections without noise

For this experiment, we applied five reconstruction algorithms to a set of $N=10,000$ free of noise projections: our proposed algorithm, RELION method [12], SPARX method [13], ART, and WBP. In Fig. 2, we show the FSC curves for the five reconstructions compared to the known ground truth (the 70S ribosome subunit at atomic resolution with a sampling rate of $1.5 \text{ \AA}/\text{pixel}$). In this figure, the horizontal axis shows the spatial frequency and the vertical axis shows the FSC values. This figure shows that the RELION method reaches higher FSC values than the SPARX method, while our proposed gridding-based DFR method is slightly more accurate at all frequencies than either method. Although the difference is not large, it proves the validity of our approach. It is obvious from this figure that the RELION, SPARX and XMIPP methods are able to yield larger FSC values for higher resolution than the ART and WBP methods.

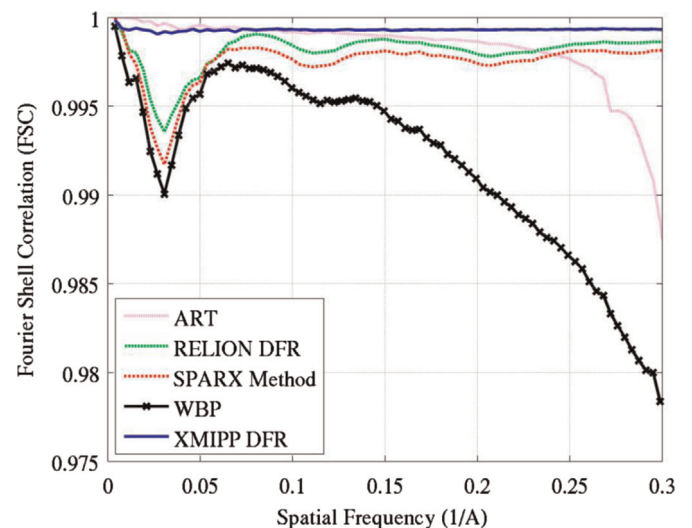


Fig. 2. Comparison of the reconstruction methods using noise-free projections of the 70S ribosome. FSC curves for the 3D reconstructions of the 70S ribosome using a set of 10,000 noise-free projections applying the following reconstruction algorithms: the proposed method (XMIPP), RELION method [12], SPARX method [13], ART and WBP.

For this reconstruction, the serial implementation of our proposed method requires 13 min and 40 s, while it takes 5 min and 37 s for RELION and 20 min and 44 s for SPARX using one core of the CPU. In Fig. 3, we show how increasing the number of threads can decrease the required computational time for our proposed method to one-fourth of the SPARX and half of the RELION

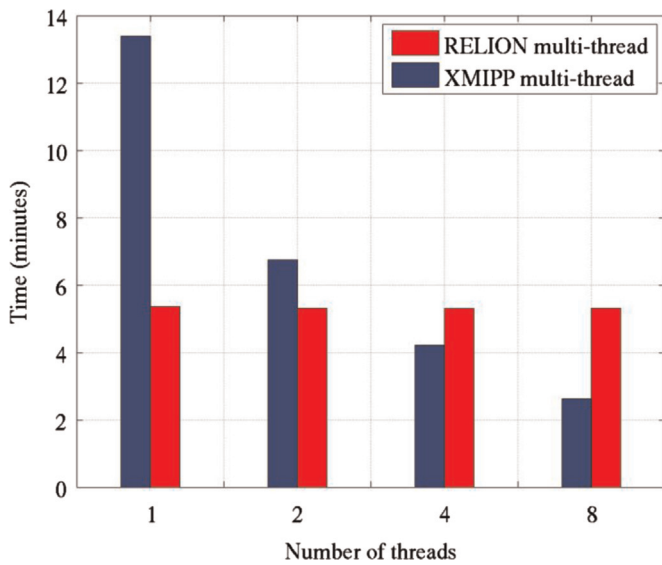


Fig. 3. Multi-thread implementation comparisons. Computational time required for the 3D reconstruction of the 70S ribosome from a set of $N=10,000$ noise-free projections applying the proposed (XMIPP) and RELION reconstructions using different numbers of threads.

reconstruction methods (2 min and 63 s using 8 CPUS). Note that although RELION supports threads, its parallel implementation does not improve the execution time significantly in this case (the threads simply compute the Fourier transform of the projections). As shown in this figure, increasing the number of threads reduces the computational time; however, more than eight threads make no or little improvement due to the required time for the initialization and synchronization of the threads.

In Fig. 4, we show the average execution times of 10 independent runs for the proposed method and the SPARX method (note that RELION implementation does not support multi-CPU parallelization) using different numbers of nodes (cluster computing parallelism). From this figure, it is evident that increasing the number of nodes improves the computational time for both methods. Indeed, increasing the number of nodes by a factor of two increases the speedup by a factor of two (which is ideal). It is obvious from this figure that the parallel implementation of our algorithm is faster than the parallel implementation of SPARX

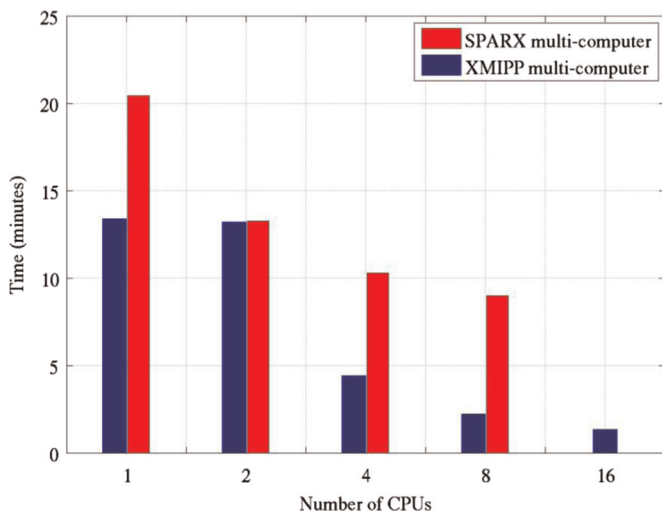


Fig. 4. Multi-computer implementation comparisons. The required computational time for the parallel execution of the proposed reconstruction method and the SPARX method for reconstructing the 70S ribosome using a set of 10,000 noise-free projections utilizing different numbers of CPUs.

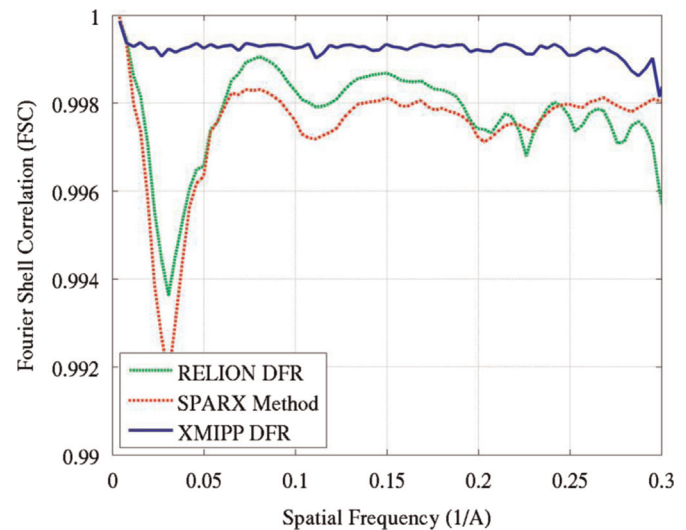


Fig. 5. Comparison of the reconstruction methods using CTF-affected projections of the 70S ribosome. FSC curves for the reconstructions of the 70S ribosome generated from $N=10,000$ projection images affected by 10 different sets of CTF parameters using our proposed reconstruction method (XMIPP), RELION method [12], and SPARX method [13]. The CTF parameters include: physical pixel size of 3.54 Å, microscope voltage of 300 kV, spherical aberration of 2.0 mm, amplitude contrast of 0.1, and different defocus values.

(approximately twice as fast for more than 2 CPUs because our implementation uses one node as a coordinator). Additionally our parallel implementation is scalable, while this does not apply to the parallel implementation of SPARX (increasing the number of CPUs from 4 to 8 provides approximately the same execution time).

3.1.2. CTF applied projections

In this experiment, we modified the entire $N=10,000$ simulated projections using 10 different sets of CTF parameters (each consequent 1000 particles have the same CTF parameters) with a physical pixel size of 3.54 Å, microscope voltage 300 kV, spherical aberration of 2.0 mm, amplitude contrast of 0.1, and different defocus values. We calculated the 3D maps for the three algorithms by enabling the CTF correction. The results for our method (XMIPP), RELION [12] and SPARX [13] with CTF correction are shown in Fig. 5. As seen in this figure, our algorithm can properly correct for the CTF, thereby maintaining the previously demonstrated pattern of providing slightly better results than the other two methods.

3.1.3. Projections with noise

In this test, which is closer to an experimental case, we introduced two different types of noise into the simulated projections to generate two sets of noise-corrupted projections. First, a Gaussian noise with a signal-to-noise ratio of approximately 0.8 was added to the pixels of the noise-free projections. Second, a Gaussian noise with zero mean and variance of 0.5° was applied to each projection direction to take into account angular inaccuracies. For each data set, $N=10,000$ projections were generated.

Similar to the previous experiment, we performed the reconstruction for each data set using our method (XMIPP), RELION [12] and SPARX [13] to evaluate the resilience of each reconstruction algorithm to noise. In Figs. 6 and 7, we compare the reconstruction results of these three methods in two situations: using projection images contaminated by noise in pixel intensities and using projection images contaminated by noise in angular assignments. From this figure, it is clear that there is a region of low frequency at which the FSC values are larger for the proposed

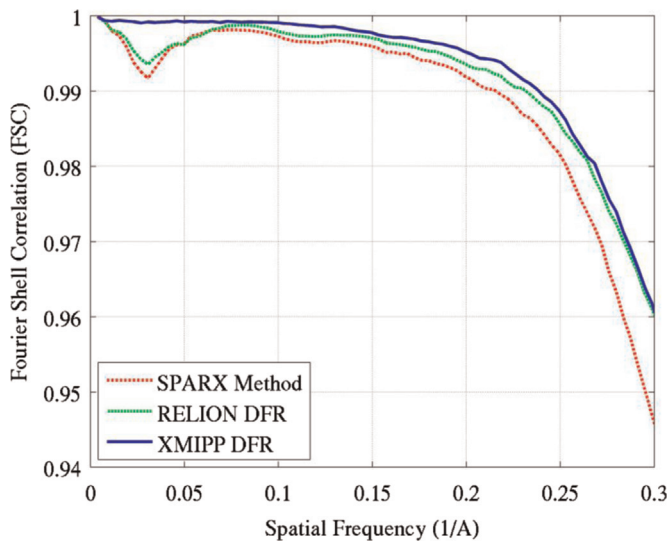


Fig. 6. Comparison of the reconstruction methods using projections of the 70S ribosome contaminated by pixel noise. FSC curves for the reconstructions of the 70S ribosome generated from $N=10,000$ projection images contaminated by pixel noise using our proposed reconstruction method (XMIPP), RELION method [12], and SPARX method [13].

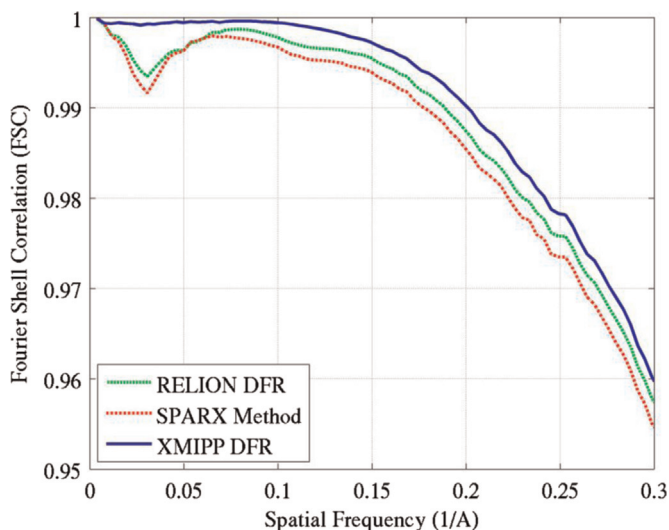


Fig. 7. Comparison of the reconstruction methods using projections of the 70S ribosome contaminated by angular noise. FSC curves for the reconstructions of the 70S ribosome generated from $N=10,000$ projection images contaminated by angular noise using our proposed reconstruction method (XMIPP), RELION method [12], and SPARX method [13].

method; at the same time, the proposed method is significantly faster than the SPARX method and faster than RELION when more than two threads are used.

3.1.4. Projections with coarser angular sampling

In this experiment, we used coarser angular sampling to compare the proposed method with two other methods when larger gaps are introduced into Fourier space. First, we generated $N=500$ evenly distributed projections (according to the Crowther frequency). For simplicity in the analysis, these projections were generated by fixing the azimuthal and in-plane rotation angles and simply changing the tilt angle. Three more datasets with $N=250$, $N=100$, and $N=50$ projections were generated in the same manner, and three reconstruction methods were applied to each data set as in the previous experiments. The obtained FSC curves resulting from each reconstruction method for these three data sets

are shown in Fig. 8. The FSC curves for the data set with 500 projections (Fig. 8a) are close to the curves shown in Fig. 2 (with the FSC valley at a low frequency for SPARX and RELION). However, when we decrease the number of projections to 250 (Fig. 8b), all of the methods show slightly lower values for the higher frequency, although the effect is more noticeable for the SPARX method. Finally, by further decreasing the number of projections to 100 and 50 (as shown in Fig. 8c and d, respectively), we notice that our reconstruction method shows remarkably higher FSC values (especially for higher frequencies) compared with the other two methods (note that RELION uses the same default values for gridding interpolation as the proposed method). These experiments show that our algorithm is more robust to gaps in the Fourier space.

3.2. 3D reconstruction of a 3D DNA-origami object

The purpose of this experiment is to determine how each reconstruction algorithm can manage memory for a large complex. For this purpose, the model of a discrete DNA object [20] (PDB ID: 2YMF) of size 400×400 (sampling rate 1.2 Å/pixel) was considered.

As in the previous experiment, we simulated $N=10,000$ projections at random orientations of this complex and used the proposed method (XMIPP), RELION method [12], and SPARX method [13] for the reconstruction. Due to the large size of this complex, neither the SPARX nor RELION reconstruction algorithms (with one CPU core) nor the parallel implementation of the proposed algorithm can be executed on our machine (with 16 gigabytes of RAM memory). However, we can take advantage of the multi-thread implementation of our algorithm to use the process resources efficiently without a lack of memory.

To accomplish this reconstruction, our method requires 1 h and 29 min with only one CPU core. Using our multi-threaded implementation, the execution times for different number of threads were computed as: 42 min and 72 s for two threads, 24 min and 83 s for four threads, and 15 min and 48 s for eight threads. Each execution time was obtained using the average of ten independent runs.

4. Conclusions

In this paper, we introduced a weighting function for gridding-based DFR in SPA based on the method published for PET by Matej and Lewitt [11] to discard the artifact connected with the non-uniformity of samples in the Fourier space. This accurate and fast weighting function corrects the values of frequency samples on a 3D regular grid to compensate for the uneven distribution of projection images before applying the final inverse FFT to recover the original 3D object. The CTF correction is performed during the reconstruction to correct for the effects of the CTF. The implementation of the proposed algorithm supports two parallelization frameworks: multi-thread parallelization and multi-computer parallelization. These parallelization frameworks allow our approach to efficiently take advantage of all cores of a node and nodes of a cluster to more quickly produce the final 3D reconstruction. When multi-computers run out of memory due to the large size of the complex (this happens when multiple MPI processes attempt to run on the same node because memory needs are multiplied by the number of processes), multi-thread implementation can still be employed to take advantage of multiple processing resources (here, the memory requirements, regardless of the number of threads, are the same as the requirements for a single MPI process). We compared our gridding-based DFR with two well-known methods in the field (the method of

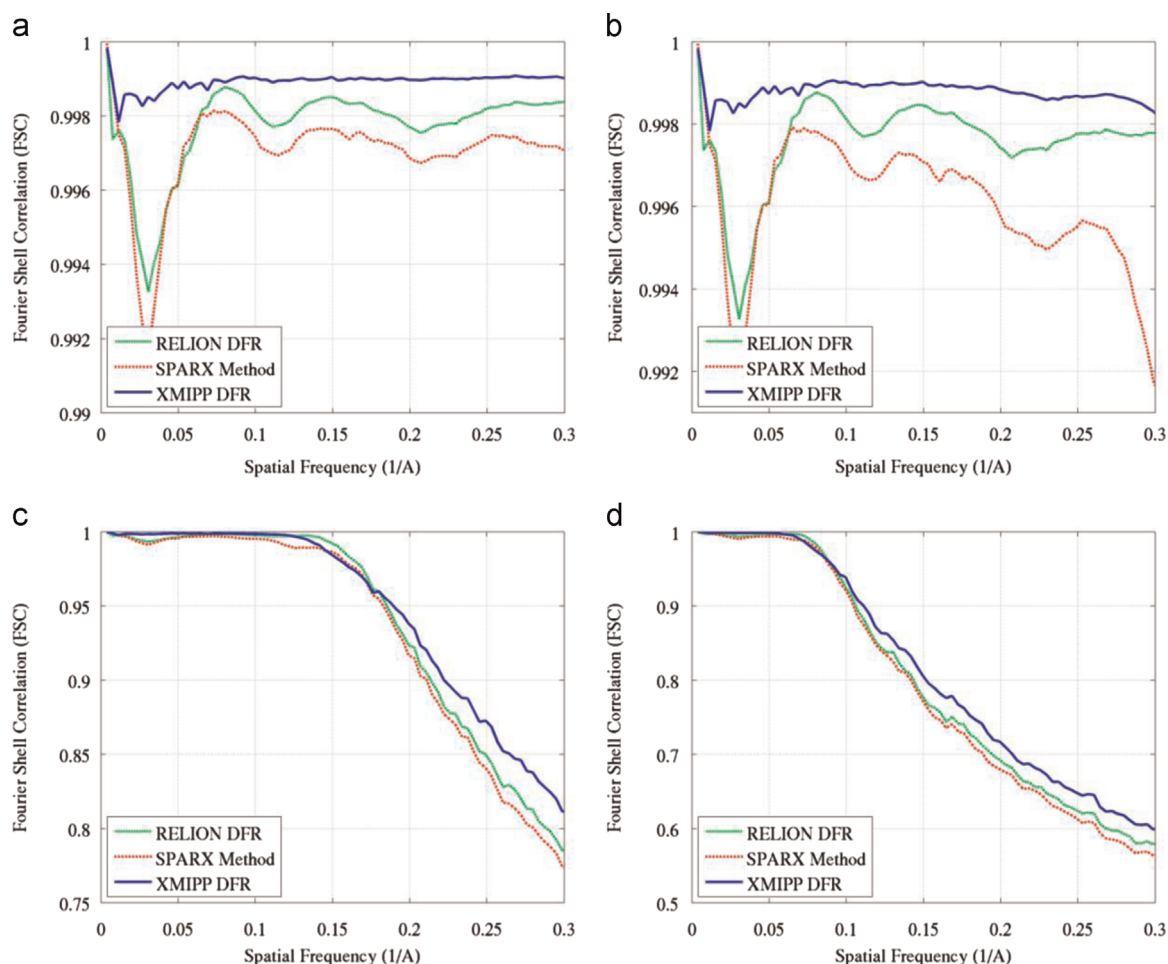


Fig. 8. Comparison of the reconstruction methods using a set of projections with larger angular sampling of the 70S ribosome. FSC curves for the reconstructions of the 70S ribosome generated from $N=500$ (a), $N=250$ (b), $N=100$ (c), and $N=50$ (d) projection images generated by an even angular sampling of the object (fixing the azimuthal and in-plane rotation angles for simplicity) using the proposed method (XMIPP), RELION method [12], and SPARX method [13].

Zhang et al. [13] implemented in SPARX and the RELION reconstruction method [12]) using data sets (sets of projections) of two test objects: the 70S ribosome as a small complex and a DNA-origami object as a large complex. The FSC to the known ground truth was used to compare the reconstruction results of these methods. The proposed method achieved slightly higher FSC values compared with the two other approaches for both the noise-free and noise-contaminated projections. However, the proposed method showed significantly higher values (especially for higher frequencies) compared to the other two methods for data sets with a coarser angular sampling (i.e., when the Crowther frequency was lower than the Nyquist frequency), which confirmed that the proposed approach could achieve more accurate reconstructions when there was a large gap in the Fourier space; this property should be further analyzed for its use in Tomography. Comparing the execution time of each algorithm for reconstructing the 70S ribosome shows that our algorithm is approximately twice as fast as RELION and SPARX when more than two CPUs are used. For the 3D DNA-origami object with a large size, the SPARX and RELION reconstruction methods ran out of memory in our machine with 16 GB of memory, while our multi-threaded implementation could improve the reconstruction speed by efficiently using processing resources. This lower memory requirement allows the proposed algorithm to work with larger volumes, which is in demand in the field.

This algorithm is included in Xmipp 3.1 under the name `reconstruct_fourier` and is downloadable from <http://xmipp.cnb.csic.es>.

The program is accessible through the protocols described by [21].

Acknowledgments

This work was funded by Instruct, part of the European Strategy Forum on Research Infrastructures (ESFRI) and supported by national member subscriptions. The authors would like to acknowledge economic support from the Spanish Ministry of Economy and Competitiveness through Grants AIC-A-2011-0638, BIO2013-44647-R, TIN2012-37483-C03, and TIN2011-15734-E, the Comunidad de Madrid through Grant CAM (S2010/BMD-2305), and a postdoctoral “Juan de la Cierva” grant with reference JCI-2011-10185 to Javier Vargas. Vahid Abrishami is a holder of the “La Caixa” scholarship, and both C.O.S. Sorzano and J.R. Bilbao-Castro are recipients of the “Ramón y Cajal” fellowship.

References

- [1] P.A. Penczek, Chapter one-fundamentals of three-dimensional reconstruction from projections, in: J.J. Grant (Ed.), *Methods in Enzymology*, Academic Press, 2010, pp. 1–33.
- [2] R. Gordon, R. Bender, G.T. Herman, Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography, *J. Theor. Biol.* 29 (1970) 471–481.
- [3] P. Gilbert, Iterative methods for the three-dimensional reconstruction of an object from projections, *J. Theor. Biol.* 36 (1972) 105–117.

- [4] C.O.S. Sorzano, J.A. Velázquez-Muriel, R. Marabini, G.T. Herman, J.M. Carazo, Volumetric restrictions in single particle 3DEM reconstruction, *Pattern Recognition* 41 (2008) 616–626.
- [5] R. Marabini, G.T. Herman, J.M. Carazo, 3D reconstruction in electron microscopy using ART with smooth spherically symmetric volume elements (blobs), *Ultramicroscopy* 72 (1998) 53–65.
- [6] G. Harauz, M. van Heel, Exact filters for general geometry three dimensional reconstruction, in: *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, 1986, pp. 146–156.
- [7] M. Radermacher, Weighted Back-projection methods, in: J. Frank (Ed.), *Electron Tomography*, Plenum, New York, 1992, pp. 91–115.
- [8] C.O.S. Sorzano, R. Marabini, N. Boisset, E. Rietzel, R. Schröder, G.T. Herman, J.M. Carazo, The effect of overabundant projection directions on 3D reconstruction algorithms, *J. Struct. Biol.* 133 (2001) 108–118.
- [9] P.A. Penczek, R. Renka, H. Schomberg, Gridding-based direct Fourier inversion of the three-dimensional ray transform, *J. Opt. Soc. Am. A* 21 (2004) 499–509.
- [10] J.I. Jackson, C.H. Meyer, D.G. Nishimura, A. Macovski, Selection of a convolution function for Fourier inversion using gridding [computerised tomography application], *IEEE Trans. Med. Imaging* 10 (1991) 473–478.
- [11] S. Matej, R.M. Lewitt, 3d-FRP: direct Fourier reconstruction with Fourier re-projection for fully 3-D PET, *IEEE Trans. Nucl. Sci.* 48 (2001) 1378–1385.
- [12] S.H. Scheres, RELION: implementation of a Bayesian approach to cryo-EM structure determination, *J. Struct. Biol.* 180 (2012) 519–530.
- [13] W. Zhang, M. Kimmel, C.M.T. Spahn, P.A. Penczek, Heterogeneity of large macromolecular complexes revealed by 3D Cryo-EM variance analysis, *Structure* 16 (2008) 1770–1776.
- [14] R.N. Bracewell, A. Riddle, Inversion of fan-beam scans in radio astronomy, *Astrophys. J.* 150 (1967) 427.
- [15] H. Schomberg, J. Timmer, The gridding method for image reconstruction by Fourier transformation, *IEEE Trans. Med. Imaging* 14 (1995) 596–607.
- [16] S. Matej, R.M. Lewitt, Efficient 3D grids for image reconstruction using spherically-symmetric volume elements, in: *Nuclear Science Symposium and Medical Imaging Conference*, 1994 IEEE Conference Record, vol. 1173, 1994, pp. 1177–1181.
- [17] E.H. Meijering, W.J. Niessen, M.A. Viergever, Quantitative evaluation of convolution-based methods for medical image interpolation, *Med. Image Anal.* 5 (2001) 111–126.
- [18] E. Gabriel, G. Fagg, G. Bosilca, T. Angskun, J. Dongarra, J. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. Castain, D. Daniel, R. Graham, T. Woodall, Open (MPI): Goals, Concept, and Design of a Next Generation (MPI) Implementation, in: *Proceedings of the 11th European PVM/MPI Users' Group Meeting*, 2004, pp. 97–104.
- [19] F. Mueller, Pthreads Library Interface, in, 1994.
- [20] X.-c. Bai, T.G. Martin, S.H. Scheres, H. Dietz, Cryo-EM structure of a 3D DNA-origami object, *Proc. Natl. Acad. Sci.* 109 (2012) 20012–20017.
- [21] J.M. de la Rosa-Trevín, J. Otón, R. Marabini, A. Zaldívar, J. Vargas, J.M. Carazo, C.O.S. Sorzano, Xmipp 3.0: An improved software suite for image processing in electron microscopy, *J. Struct. Biol.* 184 (2013) 321–328.