Gridding-based direct Fourier inversion of the three-dimensional ray transform

Pawel A. Penczek
Department of Biochemistry and Molecular Biology, The University of Texas–Houston Medical School, 6431 Fannin, MSB6.218, Houston, Texas 77030

Robert Renka
Department of Computer Science, The University of North Texas, Denton, Texas 76203

Hermann Schomberg
Philips Research Hamburg, Röntgenstrasse 24, 22335 Hamburg, Germany

Received June 4, 2003; revised manuscript received November 18, 2003; accepted November 20, 2003

We describe a fast and accurate direct Fourier method for reconstructing a function $f$ of three variables from a number of its parallel beam projections. The main application of our method is in single particle analysis, where the goal is to reconstruct the mass density of a biological macromolecule. Typically, the number of projections is extremely large, and each projection is extremely noisy. The projection directions are random and initially unknown. However, it is possible to determine both the directions and $f$ by an iterative procedure; during each stage of the iteration, one has to solve a reconstruction problem of the type considered here. Our reconstruction algorithm is distinguished from other direct Fourier methods by the use of gridding techniques that provide an efficient means to compute a uniformly sampled version of a function $g$ from a nonuniformly sampled version of $Fg$, the Fourier transform of $g$, or vice versa. We apply the two-dimensional reverse gridding method to each available projection of $f$, the function to be reconstructed, in order to obtain $Ff$ on a special spherical grid. Then we use the three-dimensional gridding method to reconstruct $f$ from this sampled version of $Ff$. This stage requires a proper weighting of the samples of $Ff$ to compensate for their nonuniform distribution. We use a fast method for computing appropriate weights that exploits the special properties of the spherical sampling grid for $Ff$ and involves the computation of a Voronoi diagram on the unit sphere. We demonstrate the excellent speed and accuracy of our method by using simulated data. © 2004 Optical Society of America

OCIS codes: 100.3010, 100.3190, 100.6890, 100.6950.

1. INTRODUCTION

The problem of inversion of the three-dimensional (3-D) ray transform arises in cryo-electron microscopy (cryo-EM) and is a central element in a set of associated image processing techniques, collectively known as single particle analysis. Cryo-EM is a tool for the structural analysis of large macromolecular complexes with sizes ranging from 10 to 150 nm (individual proteins, ribosomes, viruses). The cryo-electron microscope is used to obtain parallel beam projections of frozen hydrated macromolecules suspended in random orientations. Within the linear, weak-phase approximation of the transfer function of the microscope, the projections represent line integrals of the mass density of the particle under examination. The directions of the projections are random (in most cases nonuniform) and initially unknown. The number of projections is extremely large, and each projection is extremely noisy. In the reconstruction step, one seeks to reconstruct the mass density of the particle from its measured projections. If the directions of the projections were known, one could achieve this goal by inverting the 3-D ray transform. To cope with the unknown directions, one starts with initial guesses for either the projection directions or the density function of the particle. Next, the density function is refined in an iterative procedure in which 3-D reconstruction steps are alternated with estimation and reevaluation of the projection directions. At each step of the procedure, one has to solve a reconstruction problem of the type considered here. In principle, one could use any reconstruction algorithm for the 3-D ray transform that is capable of handling random projection directions. Nevertheless, to make the procedure efficient and to minimize the propagation of errors, the reconstruction algorithm should be fast and accurate.

More formally, let $f$ denote the density function to be reconstructed, and let $P$ denote the 3-D ray transform. The function $g = Pf$ may be seen as a collection of parallel beam projections of $f$, indexed by the projection direction. As indicated above, the electron microscope provides a sampled and noisy version of $g$ with unknown projection directions. During each iteration of the above-mentioned refinement procedure, one has to reconstruct $f$ from this sampled version of $Pf$, using the current guesses of the direction projections. The reconstruction can be achieved by using one of several general strategies. One general strategy is to discretize the equation $Pf = g$ and to solve the resulting linear system of equations by a suitable it-
erative method, such as the simultaneous iterative reconstruction technique (SIRT)\textsuperscript{4} or the algebraic reconstruction technique.\textsuperscript{5} Methods of this class can be very accurate but are rather slow.

Another general strategy is to evaluate a discrete version of the expression $Qg$, where $Q$ is the (or an approximate) inverse of $P$. The widely used weighted (or filtered) backprojection methods\textsuperscript{6} belong to this class of reconstruction algorithms. The random and nonuniform distribution of the projection directions is taken into account by a weighting (filter) function built into these methods. Weighted backprojection methods are still relatively slow, and as a result of difficulties with constructing a proper weighting function, they do not perform well for nonuniformly distributed projections.

So-called direct Fourier methods are also based on the second general strategy. These methods exploit the projection theorem for the 3-D ray transform, which states that the two-dimensional (2-D) Fourier transform of a projection of $f$ equals $Ff$, the 3-D Fourier transform of $f$, on a central plane perpendicular to the projection direction. Thus, by numerically 2-D Fourier transforming each projection, one obtains a discrete approximation to $Ff$ on some 3-D grid, and a subsequent numerical 3-D inverse Fourier transform gives a discrete approximation to $f$. Typically, the projections of $f$ are available on a 2-D Cartesian grid, and the 2-D fast Fourier transform (FFT) may be used for the Fourier transformation of the projections, making the preprocessing step very fast. If the 3-D inverse Fourier transform could be realized by means of the 3-D inverse FFT, one would indeed have a very fast reconstruction algorithm. Unfortunately, the preprocessing step yields $Ff$ on a nonuniform grid, and the 3-D inverse FFT is not immediately applicable. There are several possible ways to overcome this obstacle.

One possibility is to resample the nonuniformly sampled version of $Ff$ onto a 3-D Cartesian grid by some form of interpolation and to apply the 3-D inverse FFT to this resampled version of $Ff$. For example, Grigorieff used a modified trilinear interpolation scheme.\textsuperscript{7} Simple interpolation methods have been found to give inaccurate results, although more sophisticated interpolation schemes can go a long way to improve the accuracy.\textsuperscript{8,9} Unfortunately, the recommended window size makes them impractical for most applications.\textsuperscript{10}

Another method for reconstructing $f$ from a nonuniformly sampled version of $Ff$ is the 3-D gridding method\textsuperscript{11–13} With this method, one first computes an approximation to the convolution $(Fw)*((Ff))$ on an appropriate Cartesian grid, where $Fw$ is a judiciously chosen convolution kernel; then one uses the 3-D inverse FFT to compute an approximation to $wf = F^{-1}((Fw)*((Ff)))$ on a Cartesian grid; and, finally, one divides this approximation to $wf$ by $w$. For the convolution step, also known as “gridding,” the data need to be multiplied with proper “gridding weights” designed to compensate for the nonuniform distribution of the grid points. Good convolution kernels have a small support so that the gridding method is still fast, provided that the computation of the gridding weights is also fast. The accuracy of the method is strongly influenced by the distribution of the sampling points and by the choice of the gridding weights. A good weighting scheme is to partition the sampled region in Fourier space into small “sampling cells,” with one cell around each grid point, and to choose the weights as the volumes of these sampling cells.\textsuperscript{13} The Voronoi diagram of the grid points provides a good partitioning of the sampled region. In principle, the computation of Voronoi diagrams is feasible, but it can be time-consuming, particularly in 3-D applications.

The gridding method has other applications in, for example, resampling and interpolation. Reversing the steps of the gridding method yields a method for calculating a nonuniformly sampled version of the Fourier transform of some function $h$ from a uniformly sampled version of $h$.\textsuperscript{14} We call this the reverse gridding method.

In the case of cryo-EM, the calculation of the gridding weights by means of the 3-D Voronoi diagram of the nonuniformly spaced grid points for $Ff$ would lead to an accurate, but slow, direct Fourier method. The reconstruction algorithm that we present in the main part of this paper is a variation of this slow direct Fourier method that improves the speed and maintains the accuracy. Specifically, we use the 2-D reverse gridding method, rather than the 2-D FFT, to compute the Fourier transform of each projection on a 2-D polar grid rather than a 2-D Cartesian grid. In this way, we obtain $Ff$ on a 3-D spherical grid, where the grid points are located both on centered spheres and on straight lines through the origin. This makes it possible to partition the sampled region into suitable sampling cells through the computation of a 2-D Voronoi diagram on the unit sphere, rather than of a 3-D Voronoi diagram in Euclidean space. A fast algorithm for computing Voronoi diagrams on the unit sphere is available.\textsuperscript{15}

The remainder of this paper is organized as follows. Section 2 contains preparatory material. In particular, we provide an overview of gridding techniques and a brief account of spherical Voronoi diagrams. In Section 3, we state the reconstruction problem and describe our algorithm for solving it. Test results are provided and discussed in Section 4. Section 5 concludes the paper with remarks concerning possible improvements of the method and its possible impact on single particle analysis.

2. PRELIMINARIES

A. Notation

The symbols $\mathbf{R}$, $\mathbf{R}^d$, $\mathbf{Z}$, $\mathbf{Z}_+$, and $\mathbf{C}$ denote the sets of real, positive real, integer, positive integer, and complex numbers, respectively. Points in $\mathbf{R}^d$ are denoted by boldface letters, such as $\mathbf{a}$, $\theta$, or $\mathbf{N}$. The components of $\mathbf{a} \in \mathbf{R}^d$ are denoted by $a_1, \ldots, a_d$. Expressions such as $\mathbf{a} \geq \mathbf{b}$, $\mathbf{a}/\mathbf{b}$, $\mathbf{a} + 1$, or $\mathbf{a} = 1$ are understood componentwise. The orthogonal complement of $\mathbf{a} \in \mathbf{R}^d$ is the set $\mathbf{a}^\perp = \{x \in \mathbf{R}^d : \mathbf{a} \cdot x = 0\}$; if $\mathbf{a} \neq 0$, then $\mathbf{a}^\perp$ is the central hyperplane perpendicular to $\mathbf{a}$. The Euclidean norm of $\mathbf{a}$ is denoted by $|\mathbf{a}|$. The $d$-dimensional ($d$-D) volume of a region $A \subset \mathbf{R}^d$ is denoted by $\text{vol}(A)$. The closed centered ball in $\mathbf{R}^d$ with radius $r$ is denoted by $B_d(r)$. The surface of the ball $B_d(1)$ is the unit sphere in $\mathbf{R}^d$ and conventionally denoted by $S^{d-1}$. The terms Cartesian grid and uniform grid are used interchangeably.
B. Fourier Transform

The $d$-D Fourier transform of a sufficiently regular function $h: \mathbb{R}^d \to \mathbb{C}$ is defined by

$$ (Fh)(y) = \int_{\mathbb{R}^d} h(x) \exp(-i 2\pi x \cdot y) \, dx. \quad (1) $$

We also write $\hat{h}$ instead of $Fh$. The inverse Fourier transform is given by

$$ h(x) = (F^{-1}\hat{h})(x) = \int_{\mathbb{R}^d} \hat{h}(y) \exp(i 2\pi x \cdot y) \, dy. \quad (2) $$

C. Ray Transform

The $d$-D ray transform of a sufficiently regular function $h: \mathbb{R}^d \to \mathbb{C}$ is defined by

$$ (P h)(\theta, u) = \int_{\mathbb{R}} h(u + t \theta) \, dt, \quad \theta \in S^{d-1}, \; u \in \Theta^d. \quad (3) $$

The function

$$ P \hat{h} = (P h)(\theta, \cdot) \quad (4) $$

is called a projection of $h$ or, more precisely, the $(d-1)$-D parallel beam projection of $h$ with direction $\theta$. The ray transform is related to the Fourier transform by the projection theorem, which states that

$$ (P \hat{h})(\xi) = \hat{h}(\xi), \quad \xi \in \Theta^d, \quad (5) $$

i.e., each Fourier-transformed projection of $h$ yields the Fourier transform of $h$ on the central hyperplane whose normal is parallel to the projection direction.

D. Gridding Method

The following problem is often encountered in scientific and technical applications. We are given a sampled version \{$(h_1, y_1), \ldots, (h_J, y_J)$\} of the Fourier transform $\hat{h}$ of some function $h: \mathbb{R}^d \to \mathbb{C}$, where $J$ is some finite index set and the grid points $y_j \in \mathbb{R}^d$ form a possibly nonunif orm sampling grid in some region $S \subset \mathbb{R}^d$. We wish to reconstruct $h$ on a Cartesian grid of the form \{$(k, z)^T \in \mathbb{Z}^d \times (-K \leq k < K/2)$\}, where $a \in \mathbb{R}^d$ is the grid spacing and $K \in \mathbb{Z}^d$. The gridding method, outlined below, is a potentially efficient and accurate method for solving this kind of problem.

The method involves a number of parameters. First, we need a set of gridding weights \{$(c_j \in \mathbb{R}^d : j \in J)$\). A general recipe for the construction of good gridding weights is given below. Second, we need an even and FFT-friendly $N \in \mathbb{Z}^d$ with $N \geq K$. The ratio $N/K$ is called the oversampling factor. The oversampling factor should be greater than 1 but need not be greater than 2. The parameters $N$ and $a$ define an auxiliary Cartesian grid \{$(nb, n) \in \mathbb{Z}^d, -N/2 \leq n < N/2$\} with grid spacing $b = 1/(aN)$. \quad (6)

Note that this auxiliary grid fills the interval $[-bN/2, bN/2] = [-1/(2a), 1/(2a)]$, which is independent of $N$. Third, we need a window function $\hat{w}: \mathbb{R}^d \to \mathbb{R}$ with support in an interval of the form $[-Wb/2, Wb/2]$, where $W \in \mathbb{Z}_+$ is “small,” say $W = 4$ or $W = 6$. The weighting function $\hat{w} = F^{-1}\hat{w}$ must be positive in $[-Ka/2, Ka/2]$. A recommendable window function is the $d$-D separable Kaiser–Bessel window. Using the abbreviations \(r = Ka/2\) and $s = Wb/2$, we may write this bell-shaped window function as

$$ \hat{w}_{\text{KB}}(y) = \begin{cases} \frac{1}{2} \sum_{l=1}^{d} I_0(2 \pi \alpha_r s_l) \left( 1 - \frac{(y_l)^2}{s_l^2} \right)^{1/2} & 0, \\ y \in [-s, s], \\ y \in [-s, s], \end{cases} \quad (7) $$

where $I_0$ is the zero-order modified Bessel function and $\alpha \in \mathbb{R}^d$ is a parameter. In the common case $W = 6$, a good choice is $\alpha = 1.25$. The weighting function associated with $\hat{w}_{\text{KB}}$ is

$$ w_{\text{KB}}(x) = \frac{1}{2} \sum_{l=1}^{d} \sinh(2 \pi \alpha_r s_l) \left( 1 - \frac{(x_l)^2}{s_l^2} \right)^{1/2}. \quad (8) $$

The gridding method itself proceeds in three steps. In the first step, also known as gridding, one computes the numbers

$$ \hat{g}_n = \sum_{j \in J} c_j \hat{h}(y_j) \hat{w}(nb - y_j), \quad -N/2 \leq n < N/2. \quad (9) $$

In the second step, one uses the $d$-D inverse FFT to compute the numbers

$$ g_k = b_1 \cdots b_d \sum_{n=-N/2}^{N/2} \hat{g}_n \exp[i 2\pi k(n/N)], \quad -K/2 \leq k < K/2. \quad (10) $$

To do this, one cyclically shifts the input sequence \{$b_1 \cdots b_d \hat{g}_n$: $-N/2 \leq n < N/2$\} by $N/2$, applies the $d$-D inverse FFT, cyclically shifts the output sequence \{$g_k$: $-N/2 \leq k < N/2$\} by $N/2$, and extracts the interesting portion \{$g_k$: $-K/2 \leq k < K/2$\}. In the third step of the gridding method, one simply computes the numbers

$$ h_k = g_k / w(ka), \quad -K/2 \leq k < K/2. \quad (11) $$

The sequence \{$h_k$: $-K/2 \leq k < K/2$\} constitutes the output of the gridding method. As shown in Ref. 18, regardless of the choice of the gridding weights, if the remaining parameters are properly chosen (for example, as indicated above), then the numbers $h_k$ provide highly accurate approximations to the numbers $(s \cdot h)(ka)$, $-K/2 \leq k < K/2$, where
\[ s(x) = \sum_{j \in J} c_j \exp(i2\pi x \cdot y_j), \]  \hspace{1cm} (12)

\[ (s*h)(x) = \sum_{j \in J} c_j h(y_j) \exp(i2\pi x \cdot y_j). \]  \hspace{1cm} (13)

Thus the similarity between \( s*h \), the function delivered by the gridding method, and \( h \), the function to be reconstructed, depends on the “point-spread function” \( s \), which depends, in turn, on the grid points \( y_j \) and the gridding weights \( c_j \). The grid points are often imposed or constrained by the underlying application, but the choice of the gridding weights is under our control. A general mathematical theory that allows one to derive the shape \( g_j \) and choose \( c_j \) seems to be missing.

In Ref. 18, it was proposed to choose the gridding weights such that the right-hand side of Eq. (13) becomes a Riemann sum approximating the integral

\[ h_S(x) = \int_S h(y) \exp(i2\pi x \cdot y) dy, \]  \hspace{1cm} (14)

which is, in turn, an approximation to \( h(x) \). To find such weights, we need to partition the sampled region \( S \) into sampling cells \( C_j \) such that \( y_j \in C_j \) if and only if \( k = j \), and choose \( c_j \) as the volume of cell \( C_j \), i.e.,

\[ c_j = \text{vol}(C_j). \]  \hspace{1cm} (15)

The Voronoi diagram of the sampling points provides a good partitioning of \( S \). Depending on the sampling grid, faster methods for partitioning \( S \) may exist.

In the recent mathematical literature, several numerical methods for the efficient evaluation of trigonometric sums of the form \( \hat{g}_k = \sum_{j \in J} \hat{g}_j \exp[-i2\pi k \cdot (n/N)] \) have been investigated. These methods are closely related to the gridding method, the numbers \( \hat{g}_j \) in the above trigonometric sum corresponding to the numbers \( c_j \hat{h}(y_j) \) in Eq. (11). The choice of the gridding weights has not been discussed in these references.

There is an obvious “dual” version of the gridding method that allows one to compute the Fourier transform \( \mathcal{F}h \) of some function \( h \) on a uniform grid from a nonuniformly sampled version of \( h \). If we apply a d-D inverse FFT to the output of the dual gridding method, we obtain a reasonably fast and potentially highly accurate method for the uniform resampling of a nonuniformly sampled function \( h \).

E. Reverse Gridding Method

The reverse gridding method may be used to solve the reverse problem, which aims at the computation of the possibly nonuniformly spaced samples \( \hat{h}(y_j), j \in J \), from the uniformly spaced samples \( h(ka), -K/2 \leq k < K/2 \). The method involves the same set of parameters as that for the gridding method, except that the gridding weights \( c_j \) are not needed here. Roughly speaking, the reverse gridding method is obtained from the gridding method by carrying out the three steps of the latter in reverse order.

Thus, in the first step of the reverse gridding method, one computes the numbers

\[ \hat{g}_k = h(ka)/w(ka), \]  \hspace{1cm} (16)

In the second step, the sequence \( \{\hat{g}_k: -K/2 \leq k < K/2\} \) is symmetrically zero padded to yield the sequence \( \{\hat{g}_k: -N/2 \leq k < N/2\} \), and the d-D FFT is used to compute the numbers

\[ \hat{g}_n = a_1 \cdots a_d \sum_{k=-N/2}^{N/2} \hat{g}_k \exp[-i2\pi k \cdot (n/N)], \]  \hspace{1cm} (17)

In the third step, one computes the numbers

\[ h_j = b_1 \cdots b_d \sum_{n=-N/2}^{N/2} \hat{g}_n \hat{w}(y_j - nb), \]  \hspace{1cm} (18)

The constant weight \( b_1 \cdots b_d \) is thus assigned to the output of the reverse gridding method. An argument similar to the one presented in Ref. 18 for the gridding method leads to the conclusion that, if the parameters are properly chosen (for example, as suggested for the gridding method), the numbers \( \hat{h}_j \) provide highly accurate approximations to the numbers \( \hat{s}_{ak} \), \( \hat{h}(y_j) \), \( j \in J \), where

\[ \hat{s}_{ak}(y) = a_1 \cdots a_d \sum_{k=-K/2}^{K/2-1} \exp[-i2\pi (ka)y]. \]  \hspace{1cm} (19)

This point-spread function is closely related to the well-known Dirichlet kernel. If \( a \) is sufficiently small and \( K \) sufficiently large, \( \hat{s}_{ak} \) will resemble \( \hat{h} \).

Close relatives of the reverse gridding method have also been studied in Refs. 20–22.

There is an obvious dual version of the reverse gridding method that allows one to compute a function \( h \) on a nonuniform grid from a uniformly sampled version of \( \mathcal{F}h \). We obtain a reasonably fast and potentially highly accurate method for the nonuniform resampling of a uniformly sampled function \( h \) if we apply the d-D FFT to this uniformly sampled version of \( h \) before inverting the reverse gridding method. We obtain a reasonably fast and potentially highly accurate method for the nonuniform resampling of a nonuniformly sampled function \( h \) if we replace the FFT by the dual gridding method.

F. Spherical Voronoi Diagrams

A spherical Voronoi diagram is a Voronoi diagram defined on the surface of a ball. In this subsection, we provide the required minimum information about spherical Voronoi diagrams on \( S^2 \), the case of interest. For detailed information, see Ref. 15.

The distance between two points \( p \) and \( q \) on \( S^2 \) is defined by \( d(p, q) = \arccos(p \cdot q) \). Let \( P = \{p_1, \ldots, p_n\} \subset S^2 \) be a finite set of points, or nodes, which do not lie on a common great circle. The (spherical) Voronoi region \( V(P, i) \) of node \( p_i \) is the closure of the set of points on \( S^2 \) that are closer to \( p_i \) than to any other node, i.e.,

\[ V(P, i) = \{p \in S^2: d(p, p_i) \leq d(p, p_j), \] \hspace{1cm} (20)

The boundary of a Voronoi region consists of a finite number of vertices (points on \( S^2 \)) that are connected by segments of great circles. The set of Voronoi regions...
V(P, 1),...,V(P, n) is called the (spherical) Voronoi diagram associated with P. The Voronoi regions of a spherical Voronoi diagram provide a partitioning of $S^2$. The Voronoi diagram of n nodes on $S^2$ can be computed with $O(n \log n)$ operations. A fast algorithm that achieves this complexity is given in Ref. 15. This algorithm is incremental; i.e., if the Voronoi diagram of n nodes is already available and another node added, then the Voronoi diagram of these n + 1 nodes is computed efficiently from the already available Voronoi diagram. The storage requirement is $O(n)$.

Nevertheless, when the number of nodes is extremely large (a typical situation in single particle analysis), the required amount of storage can be large and computation time long. The storage requirements, i.e., the maximum amount of data that has to be kept in memory, may be reduced as follows. First, we partition the sphere into a large number of geometrically simple regions $S_1,...,S_q$, initially such that each region contains approximately $n/I$ nodes with $I \approx \sqrt{n}$. Next, we enlarge these regions by a "safety margin" so that the enlarged regions $S_1',...,S_q'$ overlap slightly. The number of nodes in each of the regions $S_i'$ will be larger than the number of nodes in the original region $S_i$ but still be much smaller than n. The reduced number of nodes in these regions reduces the storage requirements accordingly, i.e., to $O(\sqrt{n})$. Then we calculate the spherical Voronoi diagrams associated with the nodes in each of the enlarged regions $S_i'$. Note that this can be done independently for each $S_i'$. The areas of the Voronoi regions within the regions $S_i'$ are computed (see below), and their sums are divided by the analytically calculated surface areas of the $S_i'$. This yields the relative surface area errors for all regions $S_i'$. If any of these errors exceeds a preset error level, the respective safety margins are enlarged accordingly, and the calculations are repeated. Finally, if the regions $S_i$ and $S_i'$ have been properly chosen, then the Voronoi regions associated with the nodes in the regions $S_i$ will combine into the Voronoi diagram of the complete set of the original nodes. Incidentally, this "divide and conquer" approach also permits the reduction of the computation time by means of parallel computing: If $P$ is the number of processors, then each processor would compute the Voronoi diagrams in $IP$ of the regions $S_i'$, with all processors operating in parallel. Admittedly, our divide and conquer approach is heuristic. Its success relies on the proper choice of the regions $S_i$ and $S_i'$, whose determination may need some experimentation.

For our application, we also need to compute the areas of Voronoi regions. There is an efficient way of doing this: Let $q_1,...,q_m$ be the vertices of some Voronoi region $V$ whose area $a$ we wish to compute. Then

$$a = a_1 + \cdots + a_m - (m-2)\pi,$$

where $a_m$ is the interior angle of the region at vertex $q_m$. It is defined as the angle between the tangents to the two great circles that meet at $q_m$ and belong to the boundary of $V$. This angle is computable from vertex $q_m$ and the two neighboring vertices that are connected to $q_m$ by the two great circles. Let these neighboring vertices be $q_{a}$ and $q_{b}$, respectively, with $q_{a}, q_{b}, q_{c}$, in counterclockwise order as viewed from the outside of the sphere. Then

$$a_m = \arccos\left(\frac{(q_{a} \times q_{b}) \cdot (q_{b} \times q_{c})}{|q_{a} \times q_{b}| \cdot |q_{b} \times q_{c}|}\right),$$

where $\times$ denotes the usual cross product in $R^3$.

### 3. GRIDDING-BASED FOURIER INVERSION OF THE THREE-DIMENSIONAL RAY TRANSFORM

#### A. Reconstruction Problem

We are given a set of uniformly sampled functions $g_n: R^2 \rightarrow R$, $n = 1,...,N$. We know that these functions are parallel beam projections of some function $f: R^3 \rightarrow R$, i.e., we have $g_n = P_{\theta_n}f$, $n = 1,...,N$. The directions $\theta_n \in S^2$ are regarded as known. We wish to reconstruct $f$ on the Cartesian grid $\{ka: k \in Z^3, -K/2 < k < K/2\}$, where $a \in R^3$ is the grid spacing and $K \in Z^3$. In practice, the spacing and the extent of this grid will be suggested by the underlying application. We assume that $a = (a, a, a)$ for some $a \in R$, and $K = (K, K, K)$ for some $K \in Z_*$. The values chosen for the parameters $a$ and $K$ reflect the maximum anticipated resolution and size of the reconstructed object.

We shall loosely assume that the projection directions $\theta_n$ are such that the hyperplanes $\theta_n$ are of the form $|\theta| = \theta_n$. That is, if $P \in B_\alpha(R)$ is a point on one of these hyperplanes, then there is another one of these hyperplanes and a point $q \in B_\beta(R)$ on this other hyperplane such that $|P - q| \leq \alpha/\theta_n(R)$ for some $\beta < 1$. A condition of this type is required for an accurate and stable reconstruction of $f$. Our condition may also be viewed as a discrete version of the condition required for Orlov's inversion formula for the 3-D ray transform. In single particle analysis, our condition is often, but not always, satisfied.

#### B. Reconstruction Algorithm

The reconstruction algorithm comprises a preprocessing stage and a main stage (Fig. 1). In the preprocessing stage, the projections are numerically Fourier transformed, yielding a discrete approximation to $\hat{f}$. In the main stage, this discrete approximation to $\hat{f}$ is numerically inversely Fourier transformed, yielding a discrete approximation to $f$.

More precisely, in the preprocessing stage, we apply the 2-D reverse gridding method to each projection $g_n$ to compute $\hat{g}_n$ on a polar grid on the hyperplane $\theta_n$. We choose the radial grid spacing $d_r$ in the form

$$d_r = R/L,$$

with $R$ as in Eq. (23) and, typically, with $L = K$ or a similar integer. This choice makes the radial grid spacing of the order of $1/(2aK)$. In any case, we choose the same radial grid spacing for each projection. The angular grid spacing should be of the order of $2\pi R/L$. Then, in concert with the projection theorem, the preprocessing stage...
is numerically inversely Fourier transformed, yielding \( f_{\hat{g}} \) in. In the main stage, this nonuniformly sampled version of the object function \( f \) is numerically Fourier transformed and evaluated on a 2-D polar grid. The numerical inverse Fourier transformation is realized by the 2-D reverse gridding method. By the projection theorem, the output of the preprocessing stage gives a sampled version of the object function \( f \). Moreover, the points \( \mathbf{y}_{lm} \) are all equal to \( 0 \). In addition, if the distribution of the directions \( \theta_n \) is random, the distribution of the points \( \mathbf{p}_m \) on \( S^2 \) is effectively random, too.

In the subsequent main stage, we apply the 3-D gridding method to the sampled approximation to \( \hat{f} \) generated by the preprocessing stage. Following the recipe outlined in Subsection 2.D, we partition \( S \) into an as yet unspecified sampling cell \( C_0 \) around \( 0 \) and a set of as yet unspecified sampling cells \( C_{lm} \) around the points \( \mathbf{y}_{lm} \), \( l = 1,...,L \), \( m = 1,...,M \), and approximate the integral

\[
\langle \hat{f}(\mathbf{y}) \rangle = \int_S \hat{f}(\mathbf{y}) \exp(i2\pi\mathbf{x} \cdot \mathbf{y}) d\mathbf{y}
\]

by the Riemann sum

\[
f_S(\mathbf{x}) = c_0 \hat{f}(0) + \sum_{l=1}^L \sum_{m=1}^M c_{lm} \hat{f}(\mathbf{y}_{lm}) \exp(i2\pi\mathbf{x} \cdot \mathbf{y}_{lm})
\]

with the gridding weights

\[
c_0 = \text{vol}(C_0), \quad c_{lm} = \text{vol}(C_{lm}),
\]

\( l = 1,...,L \), \( m = 1,...,M \). (31)

We now specify the sampling cells. For the sampling cell around \( 0 \), we choose the ball \( B_2(d_r/2) \), so that

\[
c_0 = \frac{\pi}{6} d_r^3.
\]

(32)

For the sampling cells around the points \( \mathbf{y}_{lm} \), we choose the sets

\[
C_{lm} = \{ \mathbf{p} \in V(\mathbf{P}, m) \},
\]

where \( V(\mathbf{P}, m) \subset S^2 \) is the Voronoi region around \( \mathbf{p}_m \) of the spherical Voronoi diagram associated with \( \mathbf{P} = \{ \mathbf{p}_1,...,\mathbf{p}_M \} \). In principle, we can compute the Voronoi regions \( V(\mathbf{P}, m) \) and their areas \( a_{lm} \) as described in Subsection 2.F. Since in single particle analysis the number of angles \( M \) tends to be extremely large (of the order of 10^5), we may want to employ the divide and conquer approach indicated at the end of that subsection, dividing \( S^2 \) into "spherical zones" \( S_i = \{ \mathbf{p} \in S^2 : k_i \leq p_3 < h_{i+1} \}, i = 1,...,I \), with suitably chosen numbers \( h_i \), satisfying \(-1 = h_1 < \cdots < h_{I+1} = 1 \). The volume of sampling cell \( C_{lm} \) is finally obtained as

\[
c_{lm} = a_m \int_{r_{i-1}}^{r_i} r^2 dr = \frac{a_m}{3} [3r_i^2 + (d_r/2)^2].
\]

(34)
The rest of the gridding method proceeds as described in Subsection 2.D. Good choices for the parameters were already indicated there. Note that the ball $B_2(R) = B_2(1/(2a))$, the smallest centered ball that contains the spherical grid \( \{ y_{lm} : l = 0, \ldots, L, m = 1, \ldots, M \} \), is the biggest centered ball in the cube \([-1/(2a), 1/(2a)]\) = \([-bN/2, bN/2]\), the region that contains the auxiliary Cartesian grid \( \{ k \cdot a : k \in \mathbb{Z}^3, -K/2 < k < K/2 \} \). If the oversampling factor \( N/K \) is near 2, as suggested, the grid spacing \( b = 1/(aN) \) of the auxiliary Cartesian grid will be of the order of \( 1/(2aK) \), which is also the suggested order of the radial grid spacing \( d_r \) of the spherical sampling grid for \( \hat{f} \). As pointed out in Subsection 2.D, the accuracy of the main stage depends on the distribution of the grid points \( y_{lm} \) and the choice of the gridding weights \( c_{lm} \). Of course, the accuracy of the final result is also influenced by the accuracy of the preprocessing stage and the quality of the input data. However, a detailed discussion of these issues is beyond the scope of this paper.

The implementation of the above algorithm admits of an important improvement: Since \( f \) and its projections are real, the Fourier transforms of \( f \) and its projections are Hermitian. Moreover, the sampling pattern formed by the points \( y_{lm} \) is symmetric with respect to the origin. These facts may be exploited to save nearly 50% of computation time and storage space during all stages of the reconstruction procedure, including the computation of the gridding weights.

4. TEST RESULTS

We implemented the reconstruction algorithm presented in Section 3 within the framework of the SPIDER image processing system, using the FORTRAN programming language. We calculated projections of a simulated test object to find the speed and the accuracy of our algorithm and to compare it with alternative reconstruction algorithms.

The computing platform was an SGI Origin 3200 equipped with 8 Gbytes of main memory and eight R12000 processors, each running at 400 MHz.

The shape of a nonsymmetric test object (mathematically speaking, the support of the function representing this object) was made equal to the previously determined shape of the *Escherichia coli* 70S ribosome. The internal-density values were set to either 1, 2, or 3, and the background was set to 0. The object was evaluated on the uniform grid \( \{ k \cdot a : k \in \mathbb{Z}^3, -K/2 < k < K/2 \} \) with \( a = (a, a, a) \), where \( a = 1 \), and \( K = (K, K, K) \) with \( K = 75 \). (The choice \( a = 1 \) means that we are using the voxel size as the unit of length.) In Fig. 2(a), we show the central slice \( x_3 = 0 \) of the test object.

We computed a set of \( N = 3237 \) noise-free, parallel beam projections of the test object. The number of the projections and their directions were taken from one of the data sets that had led to the previous determination of the structure of the *E. coli* 70S ribosome. The projections were computed as follows: First, the 3-D reverse gridding method (with \( N = (150, 150, 150) \)) was applied to the test object to obtain the Fourier transform of the test object on a nonuniform grid composed of Cartesian grids in the central hyperplanes perpendicular to the projection directions. Then the data on each of these Cartesian grids were inversely Fourier transformed with the 2-D inverse FFT. By the projection theorem, the outputs of the inverse FFTs provide the desired projections. A second set of projections was generated by numerically computing the line integrals occurring in the definition of the ray transform, using trilinear interpolation to evaluate the object function between the sampling points. Both sets of projections are free from measurement noise. A third set of noise-corrupted projections was obtained by adding Gaussian noise with a signal-to-noise ratio of approximately 25 to the first set of noise-free projections.

We applied four reconstruction algorithms to each of these data sets. In all cases, the reconstruction grid was characterized by \( a = (1, 1, 1) \) and \( K = (75, 75, 75) \). (Again, the voxel size serves as the unit of length.) The first algorithm was our gridding-based direct Fourier reconstruction algorithm (GDFR) presented in Section 3, where we used the gridding weights (32) and (34) and \( N = (150, 150, 150) \). (The SPIDER system uses a mixed-radix FFT; therefore it can efficiently handle nonpower of two data lengths.) Note that the grid points \( y_{lm} \) fall within the ball \( B_2(1/2) \) and that the auxiliary Cartesian grid fills the cube \([-1/2, 1/2]^3\). The required spherical Voronoi diagram was computed with the divide and conquer approach outlined in Subsection 2.F. The remain-
ing parameters of the method were chosen as suggested in Subsections 2.D, 2.E, and 3.B. More specifically, we used the d-D separable Kaiser–Bessel function as a window function with support $W = 6$ and with $a = 1.25$ [see Eq. (7)], the oversampling factor $N/K$ in both reverse gridding and gridding steps was 2, and in the 2-D reverse gridding the radial grid spacing $d_r$ was set to 1/150. The Hermitian symmetry of the data was exploited. The second algorithm was the general weighting Fourier reconstruction method (WBP1) described in Ref. 27. The third algorithm was the exact filter Fourier reconstruction method (WBP2) described in Ref. 28. These two algorithms belong to the class of weighted backprojection methods. The fourth reconstruction algorithm was the simultaneous iterative reconstruction technique (SIRT) described in Ref. 4, with the number of iterations set to 200. All algorithms were implemented by using OpenMP directives to take advantage of shared-memory-type parallel processing and are publicly available as parts of SPIDER, the dedicated package for EM data processing.

For the assessment of the accuracy of these algorithms, one needs a measure of the similarity of two objects. As our primary similarity measure, we chose the Fourier shell correlation (FSC),

$$\text{FSC}(f, g; r, \epsilon) = \sum_{|\|y_n\| - r| \leq \epsilon} |\hat{f}(y_n)\hat{g}^*(y_n)|^2 \left/ \left( \sum_{|\|y_n\| - r| \leq \epsilon} |\hat{f}(y_n)|^2 \sum_{|\|y_n\| - r| \leq \epsilon} |\hat{g}(y_n)|^2 \right) \right)^{1/2},$$

where the asterisk indicates complex conjugation, $2\epsilon$ is a preselected shell thickness, and the $y_n$ form a uniform grid in Fourier space. Thus the FSC yields a one-dimensional curve of correlation coefficients as a function of the magnitude of the spatial frequency. Note that the FSC is insensitive to a scaling of the objects. An FSC curve everywhere close to 1 signals a strong similarity between $f$ and $g$; an FSC curve with values close to 0 signals a lack of similarity between $f$ and $g$. FSC curves may be used to measure both the “fidelity” and the “consistency” of reconstruction algorithms. The fidelity of an algorithm is reflected by the curve $\text{FSC}(f, g; \cdot, \cdot)$, where $f$ is the ideal object and $g$ the reconstructed object. The consistency is reflected by the curve $\text{FSC}(f_1, f_2; \cdot, \cdot)$, where $f_1$ and $f_2$ are obtained by randomly splitting the set of available projections into halves and reconstructing the associated objects.

In our tests, we evaluated the fidelity and consistency curves at the discrete radii $r_l$ defined in Eq. (26) with $\epsilon$ equal to $d_r/2$. The Fourier transforms in Eq. (35) were computed by applying a 3-D FFT of length (75,75,75) to the discrete versions of $f$ and $g$. Our secondary similarity measure was the correlation coefficient between the test object and the reconstructed object whose similarity is to be measured, where we restricted the comparison to the interior of a centered sphere with a radius of $[K/2a]$. As a variant, we also computed the correlation coefficient between the test object and a low-pass filtered version of the reconstructed object, which was obtained by setting to 0 the spatial frequencies of the reconstructed object outside the ball $B_3(1/(2a))$.

The first set of noise-free projections favors algorithm GDFR, because it also works in Fourier space. FSC curves obtained with this set of data are shown in Fig. 3. The fidelity curve of GDFR is virtually identical to 1.
The consistency curve is equally impressive. In Fig. 2(b), we show the central slice $k_3 = 0$ of the object reconstructed by GDFR from the first set of data. In comparison with the central slice of the object reconstructed by the SIRT [Fig. 2(d)], it contains small “ringing” artifacts. This is due to the inability of the GDFR algorithm to recover information in the “corners” of Fourier space (see the discussion below). The resulting truncation of the Fourier series results in Gibbs oscillations in the reconstructed object. In Fig. 2(c), we show the difference between the object reconstructed by GDFR and the test object after setting to zero the spatial frequencies outside the centered ball with radius $1/(2a)$, where $a$ is the voxel size in object space, i.e., outside the region that could not be recovered by the GDFR algorithm. The maximum absolute value of the difference is 0.04 (recall that the test object has values set between 0.0 and 3.0). The difference pattern is mainly composed of high-frequency noise with a superimposed lower-amplitude outline of the test object. The latter error is caused by the relatively higher error in recovery of very-low-frequency amplitudes by the GDFR algorithm. The next-best fidelity and consistency curves were obtained with the SIRT. [The dip of the SIRT fidelity curve at spatial frequency 0.42 in Fig. 3(a) is due to an inconsistency between the interpolation methods used internally in the SIRT to generate intermediate projections of the object to be reconstructed and the Fourier-space-based method used to generate the test data.] The curves produced by the two weighted back-projection methods WBP1 and WBP2 are markedly inferior, particularly in the high-spatial-frequency range. We attribute this to the smoothing effect of the various interpolation steps built into these methods. There is also a noticeable difference from the value 1 in the low-frequency range [Fig. 3(b)]. We believe that this difference is caused by the possibly nonoptimal weighting schemes used in these methods.

The second set of noise-free projections favors algorithm SIRT, because it also works in object space. In Fig. 2(d), we show the central slice $k_3 = 0$ of the object reconstructed by the SIRT. FSC curves obtained with this set of data are shown in Fig. 4. Both types of curves are markedly inferior to the corresponding curves obtained with the first set of data, particularly in the high-frequency region. (Note the different vertical scales of different figures.) This time, the fidelity curves obtained with GDFR and the SIRT are virtually identical. Nevertheless, the difference between the SIRT-reconstructed object and the test object [Fig. 2(e)] reveals relatively
strong reconstruction errors along the edges of the test object with the maximum absolute value of 0.36. This error is caused by the smoothing effect of the bilinear interpolations used internally in the SIRT and is much higher than the error of GDFR. The consistency curve obtained with GDFR favorably compares with the consistency curve of the SIRT. As with the first data set, both WBP1 and WBP2 yielded inferior curves in the whole frequency range.

FSC curves obtained with the noise-corrupted third data set are shown in Fig. 5. With noise-corrupted data, one expects the FSC curves to drop below 1 in the higher-spatial-frequency range as the signal level begins to approach the noise level. Given the amount of noise added to the data, the influence of the computation method for the underlying noise-free test data should be negligible.

The fidelity curves are shown in Fig. 5(a). Again, algorithm GDFR yields superior fidelity results. The consistency curves are shown in Fig. 5(b). The GDFR curve is now below the other consistency curves. This may be a consequence of the lack of smoothing built into GDFR, which will not dampen the noise either. We consider the lack of built-in smoothing as an advantage over the other algorithms.

The correlation coefficients between the test objects and the various reconstructed objects are listed in Tables 1 and 2. An inspection of the correlation coefficients confirms the observations made with the FSC fidelity curves. With one exception, the correlation coefficients obtained with GDFR are better than those obtained with the SIRT. The exception occurs with noise-free projections computed in object space (see the corresponding column of Table 1).

We think that this is because GDFR is unable to recover information that corresponds to the “corners” in Fourier space, i.e., the region \([-1/(2a), 1/(2a)]\). Indeed, if the comparison is made with the low-pass-filtered version of the reconstructed object, the difference between the correlation coefficients delivered by GDFR and the SIRT decreases (see the corresponding column of Table 2).

In summary, the above results suggest that GDFR is a competitive reconstruction algorithm in terms of accuracy, typically even surpassing the SIRT.

The reconstruction times were 2 min for GDFR, 6 min for WBP1, 6 min for WBP2, and 17 min for the SIRT. Thus GDFR is appreciably faster than the tested weighted backprojection algorithms and substantially faster than the SIRT.

5. FINAL REMARKS

Based on the test results reported in Section 4, we conclude that algorithm GDFR, the gridding-based direct Fourier reconstruction algorithm presented in this paper, is a superior method for the inversion of the 3-D ray transforms, in terms of both speed and accuracy.

Yet, there is room for improvement. For example, the efficiency and the accuracy of our heuristic divide and conquer approach for the calculation of the spherical Voronoi diagram depends sensitively on the choice of various parameters, which have not yet been fully optimized.

The current version of the program does not take advantage of possible symmetries in the data, which are caused by the inherent symmetry of many macromolecules of interest in structural biology. For example, some virus particles have icosahedral symmetry, and thus only 1/60 of the 3-D Fourier space contains independent information. It is easily verified that the corresponding distribution of nodes on the sphere will have a corresponding symmetry. Thus significant reduction of computation time and storage space can be achieved by taking advantage of the symmetry information and by constructing the Voronoi diagram for only a subregion on the sphere.

While our loose assumption on the distribution of the projection directions stated in Subsection 3.A is often satisfied, exceptions do occur. In such a case, the preprocessing stage will leave major gaps in the coverage of Fourier space. Such gaps will lead to very large gridding weights, which, in turn, will overemphasize the associated spatial frequencies. It may therefore be wise to limit the size of the gridding weights, i.e., the volumes of sampling cells to something like \(\beta/(aK)^3\) for some \(\beta\) a little larger than 1. Another possibility is to modify the weights such that the resulting signal-to-noise ratios of the 3-D Fourier coefficients in the reconstructed volume are maximized.\(^{31}\)

In single particle analysis, it is customary to oversample the input data, which are usually obtained by scanning a photographic film, in order to minimize any adverse effects of the subsequent image processing steps on the resolution of the reconstructed object. The accepted oversampling factor with respect to the target resolution is 3. In view of the excellent performance of

<table>
<thead>
<tr>
<th>Table 1. Correlation Coefficients between the Test Object and the Objects Reconstructed by the Four Algorithms Tested</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Reconstruction</td>
</tr>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>GDFR</td>
</tr>
<tr>
<td>WBP1</td>
</tr>
<tr>
<td>WBP2</td>
</tr>
<tr>
<td>SIRT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. Correlation Coefficients between the Test Object and the Low-Pass-Filtered Objects Reconstructed by the Four Algorithms Tested*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Reconstruction</td>
</tr>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>GDFR</td>
</tr>
<tr>
<td>WBP1</td>
</tr>
<tr>
<td>WBP2</td>
</tr>
<tr>
<td>SIRT</td>
</tr>
</tbody>
</table>

*The low-pass-filtered objects were obtained by setting to zero the spatial frequencies of the reconstructed objects outside the centered ball with radius \(1/(2a)\), where \(a\) is the voxel size in object space.
the gridding-based direct Fourier method, it should be sufficient to oversample the data by only a factor of \( \sqrt{2} \) in order to achieve the full target resolution.

Finally, the availability of the highly accurate interpolation and reconstruction algorithms made possible by gridding techniques may lead to the development of more accurate image processing techniques (mainly the alignment) used in single particle analysis.

ACKNOWLEDGMENTS

We thank Steven J. Ludtke for helpful discussions. This work was supported, in part, by National Institutes of Health grant NIH R01 GM 60635 (to Pawel A. Penczek).

Corresponding author Pawel A. Penczek's e-mail address is Pawel.A.Penczek@uth.tmc.edu.

REFERENCES