Single-particle reconstruction using L^2-gradient flow

Ming Li [a,b], Guoliang Xu [a,*], Carlos O.S. Sorzano [c], Fei Sun [b], Chandrajit L. Bajaj [d]

[a] LSEC, Institute of Computational Mathematics and Scientific/Engineering, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
[b] National Laboratory of Biomacromolecules, Institute of Biophysics, Chinese Academy of Sciences, Beijing, China
[c] Biocomputing Unit, National Center of Biotechnology (CSIC), Cantoblanco, Madrid, Spain
[d] Department of Computer Sciences and Institute of Computational Engineering & Sciences, University of Texas at Austin, Austin, TX, USA

ABSTRACT

In this paper, we present an iterative algorithm for reconstructing a three-dimensional density function from a set of two dimensional electron microscopy images. By minimizing an energy functional consisting of a fidelity term and a regularization term, an L^2-gradient flow is derived. The flow is integrated by a finite element method in the spatial direction and an explicit Euler scheme in the temporal direction. Our method compares favorably with those of the weighted back projection, Fourier method, algebraic reconstruction technique and simultaneous iterative reconstruction technique.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

In the past 10 years, cryo-electron microscope imaging techniques have become indispensable tools for determining the three-dimensional structure of large macromolecules and biological machineries. Different from X-ray crystallographic methods, electron microscopy does not require crystallization. Specimens are applied to a grid covered with holey carbon, and then are rapidly plunged into a cryogen-usually liquid ethane cooled by liquid nitrogen (Frank, 2006). This data acquisition process can be very fast. Hence, these methods provide good opportunities to determine the three-dimensional structures of large macromolecules that are either too large or too heterogeneous to be investigated by X-ray methods (van Heel et al., 2000). Cryo-electron microscopy (Cryo-EM) single-particle analysis (van Heel et al., 2000; Frank, 2006; Zhang et al., 2010) and cryo-electron tomography are the two main techniques in this field. The process of single-particle analysis includes several steps including image classification, orientation and three-dimensional reconstruction. These steps can be integrated into an iterative refinement using projection matching (Penczek et al., 1994; Radermacher, 1994; Sorzano et al., 2004a). Other steps include corrections for the contrast transfer function (CTF) and amplitude in order to obtain higher resolutions. In this paper, we concentrate on the three-dimensional reconstruction step of single-particle analysis.

The most common reconstruction algorithms in single-particle analysis are weighted back projection (WBP) (Radermacher, 2006), Fourier reconstruction (DeRosier and Klug, 1968; Crowther et al., 1970; Natterer and Wübbeling, 2001; Matej and Lewitt, 2001), and iterative methods including algebraic reconstruction technique (ART) (Gordon et al., 1970) and simultaneous iterative reconstruction technique (SIRT) (Gilbert, 1972). Other methods include simultaneous algebraic reconstruction technique (SART) (Andersen and Kak, 1984) and block ART (Marabini et al., 1998). For limited angle tomography, such as electron tomography (ET), regularization methods have been applied for solving the ill-posed problem. SIRT with regularization through early stopping has been applied to ET (Schoenmakers et al., 2005; Voorhout et al., 2006). Projection onto convex sets (POCS) assumes that the object f belongs to the intersection of some closed convex sets (Youla and Webb, 1982; Sezan and Stark, 1982; Carazo, 1992). Variational regularization methods employ different regularization terms such as total variation (TV) regularization (Aganj et al., 2007) and regularization using the second partial derivatives (Kybic et al., 2001; Kybic et al., 2002; Do et al., 2010).

In this paper we propose an L^2-gradient flow three-dimensional reconstruction algorithm of single-particle analysis for solving a variational model with TV regularization term. The object is represented with a function f in the tri-cubic B-spline function space. One of the advantages of using cubic B-spline functions is that we can obtain a C^2 smooth object (Unser, 1999). The compact support property of B-spline basis can be employed to accelerate the reconstruction process. The L^2-gradient flow is derived based on the first-order variation (Giaquinta and Hildebrandt, 1996) of the

* Corresponding author. Fax: +86 10 6254 2285.
  E-mail address: xuguo@lsec.cc.ac.cn (G. Xu).

energy model. An explicit Euler scheme and the finite element method are employed to yield an iteration scheme. To avoid solving large systems of linear equations, we use the orthogonal basis of tri-cubic B-spline basis functions derived from Schmidt orthogonal process. Further, we derive a temporal step-size formula which can automatically give a proper time step-size at every iteration step.

Currently, the Fourier method and WBP produce the best results for input data with good angular coverage. Our algorithm outperforms these approaches thanks to the incorporation of a priori knowledge through the regularization, and comparing with the prior TV regularization work (Kybic et al., 2001, 2002), our algorithm does not solve a linear system of equations and can be used for large number of measures.

2. Methods

To reconstruct an object in the temporal domain, we minimize the following energy functional

$$J(f) = J_1(f) + \lambda J_2(f), \tag{1}$$

where

$$J_1(f) = \sum_{\mathbf{d} \in \mathbb{D}} \int_{\mathbb{R}^2} (X_{\mathbf{d}} f - g_{\mathbf{d}})^2 \, d\mathbf{u},$$

$$J_2(f) = \int_{\mathbb{R}^3} \|\nabla f\| \, d\mathbf{x}.$$

The tri-variable function f is the unknown object to be reconstructed. Let $\mathbf{u} = [u, v]^T, \mathbf{x} = [x, y, z]^T$. Let $S^2 \subset \mathbb{R}^3$ denote the unit sphere. $X_{\mathbf{d}}$ stands for a projection of f in the direction $\mathbf{d} \in \mathbb{D} \subset S^2$. $\{g_{\mathbf{d}}\}$ is a set of two-dimensional projections measured from the electron microscope. The first term in the modes (1) measures the fidelity to the projection data. The second one is a TV regularization term which plays the role of smoothing. The parameter λ is a nonnegative weighting constant. Our goal is to search for a function f that makes the projection $X_{\mathbf{d}} f$ best fit the data $g_{\mathbf{d}}$ so that its total variation is low.

2.1. L^2-gradient flow

We want to minimize $J(f)$ by adjusting f. This goal is achieved by solving an L^2-gradient flow of $J(f)$ in the B-spline space $\mathcal{X}_B$ (see B.1). It is easy to derive the following first-order variations (see B.2 and B.3)

$$\delta(J_1(f), \psi) = 2 \sum_{\mathbf{d} \in \mathbb{D}} \int_{\mathbb{R}^2} (X_{\mathbf{d}} f - g_{\mathbf{d}}) X_{\mathbf{d}} \psi \, d\mathbf{u}, \tag{2}$$

$$\delta(J_2(f), \psi) = \int_{\mathbb{R}^3} \frac{(\nabla f)^T \nabla \psi}{\|\nabla f\|} \, d\mathbf{x}, \tag{3}$$

where ψ is a test function in the function space $\mathcal{X}_B$. Using these first-order variations, we construct the following weak-form L^2-gradient flow (Xu, 2008)

$$\int_{\mathbb{R}^3} \frac{\partial f}{\partial t} \psi \, d\mathbf{x} + \delta(J_1(f), \psi) + \lambda \delta(J_2(f), \psi) = 0. \tag{4}$$

The function $f \in \mathcal{X}_B$ is represented as

$$f(x, y, z) = \sum_{i,j,k} f_{ijk} N_i^3(x) N_j^3(y) N_k^3(z), \sum_{i,j,k} \tilde{f}_{ijk} \tilde{N}_i^3(x) \tilde{N}_j^3(y) \tilde{N}_k^3(z). \tag{5}$$

where N_α^3 ($\alpha = i, j, k$) is a cubic B-spline basis function defined on the interval $[-2 + \alpha, 2 + \alpha]$ (see A.1), and f_{ijk} is the coefficient of tri-cubic B-spline tensor product $N_i^3(x) N_j^3(y) N_k^3(z)$. $\tilde{N}_\alpha^3 (\alpha = i, j, k)$ is the orthogonal tri-cubic B-spline basis function and $\tilde{f}_{ijk}$ is the coefficient of orthogonal tri-cubic B-spline tensor product $\tilde{N}_i^3(x) \tilde{N}_j^3(y) \tilde{N}_k^3(z)$. There

is a lower-triangular matrix A connecting the two sets of basis functions,

$$[N_0^3, N_1^3, \ldots, N_n^3]^T = A[\tilde{N}_0^3, \tilde{N}_1^3, \ldots, \tilde{N}_n^3]^T, \tag{6}$$

where

$$A = \begin{bmatrix} a_{00} & & & & \\ a_{10} & a_{11} & & & \\ a_{20} & a_{21} & a_{22} & & \\ a_{30} & a_{31} & a_{32} & a_{33} & \\ & \ddots & \ddots & \ddots & \ddots \\ & & a_{n,n-3} & a_{n,n-2} & a_{n,n-1} & a_{n,n} \end{bmatrix}.$$

Matrix A can be calculated using Eq. (B.10)–(B.14).

Suppose that the size of the measured images is $n \times n$, where n is an integer number. The pixel values of each image $g_{\mathbf{d}}$ are defined on the coordinate points $(i,j)^T \in \Omega_0 = [-[n/2], -[n/2] + n - 1]^2$, where $i, j = -[n/2], -[n/2] + 1, \cdots, -[n/2] + n - 1$. Since $n > 0$, $[x]$ means taking the largest integer value that is not greater than x. The domain of function f is defined as $\Omega = [-[n/2], -[n/2] + n - 1]^3$ and the dimension of reconstructed volume in each coordinate direction is n. We define the cubic B-spline basis function on the same coordinate points as the image. The number of cubic B-spline basis functions in each coordinate direction is $n - 4$, making the total number of tri-cubic B-spline tensor products amounting to $(n - 4)^3$. Considering the one-dimensional case, for example $n = 10$ or 11, we take 6 or 7 cubic B-spline basis functions, respectively. They are shown in Fig. 1. The function f represented by these cubic B-spline basis functions vanishes at the boundary of the domain Ω.

The projection direction $\mathbf{d}$ can be represented by three Euler angles $(\theta_1, \theta_2, \theta_3)$. In this paper, we use the z–y–z convention for the definition of Euler angles. The first angle θ_1 defines a rotation matrix R_{θ_1} around the z-axis. θ_2 defines a rotation matrix R_{θ_2} around the new y axis and R_{θ_3} is the third rotation around the new z-axis. The rotation matrix R corresponding to the projection direction $\mathbf{d}$ is defined by

$$R = R_{\theta_3} R_{\theta_2} R_{\theta_1}.$$

Denote $R = [\mathbf{e}_{\mathbf{d}}^{(1)}, \mathbf{e}_{\mathbf{d}}^{(2)}, \mathbf{d}]$, where $\mathbf{e}_{\mathbf{d}}^{(1)}, \mathbf{e}_{\mathbf{d}}^{(2)}$ and $\mathbf{d}$ are the final coordinate system after three rotations. Hence, the projection of f along the direction $\mathbf{d}$ is defined by

$$(X_{\mathbf{d}} f)(i, j) = \int_{-\infty}^{\infty} f(i\mathbf{e}_{\mathbf{d}}^{(1)} + j\mathbf{e}_{\mathbf{d}}^{(2)} + t\mathbf{d}) \, dt, \quad (i, j)^T \in \Omega_0. \tag{7}$$

2.2. Iteration scheme

Employing an explicit Euler scheme in the temporal direction and the finite element method in the spatial direction with orthogonal tri-cubic B-spline basis functions, we obtain a linear system of equations (see Appendix B Discretization)

$$\tilde{X} = \tilde{B} \tag{8}$$

with $\tilde{X} = [\tilde{f}_{i'j'k'}^{(m+1)}]^T$ unknown. Therefore, solving these equations is reduced to the calculation of the vector $\tilde{B} = [\tilde{B}_{i'j'k'}]^T$. Given $i' j'$, k',

$$\tilde{B}_{i'j'k'} = \int_{\Omega} f^{(m)} \tilde{\phi}_{i'j'k'} \, d\mathbf{x} - \tau \delta(J_1(f^{(m)}), \tilde{\phi}_{i'j'k'}) - \tau \lambda \delta(J_2(f^{(m)}), \tilde{\phi}_{i'j'k'}), \tag{9}$$

where $\tilde{\phi}_{i'j'k'} = \tilde{N}_{i'}^3(x) \tilde{N}_{j'}^3(y) \tilde{N}_{k'}^3(z)$. Using (8), (9), (B.2) and (B.3), we can derive the following iteration scheme:

$$\tilde{f}_{i'j'k'}^{(m+1)} = \tilde{f}_{i'j'k'}^{(m)} - 2\tau \sum_{\mathbf{d} \in \mathbb{D}} \int_{\Omega_0} (X_{\mathbf{d}} f^{(m)} - g_{\mathbf{d}}) X_{\mathbf{d}} \tilde{\phi}_{i'j'k'} \, d\mathbf{u} - \tau \lambda \int_{\Omega}$$

$$\times \frac{(\nabla f^{(m)})^T \nabla \tilde{\phi}_{i'j'k'}}{\|\nabla f^{(m)}\|} \, d\mathbf{x} \tag{10}$$
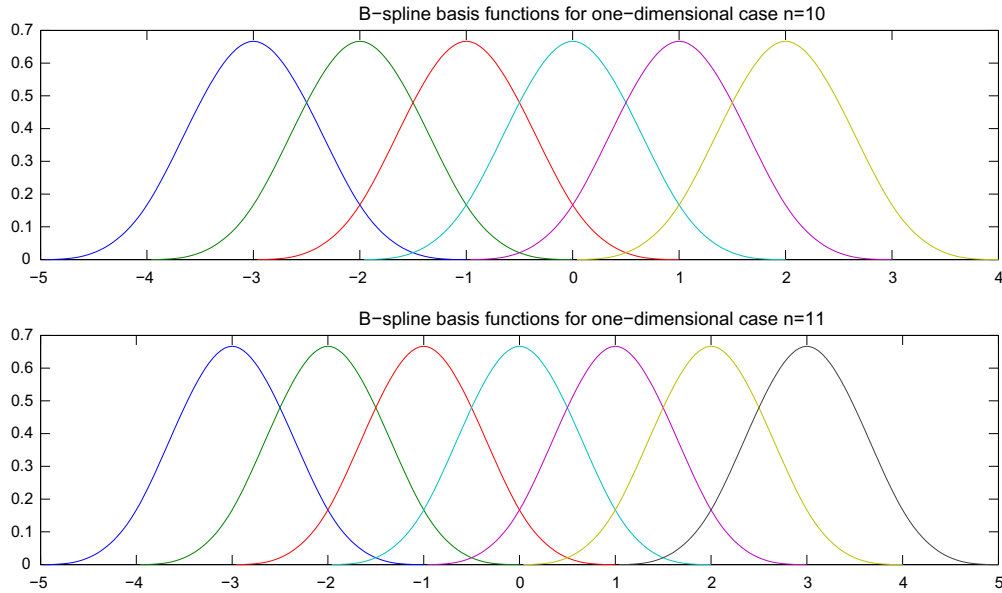
Fig. 1. The cubic B-spline basis functions in one-dimensional case. The top figure shows six B-spline basis functions for $n = 10$. The bottom figure shows seven B-spline basis functions for $n = 11$.

where $\tilde{f}^{(m)}_{i'j'k'}$ ($i',j',k' = -[n/2]+2, \cdots, -[n/2]+n-3$) is the coefficient of orthogonal tri-cubic B-spline tensor product $\tilde{\phi}_{i'j'k'}$ at the iteration step m. Trapezoidal integration formula is used to calculate these integrals in the above iteration scheme. To avoid singularity when $\|\nabla f^{(m)}\| = 0$, we replace $\|\nabla f^{(m)}\|$ with $\epsilon + \|\nabla f^{(m)}\|$ in the iteration scheme (10), where ϵ is a small positive number. In this paper, we take $\epsilon = 10^{-5}$. The gradient of function $f^{(m)}$ and $\tilde{\phi}_{i'j'k'}$ can be calculated accurately using formula (5) and (B.5), respectively. Choosing a suitable time step-length τ is crucial to efficiently reconstruct the function. A step-length which is too small will slow down the evolution process, while on the contrary using a step-length which is too large has the danger of causing the evolution blow up. In the appendix, we derive a formula to compute an optimal time step-length (see C.1 and C.2).

2.3. Final algorithm:L2GF

Based on the above L^2-gradient flow and iteration scheme (10), we give the following reconstruction algorithm named L2GF.

Algorithm 1. L2GF.

Input:
$\{g_{\mathbf{d}} : \mathbf{d} \in \mathbb{D}\}$: Projection images with $\mathbf{d} \in \mathbb{D}$ known.
$f^{(0)}$: Initial function. In this paper $f^{(0)}$ is set to be zero.
$K_{\max}$: The maximum number of iterations.
Step 1: Convert $f^{(0)}$ into orthogonal tri-cubic B-spline
  coefficients $\tilde{f}^{(0)}_{ijk}$ using Eq. (5) and (6).
Step 2: For $m = 0,1,\ldots,K_{\max}-1$
  Compute time step-length τ using (C.2).
  Obtain $\tilde{f}^{(m+1)}_{ijk}$ using the iteration scheme (10).
  Convert $\tilde{f}^{(m+1)}_{ijk}$ to $f^{(m+1)}_{ijk}$ using Eq. (5) and (6).
  Compute $f^{(m+1)}$ from $f^{(m+1)}_{ijk}$ using Eq. (5).
Step 3: Output reconstructed density map from $f^{(K_{\max})}$.

The largest cost in Algorithm 1 comes from computing the term $\delta(J_1(f^{(m)}), \tilde{\phi}_{i'j'k'})$, in which we need compute all the tri-cubic B-spline basis functions projections $\{X_{\mathbf{d}}\phi_{ijk}\}$. Given a projection direction $\mathbf{d}$, we first compute the projection of $X_{\mathbf{d}}\phi_{000}$ using rectangular numer-

ical integral. The other tri-cubic B-spine basis functions projections can be obtained from $X_{\mathbf{d}}\phi_{000}$ by shifting and interpolating. That is, we have the following formula (see (B.9)):

$$(X_{\mathbf{d}}\phi_{ijk})(u,v) = (X_{\mathbf{d}}\phi_{000})\left(u - [i,j,k]\mathbf{e}^{(1)}_{\mathbf{d}}, v - [i,j,k]\mathbf{e}^{(2)}_{\mathbf{d}}\right). \quad (11)$$

Since (see (B.8))

$$\delta(J_1(f^{(m)}), \tilde{\phi}_{i'j'k'}) = 2\sum_{\mathbf{d}\in\mathbb{D}}\int_{\Omega_0}\left(\sum_{i,j,k}f^{(m)}_{ijk}X_{\mathbf{d}}\phi_{ijk} - g_{\mathbf{d}}\right)X_{\mathbf{d}}\tilde{\phi}_{i'j'k'}\,d\mathbf{u}, \quad (12)$$

we have the following fast algorithm for computing $\delta(J_1(f^{(m)}), \tilde{\phi}_{i'j'k'})$.

Algorithm 2. Fast computation of $\delta(J_1(f^{(m)}), \tilde{\phi}_{i'j'k'})$.

Input:
$f^{(m)}$: Function at the iteration number m.
$\{\mathbf{d}\in\mathbb{D}\}$: The set of projection directions.
Step 1: $\forall \mathbf{d}\in\mathbb{D}$
  Compute $X_{\mathbf{d}}\phi_{000}$ using Eq. (7).
  Compute $X_{\mathbf{d}}\phi_{ijk}$ for all i,j,k using Eq. (11).
Step 2: Convert $X_{\mathbf{d}}\phi_{ijk}$ to $X_{\mathbf{d}}\tilde{\phi}_{i'j'k'}$ using matrix A in Eq. (6).
Step 3: Compute $\delta(J_1(f^{(m)}), \tilde{\phi}_{i'j'k'})$ using Eq. (12).
  The computational complexity of L2GF for one
  iteration step is $O(N_d n^3)$ (see Appendix B for details) where
  N_d is the number of projections in $\mathbb{D}$.

---

3. Results

3.1. Simulated data:GroEL

To validate our algorithm, we first characterize the L2GF algorithm under different conditions. In this test, all simulated noisy projections are generated using Xmipp software (Sorzano et al., 2004b).

We first perform the L2GF algorithm under different parameter λ with 60 iterations. We choose a file 1J4Z.pdb from Protein Data Bank (PDB), blur it to 10Å with a sampling rate 1.6 Å and produce 1000 projections with the size 120×120. Gaussian noise with a

signal-to-noise ratio (SNR) of 0.3 is added to these projections. To evaluate the reconstructed results, we use the following energy, error and Fourier shell correlation (FSC) formulas

$$\text{Energy}(f^{(m)}) = \frac{J_1(f^{(m)})}{N_d}, \tag{13}$$

$$\text{Error}(f^{(m)}) = \frac{\sum_{i,j,k}[f^{(m)}(i,j,k) - \text{TrueMap}(i,j,k)]^2}{\sum_{i,j,k}[\text{TrueMap}(i,j,k)]^2}, \tag{14}$$

$$\text{FSC}(r) = \frac{\sum_{r_i \in r} F_1(r_i) \cdot \overline{F}_2(r_i)}{\sqrt{\sum_{r_i \in r}|F_1(r_i)|^2 \cdot \sum_{r_i \in r}|F_2(r_i)|^2}}, \tag{15}$$

where 'TrueMap' stands for the true volume obtained from PDB file. F_1 and F_2 are the Fourier transforms of the density maps f_1 and f_2. $\overline{F}_2$ is the complex conjugate of F_2. We first perform two L2GF iterations with $\lambda = 0$, then for the rest of the iterations, different choice of λ is involved. Fig. 2 shows the energy curves calculated using Eq. (13) for different values of λ. From Fig. 2 we can see that all the energy curves decrease monotonously, and for larger λ, the energy curve decreases more slowly. Secondly, we compare the errors using Eq. (14) for different choices of λ. We can see from Fig. 3 that for $0 \leqslant \lambda \leqslant 1.0$ the errors decrease at the first several iterations and then increase gradually. This is because small values of λ weaken the smoothing effect of J_2, and the noise leads to the increased error. For $\lambda = 5$, 10 and 25, the error curves monotonously decrease. We can conclude that for low SNR data, such as SNR = 0.3, a big value of λ is needed for the L2GF algorithm to obtain a smooth solution. In addition, we compare the resolution using FSC (15) between the reconstructed map and the 10 Å map (blurred from 1J4Z.pdb) at 0.5 cutoff for different choices of λ. At the first iteration, the FSC resolution has a big value, and then falls off quickly at the second iteration. From Fig. 4 we can see that for big values of λ such as 5, 10 and 25, the values of resolution decrease with increasing iterations which means that the use of J_2 with a reasonable value of λ can improve the resolution of reconstruction.

In order to investigate the impact of noise on the reconstruction, we add Gaussian noise to the 1000 projections with different SNR: 0.01, 0.1, 1.0, and 10.0. For each set of projections, we do reconstruction using Xmipp's WBP program 'xmipp_reconstruct_wbp' (Radermacher, 2006), Fourier program 'xmipp_reconstruct_fourier' (Matej and Lewitt, 2001), block ART and SIRT program 'xmipp_reconstruct_art' (Marabini et al., 1998; Gilbert, 1972), and our L2GF program. The threshold for WBP, relaxation parameters for block ART and SIRT, and L2GF's parameter λ are well chosen to obtain desirable resolutions. The reader is referred to (Sorzano

Fig. 3. The L2GF error curves calculated from Eq. (14) for different value of λ with 60 iterations using 1000 projections (SNR = 0.3).
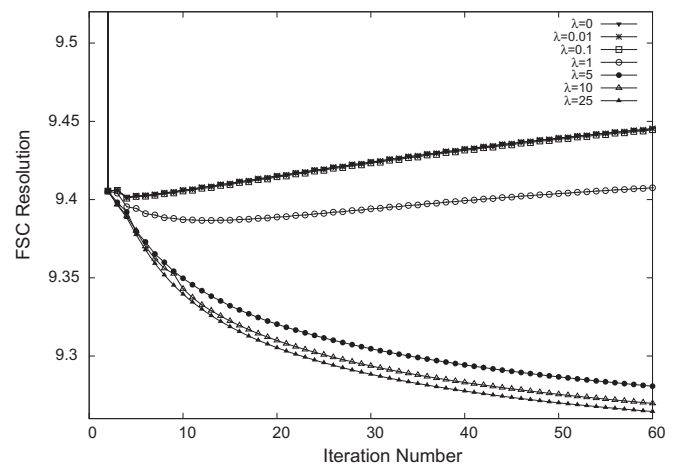
Fig. 4. The L2GF FSC resolution curves for different value of λ with 60 iterations using 1000 projections (SNR = 0.3).

et al., 2001, 2005) for a detailed discussion on the relaxation parameters of block ART and SIRT. The resolution is computed using FSC between the reconstructed map and the 10Å true map (blurred from 1J4Z.pdb) at 0.5 cutoff. For block ART, one iteration is enough to obtain the best result (Marabini et al., 1998). For SIRT and L2GF, the iteration numbers are chosen to obtain stable resolutions. The resolutions, parameters and iteration numbers are contained in Table 1. From Table 1 we can see that L2GF has significant advantages for low SNR's data over the other four methods. And for lower SNRs, L2GF needs a higher value of λ. Fig. 5 shows five slices corresponding to each reconstruction result of the five methods for projections with SNR = 0.01. All the maps have been filtered at the resolution given by the FSC. The iteration numbers for SIRT and L2GF are chosen to obtain stable resolutions. In this test, the L2GF algorithm is performed with $\lambda = 0$ for the first 10 iterations, and $\lambda \neq 0$ for the rest of iterations.

Next, we perform the L2GF algorithm and the other four methods with different number of projections 1000, 10,000, 50,000 and 100,000 under the same SNR of 0.3. The 1J4Z.pdb is blurred to a map with a resolution 3 Å and a sampling rate 1.0 Å. Each set of projections with the size 181×181 are produced using random projection directions. For each set of projections, we reconstruct the map using Xmipp's programs including WBP, Fourier, block
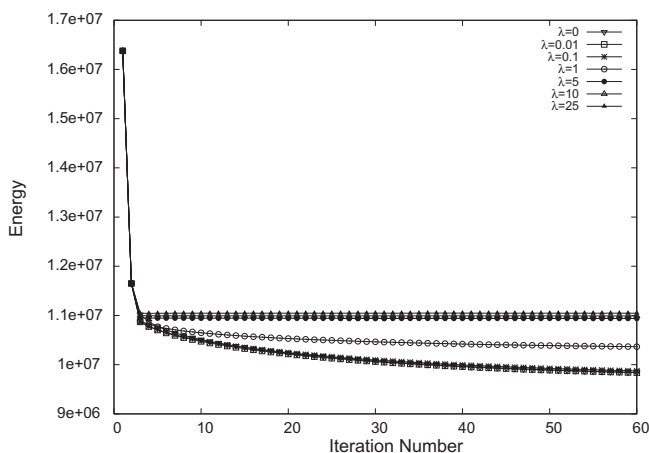

Fig. 2. The L2GF energy curves calculated from Eq. (13) for different value of λ with 60 iterations using 1000 projections (SNR = 0.3).
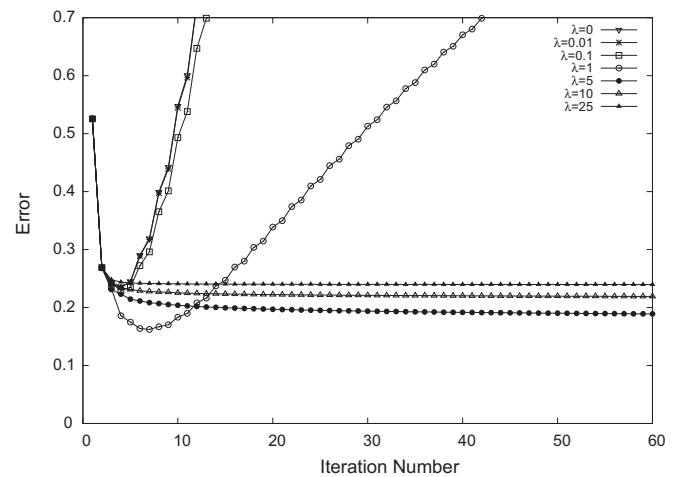
Table 1
Resolution comparison of the five methods for different SNR.

	WBP/thr	Fourier	Block ART/relax	SIRT/relax/iters	L2GF/λ/iters
0.01	17.49/0.05	18.10	17.97/0.001	17.98/0.005/50	15.95/20/50
0.1	9.80/0.03	9.94	9.98/0.005	9.96/0.1/50	9.53/5/50
1	9.26/0.02	9.27	9.33/0.01	9.32/0.1/50	9.08/5/50
10	9.01/0.005	8.99	9.02/0.01	9.01/0.5/50	8.97/1/50

ART, SIRT and our L2GF program, respectively. The threshold for WBP, relaxation parameters for block ART and SIRT, and L2GF's parameter λ are well chosen to obtain desirable resolutions. The results are given in Tables 2 and 3 and Fig. 6. From Table 2, we can see that the L2GF reconstruction yields better resolutions than the other four methods. The resolution improvement seen for the L2GF algorithm decreases as the number of projections increases. Fig. 3(a–e) shows five slices corresponding to the reconstruction results for the five methods using 10,000 projections. All these

reconstructed maps have been filtered at the resolution given by the FSC. In Table 3, an error value 0.18 is used to halt iteration for block ART, SIRT and L2GF. For block ART, one iteration is enough to obtain the error less than 0.18, while SIRT and L2GF need 16 and 30 iterations, respectively. The L2GF algorithm has been parallelized using MPI (Message Passing Interface). Twelve Intel Xeon E5630@2.53 GHz CPU cores are used for these iterative reconstructions. From Table 3, we can see that L2GF needs more time than the other four methods to obtain a better resolution. Similarly, in this test, the L2GF algorithm is performed with $\lambda = 0$ for the first 10 iterations, and $\lambda \neq 0$ for the rest of iterations.

3.2. Electron microscope data:group II chaperonin

For the electron microscope data test, we use as our test example the group II chaperonin of the archaea *Acidianus tengchongensis* strain S5T. This archaea contains two types of chaperonin subunits ATcpnα and ATcpnβ. Prior work has shown that recombinant ATcpnα (rATcpnα) assembles into an 8-fold double ring structure, and recombinant ATcpnβ (rATcpnβ) into a 9-fold structure (Wang
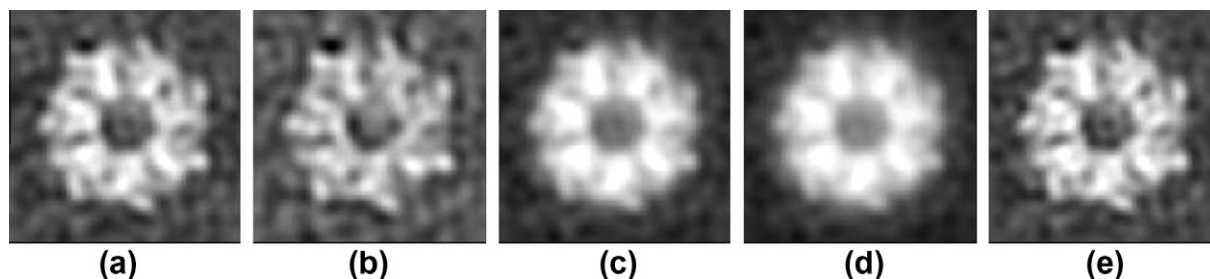
Fig. 5. From left to right, slices at $z = 26$ in the reconstructed maps of GroEL from 1000 projections with SNR = 0.01 using WBP, Fourier, block ART, SIRT and L2GF, respectively.

Table 2
FSC resolution comparison of the five reconstruction methods for different number of projections.

	WBP/thr	Fourier	Block ART/relax	SIRT/relax/iters	L2GF/λ/iters
1000	4.33/0.03	4.61	5.15/0.005	5.15/0.1/40	4.10/10/40
10,000	3.47/0.02	3.75	3.57/0.005	3.57/0.6/30	3.13/5/30
50,000	2.96/0.005	2.99	3.44/0.007	3.13/0.6/16	2.92/5/15
100,000	2.93/0.005	2.95	3.41/0.007	2.97/0.6/11	2.90/5/13

Table 3
Comparison of the five reconstruction methods.

1J4Z	WBP	Fourier	Block ART	SIRT	L2GF
$FSC_{0.5}$	3.47 Å	3.75 Å	3.57 Å	3.72 Å	3.13 Å
Error	0.109	1.047	0.102	0.176	0.176
CPU cores	1	1	12	12	12
Time (Min.)	81	32	34/per iteration	34/per iteration	87/per iteration
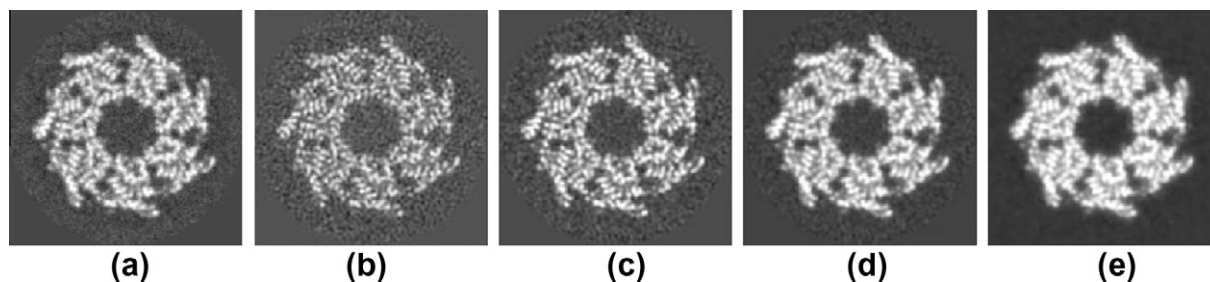
Fig. 6. From left to right, slices at $z = 36$ in the reconstructed maps from 10000 projections with SNR = 0.3 using WBP, Fourier, block ART, SIRT and L2GF, respectively.
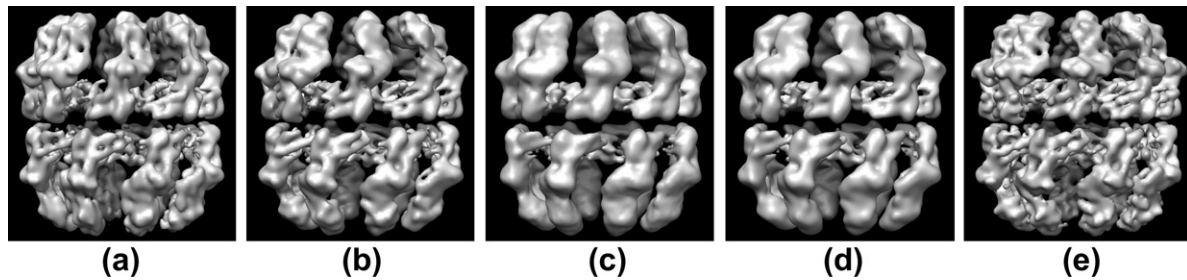
Fig. 7. From left to right, iso-surface display of the reconstructed maps of group II chaperonin using WBP, Fourier, block ART, SIRT and L2GF, respectively.
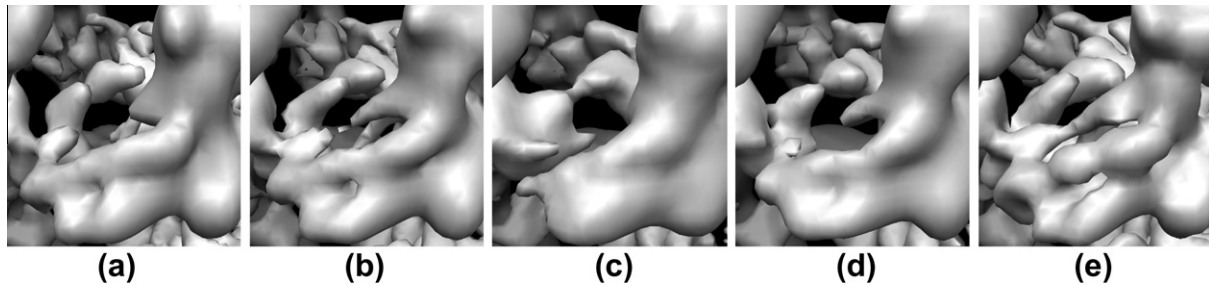
Fig. 8. The equatorial domains corresponding the five maps in Fig. 7 are zoomed in.

et al., 2010). A common feature shared by rATcpnα and rATcpnβ is a characteristic double ring structure, in which each subunit contain three domains: an equatorial domain, an apical domain and an intermediate domain. The crystal structure of rATcpnβ has been published (Huo et al., 2010). The real dataset rATcpnα used in this paper for reconstruction is collected using FEI Titan Krios (to be published).

The CTF for each micrograph is determined using CTFFIND3 (Mindell and Grigorieff, 2003). The phases are flipped with the 'applyctf' program in the EMAN suite (Ludtke et al., 1999). These micrographs are interpolated down by 2×2 binning, resulting in a pixel size of 1.866 Å. Tens of thousands of particles with the size 143×143 are picked using FindEM (Roseman, 2004) and Gautomatch (Zhang et al., 2011). Two-dimensional image analysis, intensive classification and averaging of these particles using the program 'refine2d.py' in the EMAN suite reveal many classes with top views of 8-fold symmetry. Bad classes with no clear structural information are discarded. An initial model is created using EMAN's 'startcsym' and Gaussian noise is added to it. A python script including the Xmipp's reconstruction programs and the L2GF program are written to perform the alignments and projection matching refinements. We reconstruct the 8-fold symmetric map of rATcpnα from 9310 images out of initial 13896 images using the python script with an initial angular increment of 10° until there is no significant change in angle assignments between successive rounds. This test runs on the high performance cluster at the Core Facilities for Protein Sciences, CAS. For Xmipp's program 'xmipp_reconstruct_wbp', the threshold for filter values is chosen to be 0.005. Corresponding to lower or higher than this threshold value, the reconstruction result is noisier or smoother and could not obtain a significant improved resolution. For Xmipp's block ART program, we choose the relaxation parameter to be 0.003 and for Xmipp's SIRT, 0.6. For L2GF, the parameter λ is set to be 10. All the initial functions in the three iteration reconstruction methods are set to be zero. For SIRT, 30 iterations are used for each round. For L2GF, 50 iterations are used with $\lambda = 0$ for the first 40 iterations and $\lambda = 10$ for the rest of the iterations in each refinement. These final results are shown in Figs. 7–9. The resolutions are computed by randomly splitting the dataset into two halves,
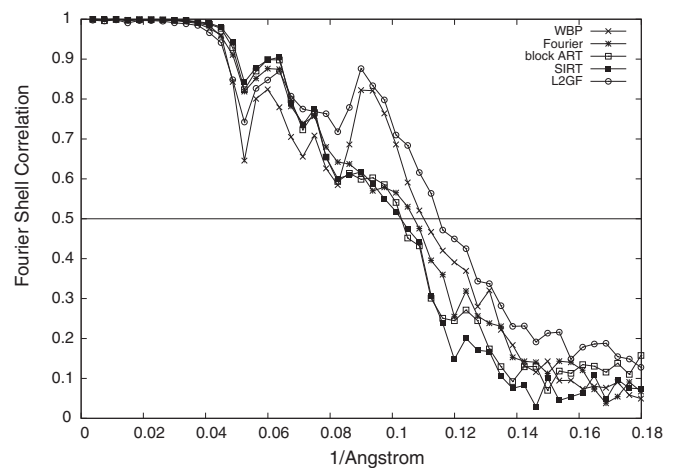
Fig. 9. The Fourier shell correlation curves of group II chaperonin reconstruction for the five methods.

doing reconstruction for each half-set and then computing the FSC between the two half-set reconstruction results. Fig. 7(a–e) shows the reconstructed density maps using WBP, Fourier method, block ART, SIRT and L2GF, respectively. These maps have been filtered at their resolutions given by the FSC, and normalized in the same way. The maps displayed in Fig. 7 have the same contour level 3.16. Enlarged portions of the equatorial domain in these maps are shown in Fig. 8. We can see clearer α helices in the equatorial domain in the L2GF map of Fig. 8(e) than the other four maps. The resolutions corresponding to WBP, Fourier, block ART, SIRT and L2GF calculated using Fig. 8 are 9.07 Å, 9.35 Å, 9.72 Å, 9.73 Å and 8.69 Å, respectively. We conclude that the L2GF method achieves a better resolution than the other four methods.

4. Discussion and conclusions

In this paper, we proposed a three-dimensional reconstruction algorithm named L2GF using the L^2-gradient flow of energy model

(1). The L^2-gradient flow was constructed using the first-order variation of the energy model. By numerical discretization in the temporal and spatial directions using the forward Euler scheme and the finite element method, respectively, the L2GF algorithm was derived. A temporal step-size formula was proposed to accelerate this algorithm. We have tested this algorithm using one simulated dataset and one experimental dataset.

In our simulated dataset, we have found that the a high value of the parameter λ should be selected for low SNR which can be progressively decreased as SNR increases. This conclusion is consistent with Eq. (1). For high SNR, λ is small, the contribution of TV regularization term J_2 is low. For low SNR, λ is high, increasing the contribution of J_2 to properly deal with noise. In our tests, we have found that λ in the range [5,25] is a good choice for data with low SNR.

In the test with different parameter choices of λ, we performed the L2GF algorithm with $\lambda=0$ for two iterations, and with $\lambda \neq 0$ for the rest of the iterations. Since the initial function f was zero, the regularization term J_2 was identically zero during the first iteration. From Figs. 2 and 3 we found that the reconstructed results had large energies and errors for $\lambda = 0$ at the first two iterations. Hence we performed the L2GF algorithm with $\lambda \neq 0$ from the third iteration. In general, we need some initial iterations with $\lambda = 0$ in performing the L2GF algorithm. The initial number of iterations with $\lambda = 0$ changed according to the dataset. In the other tests, we used 10 iterations for the simulated dataset and 40 iterations

then the cubic B-spline basis $N_0^3(x)$ with support $[-2,2]$ is defined recursively by

$$N_0^{m+1}(x) = N_0^m(x) * \text{rect}(x), \quad m = 0, 1, 2,$$

where $*$ denotes the convolution of two functions, and $N_0^0(x) = \text{rect}(x)$. Other basis functions associated with the offset α are defined by the shifting of $N_0^3(x)$, i.e.,

$$N_\alpha^3(x) = N_0^3(x - \alpha), \tag{A.1}$$

where α could be any real number.

Appendix B. Discretization

Define the tri-cubic B-spline space $\mathcal{X}_B$:

$$\mathcal{X}_B = \left\{ f(\mathbf{x}) : f(\mathbf{x}) = \sum_{i,j,k=-[n/2]+2}^{-[n/2]+n-3} f_{ijk}\phi_{ijk}(\mathbf{x}), f_{ijk} \in \mathbb{R}, \mathbf{x} \in \Omega \right\}, \tag{B.1}$$

where $\phi_{ijk}(\mathbf{x}) = N_i^3(x)N_j^3(y)N_k^3(z)$, $\mathbf{x} = [x, y, z]^T$. For any $\psi(\mathbf{x}) \in \mathcal{X}_B$, let

$$f_\epsilon = f + \epsilon\psi.$$

Then the first-order variations of $J_1(f)$ and $J_2(f)$ can be derived as follows

$$\delta(J_1(f), \psi) = \sum_{\mathbf{d} \in \mathbb{D}} \int_{\mathbb{R}^2} \frac{d\left[(X_{\mathbf{d}}f_\epsilon - g_{\mathbf{d}})^2 \right]}{d\epsilon} d\mathbf{u} \Bigg|_{\epsilon=0} = 2\sum_{\mathbf{d} \in \mathbb{D}} \int_{\mathbb{R}^2} (X_{\mathbf{d}}f - g_{\mathbf{d}})X_{\mathbf{d}}\psi \, d\mathbf{u}. \tag{B.2}$$

$$\delta(J_2(f), \psi) = \int_{\mathbb{R}^3} \frac{d[\|\nabla f_\epsilon(\mathbf{x})\|]}{d\epsilon} d\mathbf{x} \Bigg|_{\epsilon=0} = \int_{\mathbb{R}^3} \frac{d\left[\sqrt{\langle \nabla f_\epsilon(\mathbf{x}), \nabla f_\epsilon(\mathbf{x}) \rangle} \right]}{d\epsilon} d\mathbf{x} \Bigg|_{\epsilon=0} = \int_{\mathbb{R}^3} \frac{1}{2\|\nabla f(\mathbf{x})\|} \frac{d[\langle \nabla f_\epsilon(\mathbf{x}), \nabla f_\epsilon(\mathbf{x}) \rangle]}{d\epsilon} d\mathbf{x} \Bigg|_{\epsilon=0} = \int_{\mathbb{R}^3} \frac{\nabla f(\mathbf{x})^T \nabla \psi(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} d\mathbf{x}. \tag{B.3}$$

for the experimental dataset, respectively, with $\lambda = 0$, plus the rest of iterations with $\lambda \neq 0$. By comparing the L2GF algorithm with WBP, Fourier method, block ART and SIRT, we have showed that our algorithm can obtain better resolutions. The L2GF program is freely available from http://lsec.cc.ac.cn/~xuguo/misce.htm.

Acknowledgments

Appendix A. Notations

Let $\text{rect}(t)$ denote the rectangular function defined as

$$\text{rect}(t) = \begin{cases} 0 & \text{if } |t| > \frac{1}{2}, \\ \frac{1}{2} & \text{if } |t| = \frac{1}{2}, \\ 1 & \text{if } |t| < \frac{1}{2}, \end{cases}$$

Using (B.2) and (B.3), and restricting the integral domain to the image domain, we have the following weak-form L^2-gradient flow

$$\int_\Omega \frac{\partial f}{\partial t} \psi \, d\mathbf{x} = -2\sum_{\mathbf{d} \in \mathbb{D}} \int_{\Omega_0} (X_{\mathbf{d}}f - g_{\mathbf{d}})X_{\mathbf{d}}\psi \, d\mathbf{u} - \lambda \int_\Omega$$

$$\times \frac{(\nabla f)^T \nabla \psi}{\|\nabla f\|} \, d\mathbf{x}. \tag{B.4}$$

In order to solve the L^2-gradient flow (B.4), we first discretize the left-handed side using forward Euler scheme:

$$\int_\Omega \frac{\partial f^{(m)}}{\partial t} \psi \, d\mathbf{x} \approx \int_\Omega \frac{f^{(m+1)} - f^{(m)}}{\tau} \psi \, d\mathbf{x},$$

where m is the iteration number and τ is the temporal step-size. For the right-handed side of (B.4), we use the finite element method in the tri-cubic B-spline space $\mathcal{X}_B$. At every iteration step m ($m = 1, 2, \ldots$), suppose

$$f^{(m)}(\mathbf{x}) = \sum_{i,j,k} f_{ijk}^{(m)} \phi_{ijk}(\mathbf{x})$$

with $\{f_{ijk}^{(m)}\}$ known. We take the test function ψ as

$$\phi_{i'j'k'}(\mathbf{x}) = N_{i'}^3(x)N_{j'}^3(y)N_{k'}^3(z),$$
$$i', j', k' = -[n/2] + 2, -[n/2] + 3, \cdots, -[n/2] + n - 3,$$

and discretize (B.4) on the both sides and obtain a set of equations

$$MX = B$$

with $X = [f_{ijk}^{(m+1)}]^T$ unknown. The elements of matrix M and B are in the form

$$\int_\Omega N_i^3(x)N_{i'}^3(x)N_j^3(y)N_{j'}^3(y)N_k^3(z)N_{k'}^3(z) \ d\mathbf{x},$$

and

$$\int_\Omega f^{(m)}\phi_{i'j'k'} \ d\mathbf{x} - \tau\delta(J_1(f^{(m)}),\phi_{i'j'k'}) - \tau\lambda\delta(J_2(f^{(m)})\phi_{i'j'k'}),$$

respectively. The one-dimensional integrals above can be computed by Gaussian quadrature formula (see Abramowitz and Stegun, 1972; Xu and Shi, 2006). Using the fact that the B-spline basis are locally supported, these one-dimensional integrals can be efficiently calculated. Solving these equations is very time consuming when the projection images have a large size. And the number of the elements of M is $(n-4)^6$, it is very possible that the required space for storing the matrix M is beyond the memory capacity of the computer in use. To overcome these difficulties, alternatively we employ a set orthogonal basis $\tilde{N}_\alpha^3(x)$ of cubic B-spline basis functions derived from Schmidt orthogonal process. Using these new orthogonal basis functions, $f^{(m)}$ can be represented as

$$f^{(m)}(\mathbf{x}) = \sum_{ijk} \tilde{f}_{ijk}^{(m)}\tilde{N}_i^3(x)\tilde{N}_j^3(y)\tilde{N}_k^3(z), \quad \mathbf{x}=[x,y,z]^T.$$

Taking test function ψ as

$$\tilde{\phi}_{i'j'k'}(\mathbf{x}) = \tilde{N}_{i'}^3(x)\tilde{N}_{j'}^3(y)\tilde{N}_{k'}^3(z) \tag{B.5}$$

and repeating the above discretization process, we can derive a set of new equations

$$\tilde{X} = \tilde{B} \tag{B.6}$$

with $\tilde{X} = [\tilde{f}_{i'j'k'}^{(m+1)}]^T$ unknown. Therefore, solving these equations is reduced to calculating the vector $\tilde{B} = [\tilde{B}_{i'j'k'}]^T$. Given i',j',k',

$$\tilde{B}_{i'j'k'} = \int_\Omega f^{(m)}\tilde{\phi}_{i'j'k'} \ d\mathbf{x} - \tau\delta(J_1(f^{(m)}),\tilde{\phi}_{i'j'k'})$$
$$- \tau\lambda\delta(J_2(f^{(m)}),\tilde{\phi}_{i'j'k'}). \tag{B.7}$$

The largest cost in calculating vector $\tilde{B}$ is from the term $\delta(J_1(f^{(m)}),\tilde{\phi}_{i'j'k'})$.

$$\delta(J_1(f^{(m)}),\tilde{\phi}_{i'j'k'}) = 2\sum_{\mathbf{d}\in\mathbb{D}} \int_{\Omega_0} (X_\mathbf{d}f^{(m)} - g_\mathbf{d})X_\mathbf{d}\tilde{\phi}_{i'j'k'} \ d\mathbf{u}$$
$$= 2\sum_{\mathbf{d}\in\mathbb{D}} \Omega_0 \left(\sum_{i,j,k} f_{ijk}^{(m)}X_\mathbf{d}\phi_{ijk} - g_\mathbf{d}\right)X_\mathbf{d}\tilde{\phi}_{i'j'k'} \ d\mathbf{u}. \tag{B.8}$$

Given any projection direction $\mathbf{d}$, we can compute the projection $X_\mathbf{d}\phi_{000}$ quickly based on the property of local support of B-spline basis function. The other tri-cubic B-spline basis functions projections $X_\mathbf{d}\phi_{ijk}$ can be easily obtained using the following shift formula:

$$(X_\mathbf{d}\phi_{ijk})(u,v) = (X_\mathbf{d}\phi_{000})(u-[i,j,k]\mathbf{e}_\mathbf{d}^{(1)}, \quad v-[i,j,k]\mathbf{e}_\mathbf{d}^{(2)}). \tag{B.9}$$

Notice that $X_\mathbf{d}\phi_{ijk}$ is locally supported. The cost for computing $X_\mathbf{d}f^{(m)}$ is $O(n^3)$. Then for all $i'j'k'$ the computational complexity of $\delta(J_1(f^{(m)}),\phi_{i'j'k'})$ is $O(N_dn^3)$. Given i',j',k', the cost of converting $\delta(J_1(f^{(m)}),\phi_{i'j'k'})$ to $\delta(J_1(f^{(m)}),\tilde{\phi}_{i'j'k'})$ is $O(1)$. Hence, the computational complexity of $\delta(J_1(f^{(m)}),\tilde{\phi}_{i'j'k'})$ for all $i'j'k'$ is $O(N_dn^3)$. Similarly, the cost of $\delta(J_2(f^{(m)}),\tilde{\phi}_{i'j'k'})$ could be reduced to $O(n^3)$ in terms of the local support of the B-spline basis function. Hence, the total computational complexity of L2GF algorithm for one iteration step is $O(N_dn^3)$. From the definition of Schmidt orthogonal process, there is a lower-triangular matrix A connecting the two sets of basis functions,

$$[N_0^3, N_1^3, \ldots, N_n^3]^T = A[\tilde{N}_0^3, \tilde{N}_1^3, \ldots, \tilde{N}_n^3]^T,$$

where

$$A = \begin{bmatrix} a_{00} & & & & \\ a_{10} & a_{11} & & & \\ a_{20} & a_{21} & a_{22} & & \\ a_{30} & a_{31} & a_{32} & a_{33} & \\ & \ddots & \ddots & \ddots & \ddots \\ & & a_{n,n-3} & a_{n,n-2} & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

The matrix A can be calculated by using the following formulas:

$$a_{00} = \|N_0^3\|, \ a_{10} = \frac{\langle N_1^3, N_0^3\rangle}{a_{00}}, \ a_{11} = \sqrt{\langle N_1^3, N_1^3\rangle - a_{10}^2}, \tag{B.10}$$

$$a_{k,k-3} = \frac{\langle N_k^3, N_{k-3}^3\rangle}{a_{k-3,k-3}}, \tag{B.11}$$

$$a_{k,k-2} = \frac{\langle N_k^3, N_{k-2}^3\rangle - a_{k-2,k-3}a_{k,k-3}}{a_{k-2,k-2}}, \tag{B.12}$$

$$a_{k,k-1} = \frac{\langle N_k^3, N_{k-1}^3\rangle - a_{k-1,k-3}a_{k,k-3} - a_{k-1,k-2}a_{k,k-2}}{a_{k-1,k-1}}, \tag{B.13}$$

$$a_{kk} = \sqrt{\langle N_k^3, N_k^3\rangle - a_{k,k-3}^2 - a_{k,k-2}^2 - a_{k,k-1}^2}, k=2,3,\ldots,n. \tag{B.14}$$

For $k = 2$, we define $a_{2,-1} = 0$.

Appendix C. Temporal step-size

From the iteration scheme (10), we can see that after one step iteration, the function f is updated as

$$f^{(m+1)} = f^{(m)} + \tau\Delta f^{(m)},$$

where $f^{(m)}$ is the function at previous step, $\Delta f^{(m)}$ is the increment of $f^{(m)}$ computed using the L^2-gradient flow. Now we address the question of how large of a temporal size τ should be used. We determine τ such that

$$\sum_{\mathbf{d}\in\mathbb{D}} \int_{\Omega_0} \left[X_\mathbf{d}(f^{(m)} + \tau\Delta f^{(m)}) - g_\mathbf{d}\right]^2 \ d\mathbf{u} + \lambda\int_\Omega \|\nabla(f^{(m)} + \tau\Delta f^{(m)})\| \ d\mathbf{x}$$
$$= \min.$$

By solving this minimization problem, we have

$$\tau = -\frac{2a + \lambda c}{2b + \lambda d}, \tag{C.1}$$

where

$$a = \sum_{\mathbf{d}\in\mathbb{D}} \int_{\Omega_0} \left[X_\mathbf{d}(f^{(m)}) - g_\mathbf{d}\right]X_\mathbf{d}(\Delta f^{(m)}) \ d\mathbf{u},$$

$$b = \sum_{\mathbf{d}\in\mathbb{D}} \int_{\Omega_0} \left[X_\mathbf{d}(\Delta f^{(m)})\right]^2 \ d\mathbf{u},$$

$$c = \int_\Omega \frac{\langle\nabla(\Delta f^{(m)}),\nabla f^{(m)}\rangle}{\|\nabla f^{(m)}\|} \ d\mathbf{x},$$

$$d = \int_\Omega \left[\frac{\langle\nabla(\Delta f^{(m)}),\nabla(\Delta f^{(m)})\rangle}{\|\nabla f^{(m)}\|} - \frac{\langle\nabla(\Delta f^{(m)}),\nabla f^{(m)}\rangle^2}{\|\nabla f^{(m)}\|^3}\right] \ d\mathbf{x}.$$

If $\lambda = 0$, $\tau = -a/b$. It is possible to get a negative time step-length using (C.1). In this case the L2GF algorithm will not converge. To solve this problem, we choose the time step-length as

$$\tau_1 = \begin{cases} \tau, & \text{if} \ \ \tau \geqslant 0, \\ -\frac{a}{b}, & \text{else if} \ \ a < 0, \\ 0, & \text{else}. \end{cases} \tag{C.2}$$

References

Abramowitz, M., Stegun, I.A., 1972. Handbook of Mathematical Functions with Formulas, Graphics, and Mathematical Tables. America Dover Publications, Inc..

Aganj, I., Bartesaghi, A., Borgnia, M., Liao, H.Y., Sapiro, G., Subramaniam, S., 2007. Regularization for inverting the Radon transform with wedge consideration. In: Proc. Fourth IEEE Int. Symp. Biomedical Imaging: From Nano to Macro (ISBI2007), Arlington VA, USA, pp. 217–220.

Andersen, A.H., Kak, A.C., 1984. Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm. Ultrason. Imaging 6, 81–94.

Carazo, J.M., 1992. The fidelity of 3D reconstructions from incomplete data and the use of restoration methods. In: Frank, J. (Ed.), Electron Tomography: Three-dimensional Imaging with the Transmission Electron Microscope. Springer, London, pp. 117–166 (Chapter 6).

Crowther, R.A., DeRosier, D.J., Klug, A., 1970. The reconstruction of a three-dimensional structure from projections and its application to electron microscopy. Proc. R. Soc. Lond. 317, 319–340.

DeRosier, D.J., Klug, A., 1968. Reconstruction of three dimensional structures from electron micrographs. Nature 217, 130–134.

Do, S., Karl, W.C., Kalra, M.K., Brady, T.J., Pien, H., 2010. A variational approach for reconstructing low dose images in clinical helical CT. In: Proc. 2010 IEEE Int. Symp. Biomedical Imaging: From Nano to Macro (ISBI2010), Rotterdam, The Netherlands, pp. 784–787.

Frank, J., 2006. Three-dimensional Electron Microscopy of Macromolecular Assemblies. Oxford University Press, New York.

Giaquinta, M., Hildebrandt, S., 1996. Calculus of Variations, vol. I. Springer-Verlag, Berlin.

Gilbert, P., 1972. Iterative methods for the three-dimensional reconstruction of an object from projections. J. Theor. Biol. 36, 105–117.

Gordon, R., Bender, R., Herman, G.T., 1970. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. J. Theor. Biol. 29, 471–481.

van Heel, M., Gowen, B., Matadeen, R., Orlova, E.V., Finn, R., Pape, T., Cohen, D., Stark, H., Schmidt, R., Schatz, M., Patwardhan, A., 2000. Single-particle electron cryo-microscopy: towards atomic resolution. Q. Rev. Biophys. 33, 307–369.

Huo, Y., Hu, Z., Zhang, K., Wang, L., Zhai, Y., Zhou, Q., Lander, G., Zhu, J., He, Y., Pang, X., Xu, W., Bartlam, M., Dong, Z., Sun, F., 2010. Crystal structure of group II chaperonin in the open state. Structure 18, 1270–1279.

Kybic, J., Blu, T., Unser, M.A., 2001. Variational approach to tomographic reconstruction. In: Proc. of SPIE, San Diego CA, USA, pp. 30–39.

Kybic, J., Blu, T., Unser, M.A., 2002. A variational approach – part II: applications. IEEE Trans. Signal Process. 50, 1977–1985.

Ludtke, S.J., Baldwin, P.R., Chiu, W., 1999. EMAN semiautomated software for high-resolution single-particle reconstructions. J. Struct. Biol. 128, 82–97.

Marabini, R., Herman, G.T., Carazo, J.M., 1998. 3D reconstruction in electron microscopy using ART with smooth spherically symmetric volume elements (blobs). Ultramicroscopy 72, 53–65.

Matej, S., Lewitt, R., 2001. 3D-FRP: direct Fourier reconstruction with Fourier reprojection for fully 3-D PET. IEEE Trans. Nucl. Sci. 48, 1378–1385.

Mindell, J.A., Grigorieff, N., 2003. Accurate determination of local defcous and specimen tilt in electron microscopy. J. Struct. Biol. 142, 334–347.

Natterer, F., Wübbeling, F., 2001. Mathematical methods in image reconstruction. SIAM, Philadelphia.

Penczek, P.A., Grasucci, R.A., Frank, J., 1994. The ribosome at improved resolution: new techniques for merging and orientation refinement in 3D cryo-electron microscopy of biological particles. Ultramicroscopy 53, 251–270.

Radermacher, M., 1994. Three-dimensional reconstruction from random projections-orientational alignment via Radon transforms. Ultramicroscopy 53, 121–136.

Radermacher, M., 2006. Weighted back-projection methods. In: Frank, J. (Ed.), Electron Tomography Methods for Three-dimensional Visualization of Structures in the cell. Springer, New York, pp. 245–273 (Chapter 8).

Roseman, A.M., 2004. Find EM-a fast, efficient program for automatic selection of particles from electron micrographs. J. Struct. Biol. 145, 91–99.

Schoenmakers, R., Perquin, R.A., Fliervoet, T.F., Voorhout, W., 2005. High resolution, high throughput electron tomography reconstruction. Microsc. Microanal. 11 (suppl. 2), 312–313.

Sezan, M., Stark, H., 1982. Image restoration by the method of convex projections: part 2-applications and numerical results. IEEE Trans. Med. Imag. MI-1, 95–100.

Sorzano, C.O.S., Jonic, S., El-Bez, C., Carazo, J.M., Carlo, S.D., Thévenaz, P., Unser, M., 2004a. A multiresolution approach to pose assignment in 3-D electron microscopy of single particles. J. Struct. Biol. 146, 381–392.

Sorzano, C.O.S., Marabini, R., Boisset, N., Rietzel, E., Schröder, R., Herman, G.T., Carazo, J.M., 2001. The effect of overabundant projection directions on 3D reconstruction algorithms. J. Struct. Biol. 133, 108–118.

Sorzano, C.O.S., Marabini, R., Herman, G.T., Carazo, J.M., 2005. Multiobjective algorithm parameter optimization using multivariate statistics in three-dimensional electron microscopy reconstruction. Pattern Recognit. 38, 2587–2601.

Sorzano, C.O.S., Marabini, R., Velázquez-Muriel, J., Bilbao-Castro, J.R., Scheres, S.H.W., Carazo, J.M., Pascual-Montano, A., 2004b. Xmipp: a new generation of an open-source image processing package for electron microscopy. J. Struct. Biol. 148, 194–204.

Unser, M., 1999. Splines. A perfect fit for signal and image processing. IEEE Signal Process. Mag. 16, 22–38.

Voorhout, W., Haas, F.D., Frederik, P., Schoenmakers, R., Busing, W., Hubert, D., 2006. An optimized solution for cryo-electron tomography. Microsc. Microanal. 12, 1110–1111.

Wang, L., Hu, Z., Luo, Y., Huo, Y., Ma, Q., He, Y., Zhang, Y., Sun, F., Dong, Z., 2010. Distinct symmetry and limited peptide refolding activity of the thermosomes from the acidothermophilic archaea Acidianus tengchongensis S5 T. Biochem. Biophys. Res. Commun. 393, 228–234.

Xu, G., 2008. Geometric partial differential equation methods in computational geometry. Science Press, Beijing.

Xu, G., Shi, Y., 2006. Progressive computation and numerical tables of generalized Gaussian quadrature formulas. J. Numer. Methods Comput. Appl. 27, 9–23.

Youla, D., Webb, H., 1982. Image restoration by the method of convex projections: part 1-theory. IEEE Trans. Med. Imag. MI-1, 81–94.

Zhang, K., Li, M., Sun, F., 2011. Gautomatch: an efficient and convenient gpu-based automatic particle selection program. Unpublished .

Zhang, K., Zhang, Y., Hu, Z., Ji, G., Sun, F., 2010. Development and frontier of electron microscopy 3D reconstruction. Acta Biophys. Sin. 26, 533–559.