Journal of Computational Mathematics
Vol.29, No.5, 2011, 501–525.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1106-m3302

# INVERSION OF ELECTRON TOMOGRAPHY IMAGES USING $L^2$-GRADIENT FLOWS —- COMPUTATIONAL METHODS[*]

Guoliang Xu and Ming Li
*LSEC, ICMSEC, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100190, China*
*Email: xuguo@lsec.cc.ac.cn, liming@lsec.cc.ac.cn*
Ajay Gopinath and Chandrajit L. Bajaj
*Department of Computer Sciences and Institute of Computational,Engineering & Sciences, University of Texas at Austin, Austin TX 78712, USA*
*Email: ajay.gopinath@gmail.com, bajaj@cs.utexas.edu*

**Abstract**

In this paper, we present a stable, reliable and robust method for reconstructing a three dimensional density function from a set of two dimensional electric tomographic 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 temporal direction. The experimental results show that the proposed method is efficient and effective.

*Mathematics subject classification:* 65D17.
*Key words:* Computational Inversion, Reconstruction, Electric Tomography.

## 1. Introduction

Electron microscopy (EM) is a preferred tool for structural biologists to visualize three-dimensional structures of molecular and cellular complexes in-vitro. Electron tomography (ET) involves acquiring planar EM images of the biological sample from different projection angles and reconstructing a 3D image from these projections [9]. The forward operation assumes the Born approximation of electron beam-specimen interaction [8]. This holds true only when the specimen is weakly scattering. The scattering function depends on the electrostatic properties of the specimen. Images also suffer from effects of blurring owing to specimen deformation during exposure to the electron beam. Another challenge is that the angle of rotation cannot exceed $\pm 70°$(from the horizontal plane) since the beam length through the specimen becomes large at higher tilt angles as it passes through the plane of the sample resulting in poor images. Hence projections are available only for a limited range of tilt angles, which is known as the missing wedge problem [14]. Since there is no unique solution to the inverse reconstruction problem it makes image processing and visualization of ET data a huge challenge. Typically, acquisition and reconstruction processes contribute largely to the error in modeling biological specimens from EM images.

In this paper, we propose an iterative variational reconstruction method. 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

[*] Received January 20, 2010 / Revised version received November 29, 2010 / Accepted January 17, 2011 / Published online September 9, 2011 /

an explicit Euler scheme in the temporal direction. The experimental results show that the proposed method is stable, reliable and robust. By stable, we mean the reconstruction result does not depend on the initialization of the iterative scheme, and the reconstruction result is unique. By reliable, we mean the the projected images of the reconstruction result are as close to the given images as possible. The robustness implies that the reconstruction is not sensitive to small noise in the 2D images.

Our results are demonstrated on a phantom data as well as tomographic images of the AIDS virus complex, specifically the Simian Immunodeficiency Virus (SIV) [4] which is similar to the HIV. Liu, et al. [15] describe the structure of the HIV virus and the action of the spike protein in infecting the host. The phantom data is a spherical shell with nonzero thickness.

The rest of this paper is organized as follows. In Section 2, we review existing 3D EM reconstruction techniques. In Section 3, we establish notation and define several parameters and data structures including image size, B-spline basis function grids and volume grids are described. Sections 4 and 5 provides details of our algorithms. Section 6 gives some illustrative examples. Section 7 contains the conclusion of this paper. Additionally important mathematical facts and proofs on the existence and uniqueness of the reconstruction are given in the Appendix.

## 2. A Review of Existing ET Reconstruction Techniques

Reconstruction methods try to invert the following system of equations that describe the projection process, also called the *forward problem*,

$$I_i = \sum_{n=1}^{N} w_{in} f_n, \quad i = 0, 1, \cdots,$$

where $I_i$ is the measured projection data of the $i$-th projection ray through a space volume with $N$ voxels. $w_{in}$ is a weight that describes the contribution of the $n$-th voxel to the $i$-th projection. Given $I_i$ and $w_{in}$, the goal of the *inverse problem* is to compute $f_n$.

### 2.1. Algebraic Reconstruction Technique (ART)

An iterative approach to solve the inverse problem is to use the Kaczmarz method [12,13,16]. In this technique each tomogram image is treated like a hyperplane with the equation $\mathbf{w}_i{}^T \mathbf{f} = I_i$ in an $N$ dimensional space, where

$$\mathbf{w}_i = [w_{i,1}, \cdots, w_{i,N}]^T, \quad \mathbf{f} = [f_1, \cdots, f_N]^T.$$

A solution to the entire equation system for all tomograms $I = [I_0, I_1 \cdots]^T$ is a point in the $N$ dimensional space where all the hyperplanes intersect. To solve for this solution using an iterative method, the algorithm starts off with a random solution (typically zero) $\mathbf{f}^{(0)}$. This initial solution is then projected onto the first hyperplane. The projected point on the first hyperplane is then projected onto the next hyperplane and so on for all the image values $I_i$. The iterative process will converge to the point of intersection of all the hyperplanes [13]. If there is no unique solution which is typically the case, then the iterative scheme will not converge and will oscillate. To work around this problem a regularizing term $\lambda$ is used in the update equation [5]:

$$\mathbf{f}^{(i+1)} = \mathbf{f}^{(i)} - \lambda \frac{\mathbf{w}_{i+1}^T \mathbf{f}^{(i)} - I_{i+1}}{||\mathbf{w}_{i+1}||^2} \mathbf{w}_{i+1},$$

where $\mathbf{f}^{(i)} = [f_1^{(i)}, \cdots, f_N^{(i)}]^T$.

### 2.2. Simultaneous Iterative Reconstruction Technique (SIRT)

In this technique, first proposed by Gilbert [10], voxels are updated after completely iterating through all the projections several times. The update for a voxel from each projection is not applied immediately. Instead the update is calculated for all the projections as in ART but repeated over several iterations ($M$) and averaged. Each new iteration starts off with an initial solution from the previous iteration. This makes the reconstructed volume less prone to noise but the trade off is decreased contrast.

$$\Delta f_n = \sum_{i=1}^{M} \frac{\Delta f_n^{(i)}}{M},$$

where $\Delta f_n$ is the correction for the $n$-th voxel calculated for all projections and the averaged value is used as the update. This process is performed for $M$ iterations.

### 2.3. Weighted Back Projection (WBP)

In this technique, the projection data is weighted in Fourier space giving more weight to higher frequencies and inverted using back-projection into the spatial domain. Herman [11] describes the implementation details of the back-projection operator and Radermacher [17] provides details of the general weighting function for arbitrary tilt geometry. Let $\tilde{g}_i$ be the 2D Fourier transform of the 2D projection image $g_i$. A weighting function is used to weight the higher frequencies of $\tilde{g}_i$ and this weighted Fourier transform is inverted and back-projected. That is, $g_i^{\text{weighted}} = F_{2D}^{-1}(|\text{weight}|\tilde{g}_i)$ and

$$f(x, y, z) = \sum_i g_i^{\text{weighted}}(x \cos(i) + y \sin(i), z).$$

Due to the radial nature of the Fourier space, at higher frequencies the sampling is much smaller than at lower frequencies. To offset this, a weighting function is used which gives more weight to the higher frequencies than the lower frequencies. Further discussion and mathematical details on iterative and Fourier based methods is available in [16].

## 3. Problem Setting

Let $\{g_{\mathbf{d}}\}$ be a set of the two-dimensional images measured from an unknown three-dimensional function (electric-potential) $f$ by the projection $X_{\mathbf{d}}$ in the direction $\mathbf{d} \in \mathbb{D} \subset S^2$, here $S^2$ stands for the unit sphere in $\mathbb{R}^3$, $\mathbb{D}$ is the collection of all the projection directions. Our problem is to construct $f(\mathbf{x})$, $\mathbf{x} = [x, y, z]^T \in \Omega \subset \mathbb{R}^3$, such that $X_{\mathbf{d}} f$ is as close to $g_{\mathbf{d}}$ as possible and satisfies the boundary condition $f(\mathbf{x}) = 0$ for any $\mathbf{x} \in \partial\Omega$.

We assume that all the measured images have the same size $(n + 1) \times (n + 1)$, the pixel values $g_{\mathbf{d}}$ of each image are defined on integer grid points $(i, j)^T \in \left[-\frac{n}{2}, \frac{n}{2}\right]^2$ (we assume $n$ is an even number). A resampling step can be taken if these assumptions are not satisfied. Since $g_{\mathbf{d}}$ are the projection of $f$, we therefore define $\Omega$ as

$$\Omega = \left[-\frac{n}{2} - 1, \frac{n}{2} + 1\right]^2 \times \left[-\frac{\tau}{2} - 1, \frac{\tau}{2} + 1\right],$$

where $\tau/n$ is a small number since the sample to be reconstructed is thin. $\tau + 2$ is the thickness of the $\Omega$. Again, we assume $\tau$ is an even integer. Setting one layer offset in $\Omega$ is for having room to satisfy the zero boundary condition.

For a given $\mathbf{d} \in \mathbb{D}$. The image values of $g_{\mathbf{d}}$ at the grid points are defined as

$$g_{\mathbf{d}}(i,j) = \int_{-\infty}^{\infty} f(i\mathbf{e}_{\mathbf{d}}^{(1)} + j\mathbf{e}_{\mathbf{d}}^{(2)} + t\mathbf{d})\, \mathrm{d}t, \quad (i,j)^T \in \left[-\frac{n}{2}, \frac{n}{2}\right]^2,$$

for the unknown function $f$, where $\mathbf{e}_{\mathbf{d}}^{(1)}$ and $\mathbf{e}_{\mathbf{d}}^{(2)}$ are two directions satisfying

$$\|\mathbf{e}_{\mathbf{d}}^{(1)}\| = \|\mathbf{e}_{\mathbf{d}}^{(2)}\| = 1, \quad ,\langle \mathbf{e}_{\mathbf{d}}^{(1)}, \mathbf{e}_{\mathbf{d}}^{(2)} \rangle = 0, \quad \langle \mathbf{e}_{\mathbf{d}}^{(1)}, \mathbf{d} \rangle = 0, \quad \langle \mathbf{e}_{\mathbf{d}}^{(2)}, \mathbf{d} \rangle = 0. \tag{3.1}$$

$\mathbf{e}_{\mathbf{d}}^{(1)}$ and $\mathbf{e}_{\mathbf{d}}^{(2)}$ also determine the in-plane rotation.

Given an even and positive integer $m = 2(l + 2)$, let

$$h = \frac{n+2}{m}, \quad t = \frac{\tau + 2}{2h} - 2.$$

For simplicity, we require that $h$ and $t$ are integers. The following two cases are the ones we often use: (i) Take $m = n + 2$, which gives $h = 1$, $l = (n/2) - 1$, $t = (\tau/2) - 1$; (ii) Take $m = (n + 2)/2$, which gives $h = 2$, $l = [(n + 2)/4] - 2$, $t = [(\tau + 2)/4] - 2$.

Using spacing $h$, the domain $\Omega$ can be uniformly partitioned with grid point $(i, j, k)h$ for $(i, j, k) \in [-l - 2, l + 2]^2 \times [-t - 2, t + 2]$. The function $f$ to be reconstructed is represented as

$$f(\mathbf{x}) = \sum_{i=-l}^{l} \sum_{j=-l}^{l} \sum_{k=-t}^{t} f_{ijk} N_i(x) N_j(y) N_k(z), \quad \mathbf{x} = (x, y, z)^T \in \Omega, \tag{3.2}$$

where $N_\beta$ are one dimensional cubic B-spline basis functions (see Appendix A) defined on the uniform grid $[-2h + \beta h, -h + \beta h, \cdots, 2h + \beta h]$.

## 4. Reconstruction of $f$

Let $\Gamma_c = \{\mathbf{x} \in \mathbb{R}^3 : f(\mathbf{x}) = c\}$ be a level-set of function $f$. To reconstruct function $f$, we minimize the following energy functional.

$$J(f) = J_1(f) + \alpha J_2(f), \tag{4.1}$$

where

$$J_1(f) = \sum_{\mathbf{d} \in \mathbb{D}} \int_{\mathbb{R}^2} (X_{\mathbf{d}} f - g_{\mathbf{d}})^2\, \mathrm{d}u \mathrm{d}v, \qquad J_2(f) = \int_{-\infty}^{\infty} \int_{\Gamma_c} g(\mathbf{x}) \mathrm{d}A\ \mathrm{d}c,$$

and $\alpha \geq 0$ is a given weighting factor balancing the two $J's$. We first explain the roles played by $J_1$ and $J_2$:

1. $J_1$ is used to minimize the total error of the measured images and the projections of the reconstructed function. Hence, this term is usually called the fidelity term.
2. $J_2$ is a regularization term that makes the iso-surface $\Gamma_c$ smooth in certain sense, depending on what kind $g$ is used. From co-area formula (see [7]), we have

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

where $\nabla$ is the usual gradient operator acting on a three dimensional function. In this paper, we take $g(\mathbf{x}) = 1$ or $g(\mathbf{x}) = \|\nabla f(\mathbf{x})\|$. When $g(\mathbf{x}) = 1$, $J_2$ represents the total surface area, and minimizing $J_2$ leads to the level sets of the solution close to minimal surfaces. When $g(\mathbf{x}) = \|\nabla f(\mathbf{x})\|$, minimizing $J_2$ leads to $f$ approaching a constant function.

Since the energy functional $J(f)$ is nonlinear with respect to the unknown function $f$, we use the following iterative algorithm to minimize the energy:

**Algorithm 4.1** *Reconstruction of $f$.*

1. Load images $g_{\mathbf{d}}$ for all $\mathbf{d} \in \mathbb{D}$.
2. Set $k = 0$, and choose an initial $f^{(0)}$.
3. Compute $f^{(k+1)}$ by minimizing $J(f)$ by solving an $L^2$-gradient flow.
4. Check the termination conditions, if they are satisfied, stop the iteration, otherwise set $k$ to be $k+1$ and then go back to step 3.
5. Output $f = f^{(k+1)}$.

In principle, our initial $f^{(0)}$ can be arbitrarily chosen, including taking it to be zero. If a rough approximation of $f$ is available, e.g., from application of other ET reconstruction methods, that can be used as an initial $f^{(0)}$. The choice of the initial $f^{(0)}$ can influence the overall running time, but cannot affect the final reconstructed $f$. Hence our method can also be considered as a further refinement of prior ET reconstruction schemes. The termination conditions are set as

$$|J(f^{(k+1)}) - J(f^{(k)})| < \epsilon,$$

where $\epsilon$ is a chosen residual error threshold.

Obviously, the central task in the above algorithm is to minimize $J(f)$. This goal is achieved by solving an $L^2$-gradient flow of $J(f)$ in the B-spline function space. We construct the flow by a variation of $J_1(f)$ and $J_2(f)$. Let $\delta(J_i(f), \psi)$ be the variation of $J_i(f)$ with respect to a trial function $\psi$. Then it is easy to derive that

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

For $g(\mathbf{x}) = 1$ and $g(\mathbf{x}) = \|\nabla f(\mathbf{x})\|$, $\delta(J_2(f), \psi)$ are respectively given as follows:

$$\delta(J_2(f), \psi) = \int_{\mathbb{R}^3} \frac{(\nabla f)^T \nabla \psi}{\|\nabla f\|} d\mathbf{x}, \tag{4.2}$$

$$\delta(J_2(f), \psi) = 2\int_{\mathbb{R}^3} (\nabla f)^T \nabla \psi d\mathbf{x}. \tag{4.3}$$

Using these first order variations, we construct the following weak form $L^2$-gradient flow:

$$\int_{\mathbb{R}^3} \frac{\partial f}{\partial t} \psi \mathrm{d}\mathbf{x} + \delta(J_1(f), \psi) + \alpha \delta(J_2(f), \psi) = 0, \tag{4.4}$$

for all $\psi$ in the B-spline function space. We solve (4.4) using numerical methods. For the discretization in the temporal direction, we use an explicit forward Euler scheme. For the discretization in the spatial direction, we use tri-cubic B-spline finite elements. Computational details are presented in the next section.

We choose B-splines as the finite elements for several reasons. First, our regularizer involve partial derivative computation. Hence smooth functions are required. Secondly, B-spline basis have local support and a closed form Fourier transform (see Appendix). Finally, fast algorithms exist for transforming discrete sampled (i.e. imaging) data to B-spline representations (see [3]). These properties make our finite element computation using B-splines very efficient.

## 5. Computational Details

Let $N_i(s)$ be the one-dimensional cubic B-spline basis functions defined on the 1-D uniform grid $[ih-2h,\cdots,ih+2h]$ (see Appendix for details). Then over the volume $\Omega$, we represent $f(\mathbf{x})$ as

$$f(\mathbf{x}) = \sum_{i=-l}^{l}\sum_{j=-l}^{l}\sum_{k=-t}^{t} f_{ijk}\phi_{ijk}(\mathbf{x}), \quad \phi_{ijk}(\mathbf{x}) = N_i(x)N_j(y)N_k(z), \quad \mathbf{x}=[x,y,z]^T, \tag{5.1}$$

with $f_{ijk}$ as unknown coefficients. Taking the test function $\psi$ in (4.4) as $\phi_{\alpha\beta\gamma}(\mathbf{x})$, we then obtain a matrix form $MX = B$ for (4.4). The entries of matrix $M$ are in the form

$$\int_{\mathbb{R}^3}\phi_{ijk}(\mathbf{x})\phi_{\alpha\beta\gamma}(\mathbf{x})\mathrm{d}\mathbf{x} = \int_{\mathbb{R}} N_i(x)N_\alpha(x)\mathrm{d}x\int_{\mathbb{R}} N_j(y)N_\beta(y)\mathrm{d}y\int_{\mathbb{R}} N_k(z)N_\gamma(z)\mathrm{d}z.$$

The one-dimensional integrals above can be computed by Gaussian quadrature formulas (see [1,21]). Using the fact that the B-splines are locally supported, these one-dimensional integrals can be calculated efficiently. Note that matrix $M$ is fixed during the entire reconstruction process. Hence, $M^{-1}$ could be pre-computed. The elements of the vector $B$ are

$$-[\delta(J_1(f),\phi_{\alpha\beta\gamma}) + \delta(J_2(f),\phi_{\alpha\beta\gamma})]$$

that are represented as the integrations of $f$, the B-spline basis functions and their derivatives. In the following, we discuss how matrix $M$ and vector $B$ are efficiently computed.

### 5.1. Computation of $M$

Let $L$ be the number of the utilized basis functions $N_i(x)$ (and $N_j(y)$) and $T$ be the number of basis functions $N_k(z)$. Then the total number of tensor product basis functions $\phi_{ijk}(x,y,z)$ is $L^2T$. Hence the number of the elements of $M$ is $L^4T^2$. It is very possible that the required space for storing the matrix $M$ is beyond the main memory capacity of the computer. To overcome this difficulty, we orthogonalize the basis functions $N_i(s)$ using the Schmidt orthogonalization process (see [6]), obtaining a set of new basis functions $\tilde{N}_i(s)$, such that

$$\int_{\mathbb{R}} \tilde{N}_i(s)\tilde{N}_j(s)\mathrm{d}s = \delta_{ij}, \quad i,j=-l,-l+1,\cdots,l.$$

Let

$$\tilde{\phi}_{ijk}(\mathbf{x}) = \tilde{N}_i(x)\tilde{N}_j(y)\tilde{N}_k(z).$$

Then by representing the function $f$ using the new basis $\tilde{\phi}_{ijk}$ and taking the test function $\psi = \tilde{\phi}_{\alpha\beta\gamma}$, we obtain a unit matrix $M$. Hence there is no need to store and invert the matrix.

Now we have representation (5.1) of $f$ and the representation

$$f(\mathbf{x}) = \sum_i\sum_j\sum_k \tilde{f}_{ijk}\tilde{\phi}(\mathbf{x}), \quad \mathbf{x}=[x,y,z]^T. \tag{5.2}$$

Since there is a lower-triangular matrix $A$ connecting the old and the new basis,

$$[N_1, N_2, \cdots, N_n]^T = A[\tilde{N}_1, \tilde{N}_2, \cdots, \tilde{N}_n]^T.$$

The two representations of $f$ can be converted easily from one to the other. We take the advantageous of having the above dual form representations to reduce the cost for the computation of $B$.

## 5.2. Computation of $\delta(J_1, \psi)$

Consider the computation of the term $\delta(J_1(f), \psi)$ in (4.4). Using (5.1) and taking $\psi = \tilde{\phi}_{\alpha\beta\gamma}$, we have

$$\delta(J_1(f), \tilde{\phi}_{\alpha\beta\gamma}) = 2\sum_{\mathbf{d}\in\mathbb{D}} \int_{\mathbb{R}^2} (X_{\mathbf{d}} f - g_{\mathbf{d}}) X_{\mathbf{d}} \tilde{\phi}_{\alpha\beta\gamma} \mathrm{d}u\mathrm{d}v. \tag{5.3}$$

We explain how (5.3) can be efficiently and accurately computed. The computation mainly includes two steps:

1. Compute $\delta(J_1(f), \phi_{\alpha\beta\gamma})$ for all $\alpha, \beta, \gamma$, to generate the set $\{\delta(J_1(f), \phi_{\alpha\beta\gamma})\}$.
2. Convert the computed set $\{\delta(J_1(f), \phi_{\alpha\beta\gamma})\}$ to the set $\{\delta(J_1(f), \tilde{\phi}_{\alpha\beta\gamma})\}$.

Since the conversion from $\{\delta(J_1(f), \phi_{\alpha\beta\gamma})\}$ to $\{\delta(J_1(f), \tilde{\phi}_{\alpha\beta\gamma})\}$ is straightforward, we focus our attention on the computation of $\delta(J_1(f), \phi_{\alpha\beta\gamma})$. We compute these scalar values using Parseval's theorem. Let $f_1(t)$ and $f_2(t)$ be two integrable functions in the $L^2$-sense, $F_1(\lambda)$ and $F_2(\lambda)$ the Fourier transforms of $f_1(t)$ and $f_2(t)$, respectively. Then Parseval's theorem yields

$$\int_{\mathbb{R}} f_1(t) f_2(t) \mathrm{d}t = \int_{\mathbb{R}} \bar{F}_1(\lambda) F_2(\lambda) \mathrm{d}\lambda,$$

where $\bar{F}_1$ stands for the complex conjugate of $F_1$. This theorem also holds for higher dimensional integrals. Let $\Phi_{\alpha\beta\gamma}$ be the three dimensional Fourier transform of $\phi_{\alpha\beta\gamma}$. Then by Parseval's theorem and the central slice theorem (see Appendix), we have

$$\begin{aligned}
\delta(J_1(f), \phi_{\alpha\beta\gamma}) =& 2\sum_{\mathbf{d}\in\mathbb{D}} \int_{\mathbb{R}^2} (X_{\mathbf{d}}(f) - g_{\mathbf{d}}) X_{\mathbf{d}} \phi_{\alpha\beta\gamma} \mathrm{d}u\mathrm{d}v \\
=& 2\sum_{\mathbf{d}\in\mathbb{D}} \int_{\mathbb{R}^2} \overline{\mathcal{F}_2\left(X_{\mathbf{d}}(f) - g_{\mathbf{d}}\right)} \mathcal{F}_2(X_{\mathbf{d}} \phi_{\alpha\beta\gamma}) \mathrm{d}u\mathrm{d}v \\
=& 2\sum_{\mathbf{d}\in\mathbb{D}} \int_{\mathbb{R}^2} \left( \overline{\mathcal{F}_3(f)}\Big|_{P_{\mathbf{d}}} - \overline{\mathcal{F}_2(g_{\mathbf{d}})} \right) \Phi_{\alpha\beta\gamma}\Big|_{P_{\mathbf{d}}} \mathrm{d}u\mathrm{d}v,
\end{aligned} \tag{5.4}$$

where $P_{\mathbf{d}}$ is the plane defined by $\{\mathbf{x} : \mathbf{x}^T\mathbf{d} = 0\}$, $\mathcal{F}_k$ stands for the k-dimensional Fourier transform. To speedup the computation of the double integral in (5.4), we utilize the decay property of $\Phi_{\alpha\beta\gamma}$. It follows from (A.1) that

$$\Phi_{\alpha\beta\gamma}(\omega_1, \omega_2, \omega_3) = h^3 e^{-ih(\alpha\omega_1 + \beta\omega_2 + \gamma\omega_3)} \prod_{i=1}^{3} \left( \frac{\sin(h\omega_i/2)}{h\omega_i/2} \right)^4.$$

Hence the magnitude of $\Phi_{\alpha\beta\gamma}(\omega_1,\omega_2,\omega_3)$ is bounded by $h^3\prod_{i=1}^3(\frac{1}{h|\omega_i|/2})^4$. Since the maximal value of $\Phi_{\alpha\beta\gamma}(\omega_1,\omega_2,\omega_3)$ is $h^3$, we therefore, restrict the range of $[\omega_1,\omega_2,\omega_3]$ in the following region.

$$\left(\frac{1}{h|\omega_i|/2}\right)^4 < \epsilon, \quad i=1,2,3, \qquad \prod_{i=1}^3\left(\frac{1}{h|\omega_i|/2}\right)^4 < \epsilon.$$

We define a restricted sub-domain $\Omega_\omega$ on which $\Phi_{\alpha\beta\gamma}(\omega_1,\omega_2,\omega_3)$ are evaluated and outside this domain the function is ignored.

$$\Omega_\omega = \left\{(\omega_1,\omega_2,\omega_3)\in\mathbb{R}^3:\ |\omega_i| < 2\epsilon^{-\frac{1}{4}}h^{-1} \ \text{ and } \ |\omega_1\omega_2\omega_3| < 8\epsilon^{-\frac{1}{4}}h^{-3}\right\}.$$

It is not difficult to calculate that the volume of $\Omega_\omega$ for $h=2$ is

$$V(\Omega_\omega) = 16\epsilon^{-\frac{1}{4}}\log(\epsilon^{-\frac{1}{4}})[1+\log(\epsilon^{-\frac{1}{4}})] + 8\epsilon^{-\frac{1}{4}}.$$

Comparing with the volume $8\epsilon^{-\frac{3}{4}}$ of the domain $\Omega_\epsilon = [-\epsilon^{-\frac{1}{4}},\epsilon^{-\frac{1}{4}}]^3$, the ratio of the volumes of the two domains is decreasing to zero. For instance, if $\epsilon=(2\pi)^{-4}$, the ratio is 0.289559. Hence, there is more than 70% computing time reduction by restricting the computation to the domain $\Omega_\omega$. In Table 5.1, we list $V(\Omega_\omega), V(\Omega_\epsilon)$ and the ratios of the two volumes for different $\epsilon$.

As we know, the approximation order of tri-cubic B-spline is $\mathcal{O}(h^4)=\mathcal{O}(n^{-4})$. Hence, if we require $\epsilon=\mathcal{O}(n^{-4})$, then ratio $V(\Omega_\omega)/V(\Omega_\epsilon)=\mathcal{O}(\log(n)/n)^2$.

Table 5.1: Volumes and Ratios of $\Omega_\omega$ and $\Omega_\epsilon$

| $\epsilon$ | $V(\Omega_\omega)$ | $V(\Omega_\epsilon)$ | $V(\Omega_\omega)/V(\Omega_\epsilon)$ |
|---|---|---|---|
| $10^0$ | 8.0 | 8.0 | 1.000 |
| $10^{-1}$ | 40.03 | 44.98 | 8.899e-1 |
| $10^{-2}$ | 150.61 | 252.98 | 5.953e-1 |
| $10^{-3}$ | 468.70 | 1244.62 | 3.294e-1 |
| $10^{-4}$ | 1296.71 | 8000.0 | 1.620e-1 |
| $10^{-5}$ | 3318.25 | 44987.30 | 7.376e-2 |

### 5.2.1. The first term of $\delta(J_1,\psi)$

Given a projection direction $\mathbf{d}$, the plane $P_{\mathbf{d}}$ is defined by

$$[\omega_1(\mathbf{u}),\omega_2(\mathbf{u}),\omega_3(\mathbf{u})]^T = u_1\mathbf{e}_{\mathbf{d}}^{(1)} + u_2\mathbf{e}_{\mathbf{d}}^{(2)}, \qquad \mathbf{u}=[u_1,u_2]^T\in\mathbb{R}^2.$$

Denote

$$\mathbf{e}_{\mathbf{d}}^{(1)} = [e_x^{(1)},e_y^{(1)},e_z^{(1)}]^T, \qquad \mathbf{e}_{\mathbf{d}}^{(2)} = [e_x^{(2)},e_y^{(2)},e_z^{(2)}]^T.$$

Without loss of generality, we may assume that the two dimensional vectors $[e_x^{(1)},e_y^{(1)}]^T$ and $[e_x^{(2)},e_y^{(2)}]^T$ are linearly independent. This assumption is surely satisfied for ET data, since there are no projections in $x$- or $y$-directions. Then

$$\mathbf{u} = E_{\mathbf{d}}^{-1}[\omega_1,\omega_2]^T \quad \text{with} \quad E_{\mathbf{d}} = \begin{bmatrix} e_x^{(1)} & e_x^{(2)} \\ e_y^{(1)} & e_y^{(2)} \end{bmatrix} \tag{5.5}$$

defines a linear transform between $u_1u_2$-plane and $\omega_1\omega_2$-plane, and

$$\mathrm{d}\mathbf{u} = \mathrm{d}u_1\mathrm{d}u_2 = \det(E_{\mathbf{d}}^{-1})\mathrm{d}\omega_1\mathrm{d}\omega_2, \quad \omega_3(\omega_1,\omega_2) = [e_z^{(1)}, e_z^{(2)}]E_{\mathbf{d}}^{-1}[\omega_1,\omega_2]^T.$$

Therefore, the first term of $\delta(J_1(f),\phi_{\alpha\beta\gamma})$, denoted by $T_{\alpha\beta\gamma}$, can be represented as

$$\begin{aligned} T_{\alpha\beta\gamma} =& 2\int_{\mathbb{R}^2}\left[\overline{\mathcal{F}_3(f)}\Phi_{\alpha\beta\gamma}\right]\Big|_{P_{\mathbf{d}}}\mathrm{d}\mathbf{u} \\ =& 2\int_{\mathbb{R}^2}\overline{\mathcal{F}_3(f)}(\omega_1(\mathbf{u}),\omega_2(\mathbf{u}),\omega_3(\mathbf{u}))\Phi_{\alpha\beta\gamma}(\omega_1(\mathbf{u}),\omega_2(\mathbf{u}),\omega_3(\mathbf{u}))\mathrm{d}\mathbf{u} \\ =& 2\int_{\mathbb{R}^2}\overline{\mathcal{F}_3(f)}(\omega_1,\omega_2,\omega_3(\omega_1,\omega_2))\Phi_{\alpha\beta\gamma}(\omega_1,\omega_2,\omega_3(\omega_1,\omega_2))\det(E_{\mathbf{d}}^{-1})\mathrm{d}\omega_1\mathrm{d}\omega_2 \\ =& 2h^3\det(E_{\mathbf{d}}^{-1})\int_{\mathbb{R}}e^{-ih\alpha\omega_1}\mathrm{d}\omega_1\int_{\mathbb{R}}Q(\omega_1,\omega_2)R_{\beta\gamma}(\omega_1,\omega_2)\mathrm{d}\omega_2, \end{aligned} \tag{5.6}$$

where

$$\begin{aligned} &Q(\omega_1,\omega_2) = \overline{\mathcal{F}_3(f)}(\omega_1,\omega_2,\omega_3(\omega_1,\omega_2))\left[\prod_{k=1}^{2}\frac{sin(h\omega_k/2)}{h\omega_k/2}\right]^4\left[\frac{sin(h\omega_3(\omega_1,\omega_2)/2)}{h\omega_3(\omega_1,\omega_2)/2}\right]^4, \\ &R_{\beta\gamma}(\omega_1,\omega_2) = e^{-ih(\beta\omega_2+\gamma\omega_3(\omega_1,\omega_2))}. \end{aligned}$$

Note that $Q(\omega_1,\omega_2)$ does not depend on $\alpha,\beta$ and $\gamma$, but depends on $f$. To calculate the integrations in (5.6), we define a uniform grid on the $\omega_1\omega_2$-plane, denoted by

$$G_{\mathbf{d}} = \left\{[\omega_1^{(j)},\omega_2^{(k)}]^T \in \mathbb{R}^2 : [\omega_1^{(j)},\omega_2^{(k)},\omega_3(\omega_1^{(j)},\omega_2^{(k)})]^T \in \Omega_\omega\right\},$$

where $\omega_1^{(j)} = \pi j/(2^d hL)$ and $\omega_2^{(k)} = \pi k/(2^d hL)$ are one dimensional uniform knots, $L = 2l+3$ is the number of B-spline basis used in the $x$ (and $y$) direction, and $d$ controls the density of the grid. Larger $d$ leads to a denser grid. We usually take $d=0$ or $d=1$. Here two zero B-spline coefficients are introduced in the $x$ and $y$ directions for satisfying periodic conditions. Then using the 2D version of the trapezoid quadrature rule, we obtain the following approximation

$$T_{\alpha\beta\gamma} \approx 2h^3\det(E_{\mathbf{d}}^{-1})\left(\frac{\pi}{2^d hL}\right)^2\sum_{j=m}^{M}e^{-ih\alpha\omega_1^{(j)}}\sum_{k=m_i}^{M_i}Q(\omega_1^{(j)},\omega_2^{(k)})R_{\beta\gamma}(\omega_1^{(j)},\omega_2^{(k)}). \tag{5.7}$$

We explain how (5.7) is used to compute $T_{\alpha\beta\gamma}$ efficiently. First consider the computation of $Q(\omega_1^{(j)},\omega_2^{(k)})$. Using the closed form of $\overline{\mathcal{F}_3(f)}(\omega_1,\omega_2,\omega_3(\omega_1,\omega_2))$, we have

$$Q(\omega_1^{(j)},\omega_2^{(k)}) = h^3 Q_1(\omega_1^{(j)},\omega_2^{(k)})Q_2(\omega_1^{(j)},\omega_2^{(k)}) \tag{5.8}$$

with

$$Q_1(\omega_1^{(j)},\omega_2^{(k)}) = \left[\frac{sin(h\omega_1^{(j)}/2)}{h\omega_1^{(j)}/2}\frac{sin(h\omega_2^{(k)}/2)}{h\omega_2^{(k)}/2}\frac{sin(h\omega_3(\omega_1^{(j)},\omega_2^{(k)})/2)}{h\omega_3(\omega_1^{(j)},\omega_2^{(k)})/2}\right]^8, \tag{5.9}$$

$$Q_2(\omega_1^{(j)},\omega_2^{(k)}) = \sum_\gamma e^{ih\gamma\omega_3(\omega_1^{(j)},\omega_2^{(k)})}Q_\gamma(\omega_1^{(j)},\omega_2^{(k)}), \tag{5.10}$$

where

$$Q_\gamma(\omega_1^{(j)},\omega_2^{(k)}) = \sum_\alpha\sum_\beta f_{\alpha\beta\gamma}e^{ih\alpha\omega_1^{(j)}}e^{ih\beta\omega_2^{(k)}}. \tag{5.11}$$

It is easy to see that $Q_\gamma(\omega_1^{(j)},\omega_2^{(k)})$ has the following periodic property:

$$Q_\gamma(\omega_1^{(j+s2^{d+1}L)},\omega_2^{(k+t2^{d+1}L)}) = Q_\gamma(\omega_1^{(j)},\omega_2^{(k)}), \qquad 0 \le j \le 2^{d+1}L-1,\ \ 0 \le k \le 2^{d+1}L-1,$$

for any integers $s$ and $t$. Using these formulations, $Q(\omega_1^{(j)},\omega_2^{(k)})$ can be computed as follows:

**Algorithm 5.1.** *Computing* $Q(\omega_1^{(j)},\omega_2^{(k)})$ , $[\omega_1^{(j)},\omega_2^{(k)}]^T \in G_{\mathbf{d}}$

1. For each $\gamma$, compute $Q_\gamma(\omega_1^{(j)},\omega_2^{(k)})$ using (5.11) by 2D discrete fast Fourier transform.
2. Compute $Q_2(\omega_1^{(j)},\omega_2^{(k)})$ using (5.10).
3. Compute $Q(\omega_1^{(j)},\omega_2^{(k)})$ using (5.8) and (5.9).

The computation in the second and third steps are straightforward. Details of the first step are as follows. Let

$$G_s = \left[0, \frac{\pi}{2^d hL}, \cdots, \frac{\pi(2^{d+1}L-1)}{2^d hL}\right] \times \left[0, \frac{\pi}{2^d hL}, \cdots, \frac{\pi(2^{d+1}L-1)}{2^d hL}\right],$$

and $\alpha = \alpha' - (l+1)$, $\beta = \beta' - (l+1)$. Then

$$\begin{aligned} Q_\gamma(\omega_1^{(j)},\omega_2^{(k)}) &= \sum_{\alpha=-l-1}^{l+1} \sum_{\beta=-l-1}^{l+1} f_{\alpha\beta\gamma} e^{ih\alpha\omega_1^{(j)}} e^{ih\beta\omega_2^{(k)}} \\ &= e^{-ih(l+1)(\omega_1^{(j)}+\omega_2^{(k)})} \sum_{\alpha'=0}^{L} \sum_{\beta'=0}^{L} f_{\alpha'-l-1,\beta'-l-1,\gamma} e^{ih\alpha'\omega_1^{(j)}} e^{ih\beta'\omega_2^{(k)}} \\ &= e^{-ih(l+1)(\omega_1^{(j)}+\omega_2^{(k)})} \overline{Q'_\gamma(\omega_1^{(j)},\omega_2^{(k)})} \end{aligned} \tag{5.12}$$

with

$$Q'_\gamma(\omega_1^{(j)},\omega_2^{(k)}) = \sum_{\alpha'=0}^{L} e^{-ih\alpha'\omega_1^{(j)}} \sum_{\beta'=0}^{L} f_{\alpha'-l-1,\beta'-l-1,\gamma} e^{-ih\beta'\omega_2^{(k)}}.$$

First compute $Q'_\gamma(\omega_1^{(j)},\omega_2^{(k)})$ using 2D fast Fourier transform for all $[\omega_1^{(j)},\omega_2^{(k)}]^T \in G_s$, where the involved coefficients with indices out of the range of $[0,L]$ are set to zero. Then compute $Q_\gamma(\omega_1^{(j)},\omega_2^{(k)})$ using (5.12) for $[\omega_1^{(j)},\omega_2^{(k)}]^T \in G_{\mathbf{d}}$. If a $[\omega_1^{(j)},\omega_2^{(k)}]^T \in G_{\mathbf{d}}\backslash G_s$, then $Q_\gamma(\omega_1^{(j)},\omega_2^{(k)})$ is obtained by the periodic properties.

It is easy to see that the cost of computing $Q_\gamma(\omega_1^{(j)},\omega_2^{(k)})$ in the first step is $\mathcal{O}(TL^2\log(L))$. The cost of the second step is $\mathcal{O}(TL^2)$. The cost of the last step is $\mathcal{O}(L^2)$. Taking the projections into account, the total cost is in the order of $\mathcal{O}(pTL^2\log(L))$. Here $p$, $T$ and $L$ stand for the number of projection directions, the number of the utilized B-spline basis $B_k(z)$ and the number of the utilized B-spline $B_i(x)$ (same as the number of $B_j(y)$), respectively. Note that $Q_1(\omega_1,\omega_2)$ does not depend on $\alpha$, $\beta$ and $\gamma$. Hence, it should be computed out of the $\alpha$, $\beta$ and $\gamma$ loops in Algorithm 5.1.

Now we consider the computation of $T_{\alpha\beta\gamma}$ using (5.7). We describe the computational steps involved via the following algorithm.

```
Algorithm 5.2. Computing $T_{\alpha\beta\gamma}$
  1. For each $\mathbf{d} \in \mathbb{D}$, compute $Q(\omega_1^{(j)}, \omega_2^{(k)})$ using Algorithm 4.1.
  2. For each integer $\gamma \in [-t, t]$, compute $U_\gamma(\omega_1^{(j)}, \omega_2^{(k)}) := Q(\omega_1^{(j)}, \omega_2^{(k)}) e^{-ih\gamma\omega_3(\omega_1^{(j)}, \omega_2^{(k)})}$.
  3. For each integer $\beta \in [-l, l]$, compute $V_{\beta\gamma}(\omega_1^{(j)}) := \sum_k U_\gamma(\omega_1^{(j)}, \omega_2^{(k)}) e^{-ih\beta\omega_2^{(k)}}$, using fast Fourier transform.
  4. For each integer $\alpha \in [-l, l]$, compute $h^3 \det(E_{\mathbf{d}}^{-1}) \left(\frac{4\pi}{n}\right)^2 \sum_j e^{-ih\alpha\omega_1^{(j)}} V_{\beta\gamma}(\omega_1^{(j)})$, using fast Fourier transform.
```

The usage of the fast Fourier transform in the third and the fourth step is similar to that of $Q_\gamma(\omega_1^{(j)}, \omega_2^{(k)})$ in (5.12). However, here it is one dimensional. Next we analyze the complexity of this algorithm. The cost of the first step has been analyzed, which is $\mathcal{O}(pTL^2 \log(L))$. The cost of step 2 is $\mathcal{O}(pTL^2)$. The costs of step 3 and step 4 are $\mathcal{O}(pTL^2 \log(L))$. The total cost of this algorithm is $\mathcal{O}(pTL^2 \log(L))$. Since $p$ and $T$ are small comparing with $L$, the algorithm is fast.

**Remark 5.1.** A direct computation of $T_{\alpha\beta\gamma}$ leads to a cost $\mathcal{O}(pTL^4)$. The above algorithms reduces the overall cost to $\mathcal{O}(pTL^2 \log(L))$, where $p$ and $T$ are considerably smaller than $L$.

### 5.2.2. The second term of $\delta(J_1, \psi)$

The second term in (5.4) can be computed together with the first term. At first the function $\mathcal{F}_2(g_d)$ is represented as

$$\mathcal{F}_2(g_d)(\mathbf{u}) = \mathcal{F}_2\Big(\sum_i \sum_j g_{\mathbf{d}}^{(ij)} N_i N_j\Big)(\mathbf{u}) = \sum_i \sum_j g_{\mathbf{d}}^{(ij)} B_i(u) B_j(v),$$

where $g_{\mathbf{d}}^{(ij)}$ are the coefficients of the B-spline representation of $g_{\mathbf{d}}$. A fast algorithm for converting the discrete data to the B-spline representation is adopted from [3, 20]. Let

$$S_{\alpha\beta\gamma} = \int_{\mathbb{R}^2} \overline{\mathcal{F}_2(g_{\mathbf{d}})} \Phi_{\alpha\beta\gamma}\Big|_{P_{\mathbf{d}}} \mathrm{d}\mathbf{u}.$$

Then for $S_{\alpha\beta\gamma}$, we have the same form expression as (5.6) that is for $T_{\alpha\beta\gamma}$,

$$S_{\alpha\beta\gamma} = h^3 \det(E_{\mathbf{d}}^{-1}) \int_{\mathbb{R}} e^{-ih\alpha\omega_1} \mathrm{d}\omega_1 \int_{\mathbb{R}} Q(\omega_1, \omega_2) R_{\beta\gamma}(\omega_1, \omega_2) \mathrm{d}\omega_2 \tag{5.13}$$

with the same $R_{\beta\gamma}(\omega_1, \omega_2)$, but a different $Q(\omega_1, \omega_2)$, which is

$$\begin{aligned} Q(\omega_1, \omega_2) =& \overline{\mathcal{F}_2(g_{\mathbf{d}})}(\mathbf{u}(\omega_1, \omega_2)) \left[\prod_{k=1}^{2} \frac{sin(h\omega_k/2)}{h\omega_k/2}\right]^4 \left[\frac{sin(h\omega_3(\omega_1, \omega_2)/2)}{h\omega_3(\omega_1, \omega_2)/2}\right]^4 \\ =& h^2 Q_1(\omega_1, \omega_2) Q_2(\mathbf{u}(\omega_1, \omega_2)), \end{aligned} \tag{5.14}$$

where

$$Q_1(\omega_1,\omega_2) = \left[\prod_{k=1}^{2} \frac{\sin(hu_k(\omega_1,\omega_2)/2)}{hu_k(\omega_1,\omega_2)/2} \frac{\sin(h\omega_k/2)}{h\omega_k/2}\right]^4 \left[\frac{\sin(h\omega_3(\omega_1,\omega_2)/2)}{h\omega_3(\omega_1,\omega_2)/2}\right]^4,$$

$$Q_2(u,v) = \sum_\alpha \sum_\beta g_{\mathbf{d}}^{(\alpha\beta)} e^{ih\alpha u_1} e^{ih\beta u_2},$$

and $\mathbf{u}(\omega_1,\omega_2)$ are given by (5.5). Algorithm 5.2 could be adjusted by subtracting the values $Q(\omega_1^{(j)},\omega_2^{(k)})$ of this $Q$ from that $Q$ in step 1. The values $Q(\omega_1^{(j)},\omega_2^{(k)})$ for this $Q$ is computed as follows.

**Algorithm 5.3.** *Computing* $Q(\omega_1^{(j)},\omega_2^{(k)})$

1. Compute $Q_2(u_1^{(j)},u_2^{(k)})$ over a uniform grid $G_s$ in the $uv$-plane using 2D discrete fast Fourier transform.
2. Compute $Q_2(u_1(\omega_1^{(j)},\omega_2^{(k)}),u_2(\omega_1^{(j)},\omega_2^{(k)}))$ from $Q_2(u_1^{(j)},u_2^{(k)})$ by the bilinear interpolation. If $\left[u_1(\omega_1^{(j)},\omega_2^{(k)}),u_2(\omega_1^{(j)},\omega_2^{(k)})\right]^T$ is out of the range of $G_s$, we use the periodical property of $Q_2(u,v)$ to generate the requited data.
3. Computing $Q(\omega_1^{(j)},\omega_2^{(k)})$ by (5.14).

The computation of $Q_2(u_1^{(j)},u_2^{(k)})$ in the first step is similar to that of $Q_\gamma(\omega_1^{(j)},\omega_2^{(k)})$ in (5.12). Hence, the cost of this step is $\mathcal{O}(pL^2\log(L))$, the cost of the second step is $\mathcal{O}(pL^2)$. Therefore, the computation cost of the $Q$ in (5.13) is much smaller than that $Q$ in (5.7).

**Remark 5.2.** Obviously, the computation of $\int_{\mathbb{R}^2}(X_{\mathbf{d}}f - g_{\mathbf{d}})X_{\mathbf{d}}\phi_{\alpha\beta\gamma}\mathrm{d}u\mathrm{d}v$ is our main task. This computation can also be conducted as follows basing on the central slice theorem:

1. Convert the discrete data of $f$ to the discrete data of $F$ in Fourier space using 3D discrete fast Fourier transform.
2. Take the slices of $F$ and $\Phi_{\alpha\beta\gamma}$ obtaining $\mathcal{F}_2(X_{\mathbf{d}}f)$ and $\mathcal{F}_2(X_{\mathbf{d}}\phi_{\alpha\beta\gamma})$.
3. Convert the slice datum to real space by the inverse 2D FFT, obtaining the data $X_{\mathbf{d}}f$ and $X_{\mathbf{d}}\phi_{\alpha\beta\gamma}$.
4. Compute the integral $\int_{\mathbb{R}^2}(X_{\mathbf{d}}f - g_{\mathbf{d}})X_{\mathbf{d}}\phi_{\alpha\beta\gamma}\mathrm{d}u\mathrm{d}v$ in real space.

The above computational strategy is also efficient. However, our experience shows that it is not accurate enough for our purpose. Our use of the discrete FFT is for fast summation only, and there is no error introduced except for the round-off errors of arithmetic operations.

## 5.3. Computation of $\delta(J_2,\psi)$

The computation of $\delta(J_2,\psi)$ is straightforward. Again the computation includes two steps:

1. Compute $\delta(J_2(f),\phi_{\alpha\beta\gamma})$ for all $\alpha,\beta,\gamma$.
2. Convert $\delta(J_2(f),\phi_{\alpha\beta\gamma})$ to $\delta(J_2(f),\tilde{\phi}_{\alpha\beta\gamma})$.

It is easy to see that, for a fixed function $f$, $\delta(J_2(f),\psi)$ is a linear function with respect to $\psi$. Hence the conversion from $\delta(J_2(f),\phi_{\alpha\beta\gamma})$ to $\delta(J_2(f),\tilde{\phi}_{\alpha\beta\gamma})$ is feasible. Therefore, the main task is the computation of $\delta(J_2(f),\phi_{\alpha\beta\gamma})$. Though there are $\mathcal{O}(TL^2)$ three dimensional integrations that need to be computed, using the local support property of the B-spline basis, the computation cost is easily reduced to $\mathcal{O}(TL^2)$.

For a uniform partitioning of $\Omega$, the integration is computed by evaluating and summing the integrand over the grid points, and then dividing by the volume of a voxel. Since $f$ is a $C^2$ continuous function, partial derivatives of $f$ up to the second order can also be computed exactly. Using these partial derivatives, the integrands in (4.3) or (4.2) can be easily calculated. For each grid point, there are only $3^3 = 27$ B-spline basis functions involved. Hence, the cost for evaluating one point but for all the $\phi_{\alpha\beta\gamma}$ is $\mathcal{O}(1)$. For all the grid points in $\Omega$, the cost is $\mathcal{O}(TL^2)$. Hence the overall computation is very efficient.

## 6. Numerical and Illustrative Results

In this section, we present several numerical and illustrative results, showing that the presented method is stable, reliable and robust. In all the examples presented, $\alpha$ in (4.1) is taken as one.

Table 6.1: $L^2$-errors for different initial functions.

| $g$ | $g=0$ | $g=\|\nabla f\|$ | $g=1$ |
|---|---|---|---|
| $L^2$-error | 2.365e-04 | 9.192e-05 | 1.426e-04 |

### 6.1. Stability experiments

In our stability experiments, we compute two steady solutions for two initial functions $f^{(0)}$ and using three different regularizers. a. $g(\mathbf{x}) = 0$ (means no regularizer); b. $g(\mathbf{x}) = \|\nabla f(\mathbf{x})\|$; c. $g(\mathbf{x}) = 1$. For each case, we compute the $L^2$-error of the two solutions from the two initial functions. The first initial function is taken as zero. The second one is taken as

$$f^{(0)}(\mathbf{x}) = \begin{cases} 0, & \|\mathbf{x}\| > 10 \text{ and } \mathbf{x} \in \Omega, \\ 1, & \|\mathbf{x}\| \le 10 \text{ and } \mathbf{x} \in \Omega. \end{cases}$$

In Table 6.1, we list the $L^2$-errors for each case after the same number (which is 8) iterations. From the table, we can see that the $L^2$-errors are around $10^{-4}$. Hence two different initial functions lead to almost the same solution. In this experiment, $12 \times 12$ uniformly distributed projection directions are taken. The function to be projected is defined as

$$f(x,y,z) = \begin{cases} 1, & \text{if } 25 < \sqrt{x^2+y^2+z^2} < 31, \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

**Remark 6.1.** If $g = 0$, $\min J(f)$ is the problem of least squares approximation. Due to the finite number of projection directions, the problem is ill-conditioned. However, the $L^2$-gradient flow still yields a reasonable solution. The reason is that the $L^2$-gradient flow searches for a minimizer in the negative gradient direction, where the energy functional $J(f)$ is decreasing. Though the minimizers may not be unique, the $L^2$-gradient flow will arrive at a minimizer.

## 6.2. Reliability experiments

In the experiments, given a function $f$ on a volume $\Omega$, we compute projection images from a set of uniformly distributed projection directions. Next we reconstruct $f$ using the projected data and using $g = 0$. On increasing the projection directions, we obtain a sequence of reconstructed functions. $L^2$-error between the reconstructed functions and the exact function $f$ are computed. Table 6.2 lists these $L^2$-errors after the same iteration number 50. The exact function to be projected is taken as

$$f(\mathbf{x}) = \sum_{i=1}^{10} \exp^{-(\|\mathbf{x}-\mathbf{x}_i\|^2 - r_i^2)}, \tag{6.2}$$

where $[\mathbf{x}_i, r_i]$ are taken as $[16.0, 27.0, 26.0, 20.0]$, $[19.2, 35.0, 32.0, 20.0]$, $[22.4, 43.0, 38.0, 20.0]$, $[25.6, 31.0, 31.0, 20.0]$, $[28.8, 39.0, 37.0, 24.0]$, $[32.0, 27.0, 30.0, 24.0]$, $[35.2, 35.0, 36.0, 24.0]$, $[38.4, 43.0, 29.0, 24.0]$, $[41.6, 31.0, 35.0, 28.0]$, $[44.8, 39.0, 28.0, 28.0]$. From the table, we can see that the errors decrease as the projection directions increase.

Table 6.2: $L^2$-errors between reconstructed functions and exact function.

| Projection # | $3 \times 3$ | $6 \times 6$ | $12 \times 12$ | $24 \times 24$ |
|---|---|---|---|---|
| $L^2$-error | 0.0127 | 0.0072 | 0.0061 | 0.0059 |

## 6.3. Robustness experiments

These experiments are similar to the reliability experiments. However, noise is added to each of the projected images. The aim is to see how the noise affects the reconstructed results. Table 6.3 lists the $L^2$-errors, now in the presence of image noise. Here the exact function to be projected is defined by (6.1). In order to add Gaussian noise with the given signal-to-noise ratio (SNR), we use the xmipp [19] package to compute the standard deviation for each given SNR value and then use the standard deviation to add Gaussian noise to each projection. For instance, for a SNR of 0.33, a noise with a standard deviation of 15.2614 is needed. All tests in this experiment have the same iteration number 7. From the table, we can see that the $L^2$ errors are quite small and the errors decrease as SNR increases.

Table 6.3: $L^2$-errors between reconstructed functions from noised data and exact function.

| Projection # | $3 \times 3$ | $6 \times 6$ | $12 \times 12$ | $24 \times 24$ |
|---|---|---|---|---|
| SNR=0.33 | 0.056 | 0.0419 | 0.0431 | 0.0247 |
| SNR=1.0 | 0.026 | 0.0179 | 0.0196 | 0.0126 |
| SNR=10.0 | 0.012 | 0.0116 | 0.0089 | 0.0074 |

## 6.4. Comparative Experiments using Phantom Example

To illustrate the efficiency of the proposed method, we compare the performance of our method with ART, SIRT and WBP using phantom data. The function $f(x, y, z)$ to be projected is (6.1). The projected images are obtained by projecting $f$ on 55 different directions (tilt series). The projection angles are chosen as follows: In the range $[-45°, 45°]$, 31 projections are taken with 3 degree increments. In each of the ranges $[45°, 69°]$ and $[-69°, -45°]$, 12 projections are

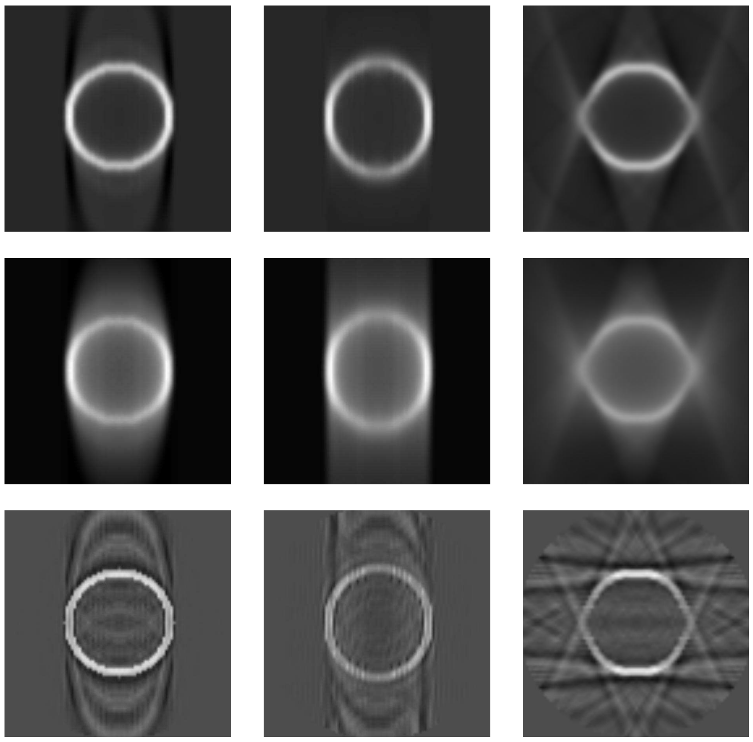

Fig. 6.1. The first, second and third rows show the slices of the reconstructed functions using ART, SIRT and WBP, respectively. The first, second and third columns are the slices on $XY$-plane, $XZ$-plane and $YZ$-plane, respectively.

taken with 2 degree increments. Hence, there are no projections in the ranges $(69°, 90°]$ and $[-90°, -69°)$ (missing wedge).

The three rows of the Fig. 6.1 show the slices of the reconstructed functions using ART, SIRT and WBP, where the first, second and third columns are the slices on $XY$-plane, $XZ$-plane and $YZ$-plane, respectively. Fig. 6.2 shows the reconstruction results of our method. The slices of the first, second and third rows are extracted from the reconstructed functions using $g = 0$, $g = \|\nabla f\|$ and $g = 1$ , respectively. The first, second and third columns are the slices on $XY$-plane, $XZ$-plane and $YZ$-plane, respectively. The grey scale in these slices is taken as black to white for density value from zero to one. From these figures we can see that our results are better in general than the results from ART, SIRT and WBP. The best results of our method is for $g = 1$. When $g = 1$, the derived partial differential equation from $J_2(f)$ is the mean curvature flow in the level set form. It is well know that the mean curvature flow has an area-shrinking property of the evolved surface defined by the level-set of the evolved $f$ and it therefore has very strong regularization effect (see [18, 20]). Fig. 6.3 shows the results of volume rendering.

## 6.5. Application to the AIDS virus tilt series

The input is a set of 2D projection slices at different tilt angles taken from an electron microscope of AIDS virus interacting with a neutralizing molecule D1D2-IgP [4]. Each 2D image is of size $512 * 512$ and are imaged with a defocus of $-8\mu m$. Preprocessing this tilt series includes alignment and image enhancement. The 2D projections are reconstructed into 3D

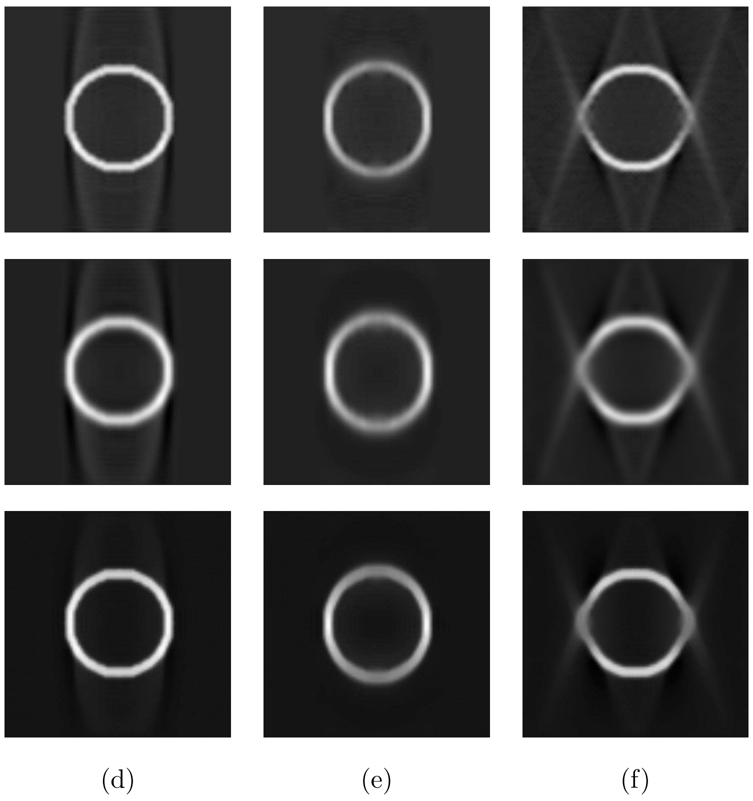

Fig. 6.2. The first, second and the third rows are extracted slices from the reconstructed functions using $g = 0$, $g = \|\nabla f\|$ and $g = 1$, respectively. The first, second and third columns are the slices on $XY$-plane, $XZ$-plane and $YZ$-plane, respectively.

volumes (also called Maps) using an $L^2$-gradient flow. In our data the number of projections images are 50 in tilt angular increments from $-60°$ to $+60°$, with each projection image of size $512 \times 512$. The tilt angle steps are $2° \sim 3°$.

Fig. 6.4 shows the slices of the constructed volumetric function, where the the first, second and third figures in the first row show the slices of the reconstructed functions using the methods of ART, SIRT and WBP, respectively. The first, second and third figures in the second row show the slices of the reconstructed functions using our methods with $g = 0$, $g = \|\nabla f\|$ and $g = 1$, respectively. All the slices are taken in the $XY$-plane. A 3D volume rendering of the reconstructed AIDS virus is shown in the figure 6.5. Reconstruction results with different kinds of regularizers and without regularizer are shown.

We perform a comparison of local contrast (LC) and signal-to-noise ratio (SNR) of the reconstructed volumes from our method with ART, SIRT and WBP. To quantify the differences of the reconstructions and compare the SNR from various reconstruction techniques, a sub-volume of the volume is extracted containing the same feature (virus spike) in all three reconstruction schemes. Spikes on the virus envelope could be observed (see Figs. 6.6 and 6.7). SNR for an image is given by:

$$\mathrm{SNR} = \frac{\mu_{\text{foreground}} - \mu_{\text{background}}}{\sigma_{\text{background}}} \tag{6.3}$$

where $\mu_{\text{foreground}}$ is the average foreground intensity value in a window placed on the virus, $\mu_{\text{background}}$ is the average background intensity and $\sigma_{\text{background}}$ is the standard deviation of the

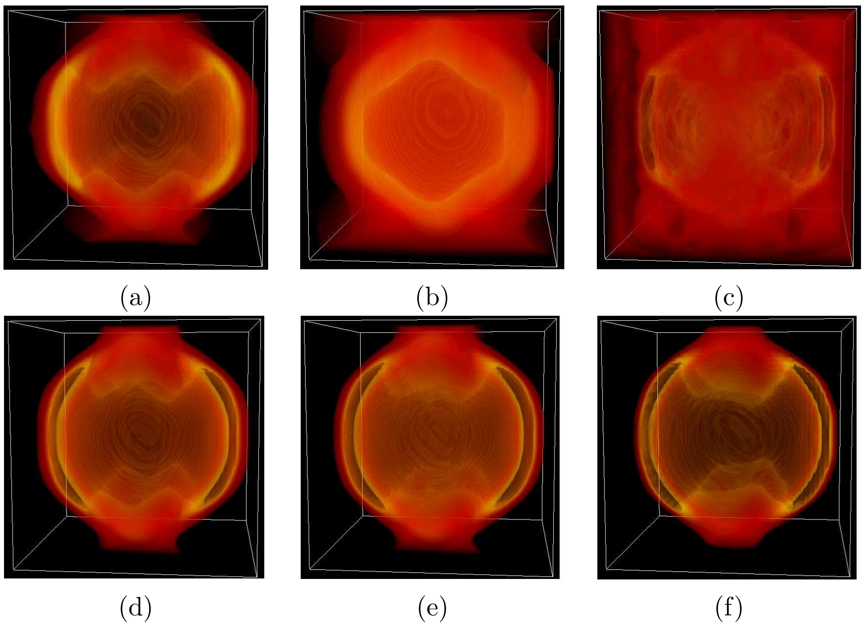

Fig. 6.3. Volume rendering: The first, second and third figures in the first row show the rendering results of the reconstructed functions using ART, SIRT and WBP. The first, second and third figures in the second row show the rendering results of the reconstructed functions using our methods with $g = 0$, $g = \|\nabla f\|$ and $g = 1$, respectively.

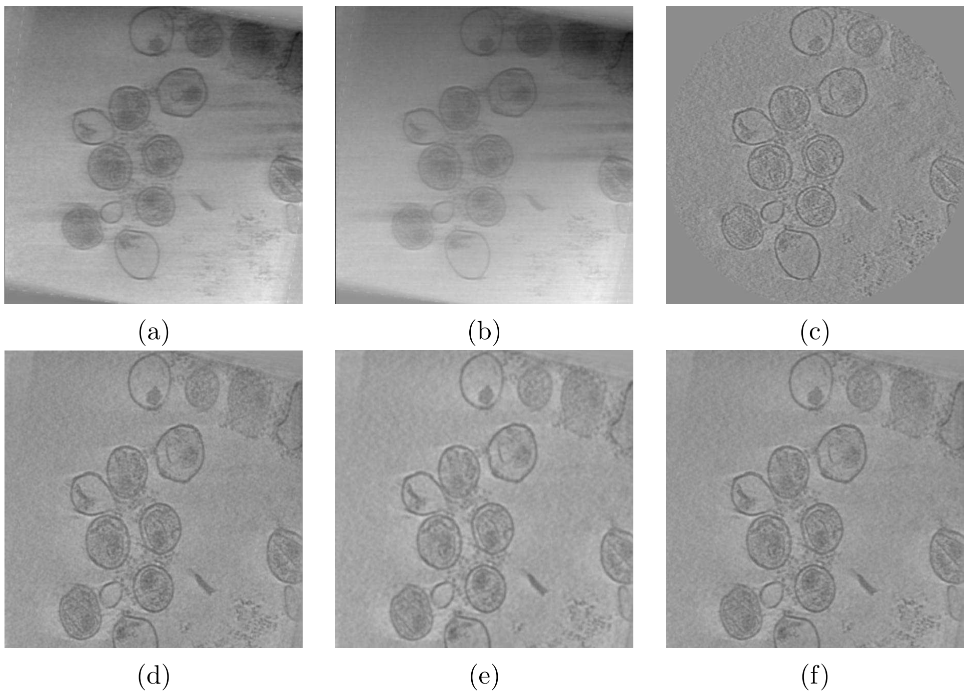

Fig. 6.4. The first, second and third figures in the first row show the slices of the reconstructed functions using ART, SIRT and WBP. The first, second and third figures in the second row show the slices of the reconstructed functions using our methods with $g = 0$, $g = \|\nabla f\|$ and $g = 1$, respectively. All the slices are taken in the $XY$-plane.

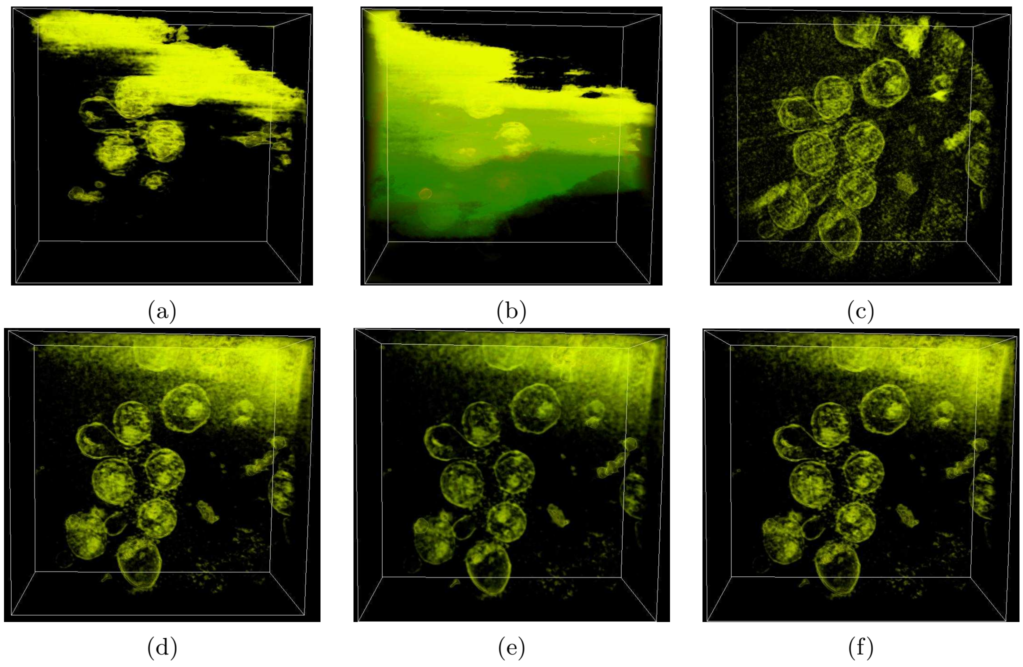

Fig. 6.5. Volume rendering: The first, second and third figures in the first row show the rendering results of the reconstructed functions using ART, SIRT and WBP. The first, second and third figures in the second row show the rendering results of the reconstructed functions using our methods with $g = 0$, $g = 1$ and $g = \|\nabla f\|$, respectively.

background measured in a window placed in the background of the image as shown in Fig. 6.7.

To compare the local contrast between reconstruction techniques, the ratio of the difference between average foreground ($\mu_{\text{foreground}}$) and average background ($\mu_{\text{background}}$) to the average background ($\mu_{\text{background}}$) intensity is calculated in the window shown in Fig. 6.7.

$$\text{LC} = \frac{\mu_{\text{foreground}} - \mu_{\text{background}}}{\mu_{\text{background}}} \tag{6.4}$$

Table 6.4: The table shows the results of SNR and LC analysis on the reconstructed Maps.

| Reconstruction Method | SNR | LC |
|---|---|---|
| ART | 12.512 | 0.0894 |
| SIRT | 24.11096 | 0.05932 |
| WBP | 7.8138 | 0.267 |
| $g = 1$ | 9.144 | 0.3534 |
| $g = \|\nabla f\|$ | 10.049 | 0.4275 |
| $g = 0$ | 8.8968 | 0.3514 |

For LC:

$$(g = \|\nabla f\|) > (g = 1) > (g = 0) > \text{WBP} > \text{ART} > \text{SIRT}.$$

For SNR:

$$\text{SIRT} > \text{ART} > (g = \|\nabla f\|) > (g = 1) > (g = 0) > \text{WBP}.$$

LC of SIRT reconstruction is very low, but the signal to noise ratio is very good. This means that there is low contrast but the overall image has less noise. High LC ensures that segmentation and structure identification will be good and features are easy to identify from background. High SNR means that background region is fairly uniform and low in intensity and hence segmentation and structure identification will be cleaner. Ideally we would like both LC and SNR to be high. The local contrast (LC) of the reconstructed volumes from our method is superior compared to WBP, ART and SIRT.

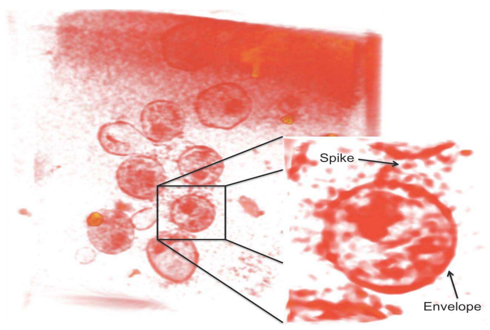

Fig. 6.6. A zoom in of a part Fig 6.5(e) that contains spikes on the virus envelope.

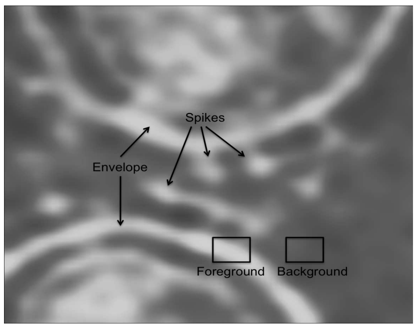

Fig. 6.7. Windows are placed on the foreground (virus envelope) and background, respectively.

## 7. Conclusions

We have presented a variational approach for ET data reconstruction. From minimizing an energy functional consisting of a fidelity term and a regularization term, an $L^2$-gradient flow has been constructed. An efficient computational method for solving the derived PDE is developed. The stability, reliability and robustness of the proposed method have been tested using phantom data as well as acquired ET data of the AIDS virus. The numerical experiments have shown that our method is efficient and effective. The theoretic analysis given in [22] also shows that the presented method is convergent.

## A. Miscellaneous Important Facts

In this appendix, we introduce the necessary material used in the paper.

### B-spline function

Let rect$(t)$ denote the rectangular function defined as

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

Then the degree $n$ B-spline basis $\beta_0^n(x)$ with support $[-\frac{n+1}{2}, \frac{n+1}{2}]$ is defined recursively by

$$\beta_0^n(x) = \beta_0^{n-1}(x) * \mathrm{rect}(x),$$

where $*$ denotes the convolution of two functions, and $\beta_0^0(x) = \mathrm{rect}(x)$. Other basis functions associate with the integer $k$ is defined by the shifting of $\beta_0^n(x)$, i.e.,

$$\beta_k^n(x) = \beta_0^n(x-k), \quad k = 0, \pm 1, \pm 2, \cdots .$$

By scaling the independent variable $x$ of $\beta_k^n(x)$, we obtain the B-spline basis on equi-spaced knots

$$\cdots, \quad -3h, \quad -2h, \quad -h, \quad 0, \quad h, \quad 2h, \quad 3h, \quad \cdots$$

as

$$N_k^n(x) := \beta_k^n(h^{-1}x) = \beta_0^n(h^{-1}x - k),$$

where $a > 0$. The support of $N_k^n(x)$ is $[h(k - \frac{n+1}{2}), h(k + \frac{n+1}{2})]$. For cubic B-spline used in this paper, we use notation $N_k(x)$ to represent $N_k^3(x)$.

**Fourier transform of B-spline function**

Let $\mathcal{F}_1$ denote the one dimensional Fourier transform defined as

$$F(\omega) = (\mathcal{F}_1 f)(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt.$$

Then it is easy to see that

$$\mathcal{F}_1(f(at))(\omega) = \frac{1}{|a|} F(\frac{\omega}{a}), \quad \mathcal{F}_1(f(t+a))(\omega) = e^{i\omega a} F(\omega).$$

Then by $\mathcal{F}_1 \beta_0^0 = \frac{\sin(\omega/2)}{\omega/2}$, we have

$$B_k^n(\omega) = (\mathcal{F}_1 N_k^n)(\omega) = h e^{-ik\omega h} \left( \frac{\sin(h\omega/2)}{h\omega/2} \right)^{n+1}, \quad k = 0, \pm 1, \cdots . \tag{A.1}$$

Note that $B_k^n(\omega)$ is not locally supported, but it goes to zero with the rate $\omega^{-(n+1)}$ as $|\omega| \to \infty$. Let $\mathcal{F}_k$ stand for the k-dimensional Fourier transform, i.e.,

$$F(\mathbf{X}) = (\mathcal{F}_k f)(\mathbf{X}) = \int_{\mathbb{R}^k} f(\mathbf{x}) \exp\left(-i\mathbf{x}^T\mathbf{X}\right) \mathrm{d}\mathbf{x}, \quad \mathbf{x}, \mathbf{X} \in \mathbb{R}^k.$$

Let $N_i(x)$, $N_j(y)$ and $N_k(z)$ be the B-spline basis in the $x$, $y$ and $z$ directions, respectively, and $\phi_{ijk}(\mathbf{x}) = N_i(x)N_j(y)N_k(z)$ their tensor product. Then

$$\begin{aligned}
\Phi_{ijk}(\mathbf{X}) &= \mathcal{F}_3(\phi_{ijk})(\mathbf{X}) \\
&= \int_{\mathbb{R}^3} N_i(x)N_j(y)N_k(z) \exp\left(-i(xX + yY + zZ\right) \mathrm{d}x\mathrm{d}y\mathrm{d}z \\
&= \mathcal{F}_1(N_i)\mathcal{F}(N_j)\mathcal{F}_1(N_k) \\
&= B_i^{(1)}(X) B_j^{(2)}(Y) B_k^{(3)}(Z).
\end{aligned}$$

**Fourier transform and discrete Fourier transform**

Let $f(t)$ be a continuous function defined on the interval $[a,b]$ with the periodic condition $f(a) = f(b)$. Let the Fourier transform of $f$ be defined as

$$F(\omega) = (\mathcal{F}_1 f)(\omega) = \frac{1}{\sqrt{2\pi}} \int_a^b f(t) e^{-i\omega t} dt.$$

Let $t = a + (b-a)\theta/2\pi$, then we have

$$\begin{aligned} F(\omega) =& \frac{b-a}{2\pi\sqrt{2\pi}} \int_0^{2\pi} f(a+(b-a)\theta/2\pi) e^{-i\omega(a+(b-a)\theta/2\pi)} d\theta \\ =& \frac{b-a}{2\pi\sqrt{2\pi}} e^{-i\omega a} \int_0^{2\pi} g(\theta) e^{-i\omega((b-a)\theta/2\pi)} d\theta, \end{aligned}$$

where $g(\theta) = f(a+(b-a)\theta/2\pi)$. Now sample $F(\omega)$ uniformly at

$$\omega_k = \frac{2k\pi}{b-a}, \quad k = -K, \cdots, -1, 0, 1, \cdots, K,$$

where $K$ is a given integer. If $a = -n/2, b = n/2, W = 2\pi$, we choose

$$K = \frac{nW}{2\pi} = n.$$

We then have

$$\begin{aligned} F(\omega_k) =& \frac{b-a}{2\pi\sqrt{2\pi}} e^{-i\omega_k a} \int_0^{2\pi} g(\theta) e^{-ik\theta} d\theta \\ \approx& \frac{b-a}{n\sqrt{2\pi}} e^{-i\omega_k a} \sum_{j=0}^{n-1} g\left(\frac{2j\pi}{n}\right) e^{-\frac{2ijk\pi}{n}} \\ =& \frac{2b}{n\sqrt{2\pi}} (-1)^k G_k(g), \end{aligned}$$

where the assumptions $b > 0$ and $a = -b$ are imposed, and

$$G_k(g) = \sum_{j=0}^{n-1} g\left(\frac{2j\pi}{n}\right) e^{-\frac{2ijk\pi}{n}}, \qquad k = -n, \cdots, -1, 0, 1, \cdots, n.$$

$G_0(g), \cdots, G_{n-1}(g)$ are discrete Fourier transform of $g$, which can be computed by the discrete fast Fourier transform, $G_n(g) = G_0(g)$. For $k = -n, -n+1, \cdots, -1$, let $\gamma = k+n$, then it is easy to see that

$$G_k(g) = G_\gamma(g).$$

Let $g = g_r + ig_i$, where $g_r$ and $g_i$ are the real and imaginary parts of $g$. Then

$$\begin{aligned} G_k(g) &= G_k(g_r) + iG_k(g_i), \qquad & k = 0, \cdots, n, \\ G_k(g) &= G_{-k}(g_r) + iG_{-k}(g_i), \qquad & k = -n, \cdots, 0, \end{aligned}$$

$G_{-K}, \cdots, G_1$ are the conjugate of $G_K, \cdots, G_1$, respectively. For the two-dimensional Fourier transform

$$F(\omega, \varpi) = \frac{1}{2\pi} \int_a^b \int_c^d f(s,t) e^{-i(\omega s + \varpi t)} \mathrm{d}s\mathrm{d}t,$$

we similarly have

$$F(\omega_{k_1}, \varpi_{k_2}) \approx \frac{(b-a)(d-c)}{2\pi n_1 n_2} e^{-i(\omega_{k_1} a + \varpi_{k_2} c)} G_{k_1 k_2},$$

where

$$G_{k_1k_2} = \sum_{j=0}^{n_1-1}\sum_{k=0}^{n_2-1} f\left(a + \frac{j(b-a)}{n_1}, c + \frac{k(d-c)}{n_2}\right) e^{-2\pi i\left(\frac{jk_1}{n_1}+\frac{kk_2}{n_2}\right)},$$

$$\omega_{k_1} = \frac{2k_1\pi}{b-a}, \quad \varpi_{k_2} = \frac{2k_2\pi}{d-c}, \qquad k_1, k_2 = -n, \cdots, -1, 0, 1, \cdots, n-1.$$

For $k_1, k_2 \geq 0$, $G_{k_1k_2}$ are computed by discrete fast Fourier transform. For $k_1 < 0$ or $k_2 < 0$,

$$G_{k_1k_2} = G_{\gamma_1\gamma_2}, \qquad \text{with} \ \ \gamma_1 = k_1(\text{mod } n_1), \ \ \gamma_2 = k_2(\text{mod } n_2) \tag{A.2}$$

If we assume $a = -b, \ c = -d$, then

$$F(\omega_{k_1}, \varpi_{k_2}) \approx \frac{2bd}{\pi n_1 n_2}(-1)^{k_1+k_2} G_{k_1k_2},$$

**Central Slice Theorem of Fourier Transform**

Let $f(x,y,z)$ be a function defined in $\mathbb{R}^3$,

$$(X_z f)(x,y) := \int_{\mathbb{R}} f(x,y,z)\mathrm{d}z,$$

where $X_z$ denotes the projection of $f$ in the $z$-direction. Then

$$(\mathcal{F}_2(X_z f))(X,Y) \ = \ (\mathcal{F}_3 f)(X,Y,Z)|_{Z=0}\,.$$

This equality says that the 2D Fourier transform of the projection in the $z$-direction of $f$ is the same as the slice of the 3D Fourier transform of $f$ at $Z = 0$. This fact, named as central slice theorem (see [16], page 10), implies that if we know all the projections, then we know all the slices, and therefore, we know $F(X,Y,Z)$. By applying the inverse of the 3D Fourier transform to $F$, we obtain $f(x,y,z)$.

## B. The Existence and Uniqueness of Solution

In this section, we consider the existence and uniqueness problem of the minimization problem of (4.1). Let $\mathcal{X}$ be the function space consists of B-spline functions on the domain $\Omega$ satisfying the zero boundary conditions:

$$\frac{\partial^{i+j+k} f(x,y,z)}{\partial^i x \partial^j y \partial^k z} = 0, \qquad \forall \mathbf{x} = [x,y,z]^T \in \Gamma_\Omega, \ \ i+j+k \leq 2.$$

Under the inner product

$$\langle f, g\rangle = \int_\Omega f(\mathbf{x}) g(\mathbf{x}) \mathrm{d}\mathbf{x},$$

$\mathcal{X}$ is a finite dimensional Hilbert space. Therefore $\mathcal{X}$ is reflective.

Since the projection directions are limited, the energy functional

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

may not be coercive (see [2] for the definition of coercive). To illustrate this, let $\mathbf{d}$ be a direction in the $x$-direction. Then for any coefficient $f_{ijk}$ satisfying

$$\sum_i f_{ijk} = 0, \quad \forall j, k,$$

we have $X_{\mathbf{d}} f = 0$. Hence, $\|f\| \to \infty$, does not imply $\int_{\mathbb{R}^2} (X_{\mathbf{d}} f - g_{\mathbf{d}})^2 \, \mathrm{d}u\mathrm{d}v \to \infty$. However, if

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

is coercive, then $J(f) = J_1(f) + \alpha J_2(f)$ is coercive for $\alpha > 0$.

Now let us prove that $J_2(f)$ is coercive. Let $f \in \mathcal{X}$. Then if $f(\mathbf{x}) = c$ for all $\mathbf{x} \in \Omega$, then $c = 0$. This implies that if $f(\mathbf{x}) \neq 0$, then $\nabla f(\mathbf{x}) \equiv 0$ does not hold. This further implies that

$$\int_\Omega \|\nabla f(\mathbf{x})\|^2 \mathrm{d}\mathbf{x} > 0. \tag{B.1}$$

Suppose $f(\mathbf{x}) = \sum_{i=1}^N c_i \varphi_i(\mathbf{x})$, where $\{\varphi_1, \cdots, \varphi_N\}$ is a basis set of $\mathcal{X}$. Then we have from (B.1) that the matrix

$$\mathbf{M} = \left[ \int_\Omega (\nabla \varphi_i(\mathbf{x}))^T \nabla \varphi_j(\mathbf{x}) \mathrm{d}\mathbf{x} \right]_{ij=1}^N$$

is nonsingular. Consider the case where $g(\mathbf{x}) = \|\nabla f(\mathbf{x})\|$. Then

$$J_2(f) = \int_\Omega \|\nabla f(\mathbf{x})\|^2 \mathrm{d}\mathbf{x} = \mathbf{C}^T \mathbf{M} \mathbf{C},$$

where $\mathbf{C} = [c_1, \cdots, c_N]^T$. Since $\mathbf{M}$ is non-singular, $J_2(f) \to \infty$ with the rate $\mathcal{O}(\|\mathbf{C}\|^2)$ as $\|\mathbf{C}\| \to \infty$. Therefore, $J_2(f)$ is coercive. Now let us consider the case $g(\mathbf{x}) = 1$. Since

$$\int_\Omega \|\nabla f(\mathbf{x})\|^2 \mathrm{d}\mathbf{x} \le \|\nabla f\|_\infty \int_\Omega \|\nabla f(\mathbf{x})\| \mathrm{d}\mathbf{x},$$

we have

$$J_2(f) = \int_\Omega \|\nabla f(\mathbf{x})\| \mathrm{d}\mathbf{x} \ge \frac{1}{\|\nabla f\|_\infty} \int_\Omega \|\nabla f(\mathbf{x})\|^2 \mathrm{d}\mathbf{x}.$$

Noticing that $\|\nabla f\|_\infty = \mathcal{O}(\|\mathbf{C}\|)$ as $\|\mathbf{C}\| \to \infty$, we have that $J_2(f)$ is coercive. Since $\mathcal{X}$ is compact, the discussion above implies the existence of the minimization problem (4.1) for $g = 1$ and $g = \|\nabla f\|$.

For the uniqueness of the solution, it is quite easy to prove that if $g = \|\nabla f\|$, the minimization problem (4.1) is quadratic, and the Euler-Lagrange equation is a non-singular linear system.

**Acknowledgments.** A part of this work was done when Guoliang Xu was visiting Chandrajit Bajaj at UT-CVC. His visit was additionally supported by the J. T. Oden Faculty Fellowship Research Program at ICES. The AIDS virus data for this paper was provided by Dr. Sriram Subramaniam from the National Cancer Institute, National Institutes of Health. Guoliang Xu and Ming Li were supported in part by NSFC under the grant 60773165, NSFC key project under the grant 10990013 and Funds for Creative Research Groups of China (grant No. 11021101). Chandrajit Bajaj and Ajay Gopinath were supported in part by NIH grants R01-EB00487, R01-GM074258 and grant from the UT-Portugal COLAB project.

## References

[1] M. Abramowitz and I.A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Dover Publications, Inc., 1972.

[2] G. Aubert and P. Kornprobst, Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations, Springer, 2000.

[3] C.L. Bajaj, G. Xu, and Q. Zhang, High-order level-set method and its application in biomolecular surface construction, *J. Comput. Sci Technol.*, **23**:6 (2008), 1026-1036.

[4] A. Bennett, J. Liu, D.V. Ryk, D. Bliss, J. Arthos, R.M. Henderson, and S. Subramaniam, Cryoelectron tomographic analysis of an hiv-neutralizing protein and its complex with native viral gp120, *J. Biol. Chem.*, **282** (2007).

[5] J. Carazo, G.T. Herman, C.O.S. Sorzano, and R. Marabini, Algorithms for three-dimensional reconstruction from the imperfect projection data provided by electron microscopy, In *Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell*, Chapter 7, pp. 217–244. Springer, 2nd ed., 2006.

[6] E.W. Cheney, Introduction to Approximation Theory, McGraw-Hill Book. Co., New York, 1966.

[7] L.C. Evans and R.F. Gariepy, Measure Theory and Fine Properties of Functions, CRC Press, 1992.

[8] D Fanelli and O. Ö Oktem, Electron tomography: a short overview with an emphasis on the absorption potential model for the forward problem, *Inverse Probl.*, **24** (2008).

[9] Joachim Frank, Introduction: Principles of electron tomography, In *Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell*, Chapter 1. Springer, 2nd ed., 2006.

[10] P. Gilbert, Iterative methods for the three-dimensional reconstruction of an object from projections, *J. Theor. Biol.*, **36** (1972).

[11] G.T. Herman, Image Reconstructions from Projections: The Fundamentals of Computerized Tomography, Computer Science and Applied Mathematics, 1980.

[12] S Kaczmarz, Angenäherte auflösung von systemen linearer gleichungen, *Bulletin International de l'Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques*, **35** (1937), 355-357.

[13] A.C. Kak and M. Slaney, Principles of Computerized Tomographic Imaging, IEEE Press, 1988.

[14] A.J. Koster and M. Barcena, Cryotomography: Low-dose automated tomography of frozen-hydrated specimens, In *Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell*, Chapter 4. Springer, 2nd ed., 2006.

[15] J. Liu, A. Bartesaghi, M.J. Borgnia, G. Sapiro, and S. Subramaniam, Molecular architecture of native hiv-1 gp120 trimers, *Nature*, September 2008.

[16] Frank Natterer and Frank Wübbeling, *Mathematical Methods in Image Reconstruction*, SIAM, 2001.

[17] M. Radermacher, Weighted back-projection methods, In *Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell*, Chapter 8, pp. 245-274, Springer, 2nd ed., 2006.

[18] G. Sapiro, Geometric Partial Differential Equations and Image Analysis, Cambridge, University Press, 2001.

[19] C.O.S. Sorzano, R. Marabini, J. Velazquez-Muriel, J.R. Bilbao-Castroa, S.H.W. Scheresa, J.M. Carazo, and A. Pascual-Montano, Xmipp: a new generation of an open-source image processing package for electron microscopy, *J. Struct. Biol.*, **148** (2004), 194-204.

[20] G. Xu, Geometric Partial Differential Equation Methods in Computational Geometry, Science Press, Beijing, China, 2008.

[21] G. Xu and Y. Shi, Progressive computation and numerical tables of generalized Gaussian quadrature formulas, *Journal on Numerical Methods and the Computer Application*, **27**:1 (2006), 9-23.

[22] C. Chen and G. Xu, Inversion of Electron Tomography Images Using $L^2$-Gradient Flows, Convergence Analysis, Technical Report ICM-11-07.