Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Structural Biology 144 (2003) 61–72

Journal of Structural Biology

www.elsevier.com/locate/yjsbi

A fast reconstruction algorithm for electron microscope tomography

Kristian Sandberg,[a,*] David N. Mastronarde,[b] and Gregory Beylkin[a]

[a] *Department of Applied Mathematics, University of Colorado at Boulder, CO 80309, USA*

[b] *Boulder Laboratory for 3-D Electron Microscopy of Cells, Department of Molecular, Cellular, and Developmental Biology, University of Colorado at Boulder, CO 80309, USA*

Received 23 May 2003, and in revised form 3 September 2003

Abstract

We have implemented a Fast Fourier Summation algorithm for tomographic reconstruction of three-dimensional biological data sets obtained via transmission electron microscopy. We designed the fast algorithm to reproduce results obtained by the direct summation algorithm (also known as filtered or R-weighted backprojection). For two-dimensional images, the new algorithm scales as $\mathrm{O}(N_\theta M \log M) + \mathrm{O}(MN \log N)$ operations, where N_θ is the number of projection angles and $M \times N$ is the size of the reconstructed image. Three-dimensional reconstructions are constructed from sequences of two-dimensional reconstructions. We demonstrate the algorithm on real data sets. For typical sizes of data sets, the new algorithm is 1.5–2.5 times faster than using direct summation in the space domain. The speed advantage is even greater as the size of the data sets grows. The new algorithm allows us to use higher order spline interpolation of the data without additional computational cost. The algorithm has been incorporated into a commonly used package for tomographic reconstruction.

Keywords: Electron tomography; Weighted backprojection; 3-D reconstruction algorithm; Unequally spaced fast Fourier transform

1. Introduction

In this paper, we describe a fast Fourier summation (FFS) algorithm for tomographic reconstruction of data obtained with a transmission electron microscope. For two-dimensional reconstructions, the algorithm scales as $\mathrm{O}(N_\theta M \log M) + \mathrm{O}(MN \log N)$ operations, where N_θ is the number of projection angles and $M \times N$ is the size of the reconstructed image. This should be compared to the direct summation in the space domain (also known as the filtered or R-weighted backprojection algorithm) which scales as $\mathrm{O}(N_\theta MN)$. Our algorithm has been applied to data of (current) typical sizes and is shown to be 1.5–2.5 times faster than the direct summation. Naturally, for larger data sets the improvement is even greater.

The problem of reconstructing an object from measured projections has a rich history and many applications. For example, X-ray tomography, radio astronomy, and seismic processing use results from the basic inversion technique first considered by Radon (1917). The Radon inversion formula was rediscovered by Cormack (1964) for X-ray tomography, and by Bracewell (1956) for radio astronomy. For an introductory overview of the subject, see Deans (1993). Reconstruction algorithms of computerized tomography can be found in books by Natterer (1986) and by Natterer and Wübbeling (2001). Reconstruction algorithms for transmission electron microscopy (TEM) imaging of biological specimens have been described by DeRosier and Klug (1968) via a Fourier based method, and by Gilbert (1972) via direct summation (for a review, see Frank, 1992).

The well-known Fourier slice theorem relates projection data to the Fourier transform of the image. The one-dimensional Fourier transform of the collected projection data corresponds to samples on a polar grid in the Fourier domain where, in our case, the polar angles are not necessarily equally spaced. Since the standard two-dimensional fast Fourier transform (FFT) requires sampling on an equally spaced rectangular grid, the inverse two-dimensional FFT cannot be used directly to reconstruct the image. Hence, fast Fourier reconstruction methods require some interpolation

[*] Corresponding author. Fax: +303-492-4066.
E-mail address: kristian.sandberg@colorado.edu (K. Sandberg).

1047-8477/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jsb.2003.09.013

scheme in the Fourier domain. A number of reconstruction algorithms based on interpolation in the Fourier domain are available in the literature, see for example, O'Sullivan (1985), Edholm and Herman (1987), Schomberg and Timmer (1995), Lanzavecchia and Bellon (1998), Waldén (2000), and Potts and Steidl (2001), and references therein.

Alternatively, fast hierarchical algorithms in the space domain have been proposed by Brandt et al. (2000) and by Basu and Bresler (2000). In these algorithms spatial interpolation is used at each level of subdivision which makes it difficult to control the resulting accuracy relative to the direct summation algorithm.

The interpolation techniques in the Fourier domain in O'Sullivan (1985), Schomberg and Timmer (1995), and Waldén (2000) use an approximation that also yields fast algorithms known as either non-equispaced fast Fourier transform (NFFT) (see Dutt and Rokhlin, 1993) or unequally spaced fast Fourier transform (USFFT) (see Beylkin, 1995). Compared to the interpolation techniques in the Fourier domain, the NFFT and USFFT algorithms use a (rigorously derived) nearly optimal relationship between the desired accuracy of the transform and the speed of the algorithm. In general, applications of NFFT or USFFT to problems of non-destructive evaluation are well understood (see e.g. Beylkin, 1995, p. 378), and can be found in Potts and Steidl (2001) and Natterer and Wübbeling (2001).

The specific difficulties of TEM that we address in this paper include unequal spacing and limited range of projection angles and, as far as the speed is concerned, a relatively small number of projections in typical measurements. Also, since the plane of the specimen may not be parallel to the tilt axis, additional corrections must be easily accommodated by the reconstruction algorithm. Due to the small number of projections, some of the fast algorithms mentioned above may not be faster than the direct summation algorithm, as correctly noted by Basu and Bresler (2000).

In this paper, we propose a technique that uses the one-dimensional USFFT for performing summation in the Fourier domain. The algorithm is designed for a limited angle reconstruction and permits unequally spaced sampling in the angular variable. Since the direct summation algorithm has been the standard tool in TEM for several decades, we match the result of FFS with that of the direct summation within any desired accuracy.

The method we propose guarantees accuracy relative to the direct summation while controlling the computational cost. As it is true for all Fourier methods, we gain flexibility in choosing interpolation schemes and applying corresponding filters in the Fourier domain at no extra cost. The FFS algorithm has been incorporated into the IMOD package (version 2.50 and higher; see http://www.bio3d.colorado.edu/imod) in 2001.

This paper is organized as follows. In Section 2, we introduce and formulate the inversion problem. We then give a brief review of the direct summation algorithm in the space domain. In Section 3, we derive an inversion formula for limited angle reconstruction in the Fourier domain. In Section 4, we discretize the inversion formula using the USFFT, select sampling, and show how to apply higher order interpolation. Finally, in Section 5 we demonstrate the algorithm on data sets collected at the Boulder Laboratory for 3-D Electron Microscopy of Cells at the University of Colorado at Boulder and compare the results with those obtained by using the direct summation algorithm. We do this for three-dimensional tomographic reconstructions as a part of the IMOD package (http://www.bio3d.colorado.edu/imod; Kremer et al., 1996).

2. Preliminaries

2.1. Formulation of the problem

We consider the problem of estimating the density of a biological specimen. We restrict ourselves to reconstructing densities in the plane and build the three-dimensional volume as a collection of two-dimensional slices. We consider a specimen illuminated by an electron beam and the intensity of the beam is measured after it passes through the specimen. This procedure is repeated for different tilt angles of the electron beam relative to the specimen as schematically shown in Fig. 1 below. In practice, the tilt angles θ are limited to some interval and typically range between $\simeq \pm 70^\circ$ with an angular separation of 1°–2°. The intensity is measured at M points for each angle θ_l, $l = 1, 2, \ldots, N_\theta$. The problem is then formulated as that of finding a discrete approximation to the density of the specimen, $g(x,z)$, on a rectangular (equally spaced) grid with M points in the x-direction and N points in the z-direction. The number of points in the x-direction is typically 500–2000. The number of points in the z-direction is usually less than the number of points in the x-direction.

As is customary, we assume that the intensity of the electron beam decays along straight lines through the specimen. For a comprehensive treatment of the physics of electron microscopy, see Reimer (1997). We consider a family of straight lines through the specimen, $C_{t,\theta} = \{(x,z) \in \mathbb{R}^2 | t = x\cos\theta + z\sin\theta\}$ (see Fig. 1). For a given projection angle θ, we define t as

$$t(x,z) = x\cos\theta + z\sin\theta, \tag{1}$$

and the function $R_\theta(t)$ as the line integral of the density $g(x,z)$ along the line $C_{t,\theta}$

$$R_\theta(t) = \int_{C_{t,\theta}} g(x,z)\,\mathrm{d}s, \tag{2}$$

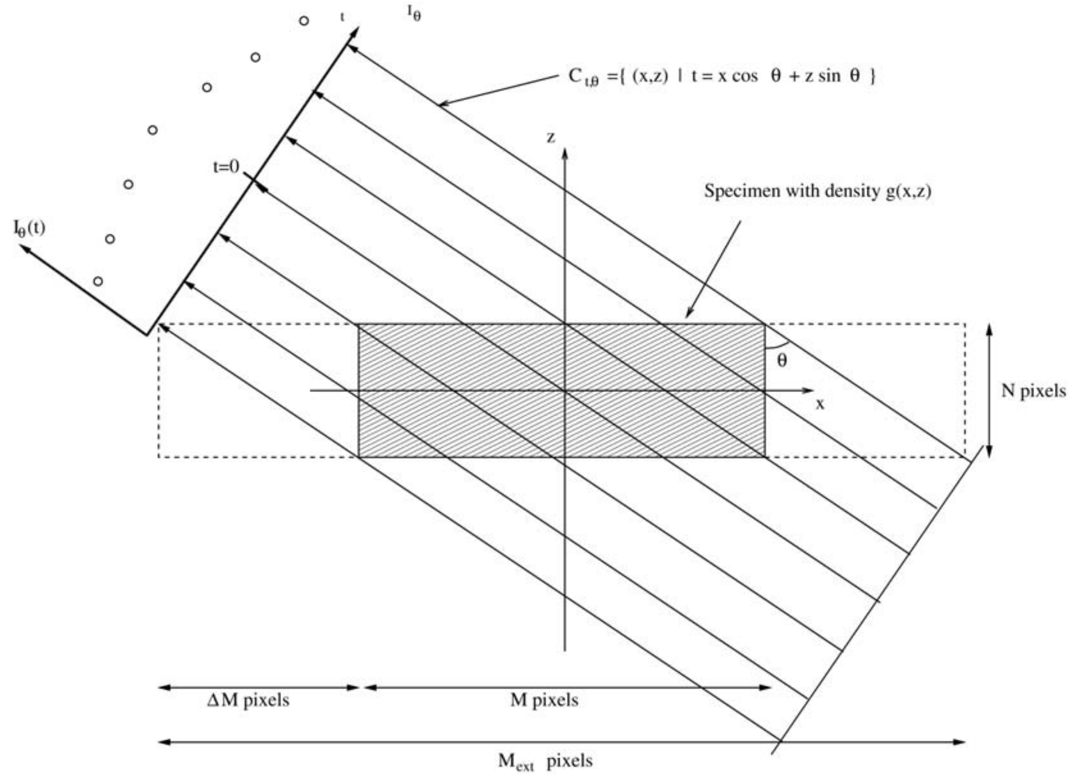
t
I_θ
$C_{t,\theta}$ ={ (x,z) | t = x cos θ + z sin θ }
z
t=0
Specimen with density g(x,z)
$I_\theta(t)$
θ
x
N pixels
ΔM pixels
M pixels
M_{ext} pixels

Fig. 1. Experimental setup. A specimen with density distribution $g(x,z)$ is illuminated by an electron beam through different angles θ. The variables ΔM and M_{ext} are discussed in Section 4.3.

where *ds* denotes the standard Euclidean measure on the line. We note that evaluating $R_\theta(t)$ for all lines is equivalent to computing the Radon transform of $g(x,z)$.

Assuming that the measured intensity is described by $I_\theta(t) = \mathrm{e}^{-R_\theta(t)}$, our goal is to approximate $g(x,z)$ by measuring $I_\theta(t)$. Sampling $R_\theta(t) = -\ln I_\theta(t)$ at a set of projection angles $\theta_1 < \cdots < \theta_l < \cdots < \theta_{N_\theta}$ and at a set of distances $t_0 < \cdots < t_k < \cdots < t_{M-1}$, yields the matrix

$$r_{kl} = R_{\theta_l}(t_k), \tag{3}$$

where $k = 0,1,\ldots,M-1$ and $l = 1,2,\ldots,N_\theta$. Each column l of the matrix in (3) contains all measurements for the angle θ_l.

The problem can now be formulated as follows: given the measurement data r_{kl}, find an approximation to $g(x_m,z_n)$, where the points x_m,z_n form a grid with $m = 1,2,\ldots,M$ and $n = 1,2,\ldots,N$. In TEM the total amount of input data is quite significant since such data are generated from measurements of a large number of two-dimensional slices of a specimen. Therefore, we not only need to find an accurate approximation of the density, but we also need to compute it in an efficient manner.

2.2. Inversion of the Radon transform

As is well known (see e.g. Deans, 1993) the two-dimensional density $g(x,z)$ can be recovered from the line integrals $R_\theta(t)$ in (2) as

$$g(x,z) = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} (\rho\ * R_\theta)(t(x,z))\,\mathrm{d}\theta, \tag{4}$$

where "*" denotes convolution and ρ is a filter with the Fourier transform given by

$$\hat{\rho}(\omega) = |\omega|. \tag{5}$$

In practice, this filter is often modified by a bandlimiting window. For example, if we use

$$\hat{\rho}(\omega) = \begin{cases} |\omega|, & |\omega| \leqslant \omega_{\mathrm{c}}, \\ \text{a smooth transition such that} & \\ \hat{\rho}(\pm\omega_{\mathrm{c}}) = \omega_{\mathrm{c}} \text{ and} \hat{\rho}\left(\pm\frac{1}{2}\right) = 0, & \omega_{\mathrm{c}} < |\omega| \leqslant \frac{1}{2}, \\ 0, & |\omega| > \frac{1}{2}, \end{cases} \tag{6}$$

where $0 < \omega_{\mathrm{c}} < \frac{1}{2}$ is a user specified parameter, then the density is approximated by

$$g(x,z) \approx \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} \hat{\rho}(\omega)\left(\int_{-\infty}^{\infty} R_\theta(s)\,\mathrm{e}^{2\pi\mathrm{i}s\omega}\,\mathrm{d}s\right)\mathrm{e}^{-2\pi\mathrm{i}\omega t}\,\mathrm{d}\omega\,\mathrm{d}\theta.$$

Here, t depends on θ, x, and z as in (1), but in what follows we may suppress this dependence in our notation.

2.3. Direct summation

If we assume that the electron beam is modeled by line integrals then reconstructing the density of a specimen from its projections can be viewed as the inversion of the Radon transform. Many reconstruction algorithms rely on this fact and solve the inversion problem by discretizing the inverse Radon transform (Deans, 1993; Gilbert, 1972). In this section, we describe the widely used direct summation algorithm (also known as the filtered or R-weighted back projection).

We discretize (4) and obtain

$$g(x_m, z_n) = \sum_{l=1}^{N_\theta} w_l(\rho \ * R_{\theta_l})(t(x_m, z_n)), \tag{7}$$

where $t(x_m, z_n)$ is given by (1) and w_l are weights to compensate for unequally spaced angles. For measurements performed over equally spaced angles, the weights w_l are usually set to one. Since we have measurements only for a discrete set of values of t, the elements $(\rho \ * R_{\theta_l})(t(x_m, z_n))$ are estimated from $\{(\rho \ * \ R_{\theta_l}) \ (t_k)\}_{k=0}^{M-1}$ by some interpolation scheme, usually piecewise linear interpolation. Let us summarize the steps for estimating the density $g(x, z)$ from the measurements of projections.

1. Filter the data to obtain $(\rho \ * \ R_{\theta_l})(t_k)$, $k = 0, 1, \ldots, M-1$ by:
   1.1. applying the FFT along the columns of the matrix r_{kl} defined by (3),
   1.2. multiplying each element of the transformed matrix by the (pre-computed) filter coefficients and the weights w_l if necessary, and
   1.3. applying the inverse FFT column-wise.
2. Sum the result of the previous step to obtain the density by:
   2.1. computing $t(x_m, z_n)$ for each given (x_m, z_n),
   2.2. finding $(\rho \ * R_{\theta_l})(t(x_m, z_n))$ by linearly interpolating $(\rho \ * R_{\theta_l})(t_k)$, and
   2.3. summing the result according to (7).

Step 2 dominates the computational cost since we have to sum over N_θ terms $N \times M$ times, for the total computational cost of $\mathrm{O}(N_\theta MN)$. Usually N_θ, M, and N are of the same order of magnitude so the above algorithm has the computational cost $\mathrm{O}(N^3)$.

3. Inversion formula in the Fourier domain

In this section, we derive an inversion formula in the Fourier domain that is equivalent to the direct summation formula in (7). We will show that if

$$x_m = -x_\mathrm{s} + m, \quad m = 1, 2, \ldots, M_\mathrm{f},$$

where x_s is a shift parameter that depends on the selection of the coordinate system in the x-variable, then $g(x_m, z_n)$ can be written as

$$\sum_{l=1}^{N_\theta} w_l(\rho \ * R_{\theta_l})(t(x_m, z_n)) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\sum_{l=1}^{N_\theta} v_l(\omega)\,\mathrm{e}^{-2\pi\mathrm{i}\omega z_n \tan\theta_l} \right) \mathrm{e}^{-2\pi\mathrm{i}\omega x_m}\,\mathrm{d}\omega,$$

where

$$v_l(\omega) = \frac{w_l}{\cos\theta_l}\,\mathrm{e}^{-2\pi\mathrm{i}x_\mathrm{s}\frac{\omega}{\cos\theta_l}}\hat{\rho}\left(\frac{\omega}{\cos\theta_l}\right)\left(\frac{\sin\frac{\pi\omega}{\cos\theta_l}}{\frac{\pi\omega}{\cos\theta_l}}\right)^2 \times \sum_{k=0}^{M-1} R_{\theta_l}(t_k)\,\mathrm{e}^{2\pi\mathrm{i}k\frac{\omega}{\cos\theta_l}}. \tag{8}$$

In Section 4.2, we will describe a numerical implementation of this formula that results in a fast $\mathrm{O}(N_\theta M \log M) + \mathrm{O}(MN \log N)$ algorithm. We note that shifting the coordinate system in x by a constant x_s may be necessary to account for the deformation of a specimen during the data collection.

In order to obtain (8), we will discretize the z-variable but keep x as a continuous variable until the very end of our derivation. Let $\{z_n\}_{n=1}^{N}$ be an equally spaced grid in the z-variable. If we fix $z = z_n$ while treating x as a continuous variable, then we write $g_n(x) = g(x, z_n)$ and $t_n(x) = t(x, z_n)$, where t is defined in (1). In the following derivation $\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)\,\mathrm{e}^{2\pi\mathrm{i}t\omega}\,\mathrm{d}t$ denotes the Fourier transform of a function $f(t)$.

By introducing the notation $f_l(t_n(x)) = w_l(\rho \ * R_{\theta_l})(t(x, z_n))$, we write the sum for the direct summation (7) as

$$g_n(x) = \sum_{l=1}^{N_\theta} f_l(t_n(x)), \tag{9}$$

where x is a continuous variable. To derive (8), we apply the Fourier transform with respect to x to both sides of (9) and perform the summation over angles in the Fourier domain. The Fourier transform of (9) with respect to x gives

$$\begin{aligned} \hat{g}_n(\omega) &= \sum_{l=1}^{N_\theta} \int_{-\infty}^{\infty} f_l(x\cos\theta_l + z_n \sin\theta_l)\,\mathrm{e}^{2\pi\mathrm{i}x\omega}\,\mathrm{d}x \\ &= \sum_{l=1}^{N_\theta} \frac{\mathrm{e}^{-2\pi\mathrm{i}\omega z_n \tan\theta_l}}{\cos\theta_l} \int_{-\infty}^{\infty} f_l(s)\,\mathrm{e}^{2\pi\mathrm{i}s\frac{\omega}{\cos\theta_l}}\,\mathrm{d}s \\ &= \sum_{l=1}^{N_\theta} v_l(\omega)\,\mathrm{e}^{-2\pi\mathrm{i}\xi_l(\omega)z_n}, \end{aligned} \tag{10}$$

where $\xi_l(\omega) = \omega\tan\theta_l$ and

$$v_l(\omega) = \frac{1}{\cos\theta_l} \int_{-\infty}^{\infty} f_l(s)\,\mathrm{e}^{2\pi\mathrm{i}s\frac{\omega}{\cos\theta_l}}\,\mathrm{d}s. \tag{11}$$

By definition, we have

$$\begin{aligned} f_l(t_n(x)) &= w_l(\rho \ * R_{\theta_l})(t(x, z_n)) \\ &= w_l \int_{-\infty}^{\infty} \hat{\rho}(\omega)\hat{R}_{\theta_l}(\omega)\,\mathrm{e}^{-2\pi\mathrm{i}\omega t}\,\mathrm{d}\omega, \end{aligned} \tag{12}$$

which combined with (11) gives us

$$v_l(\omega) = \frac{w_l}{\cos\theta_l}\hat{\rho}\left(\frac{\omega}{\cos\theta_l}\right)\hat{R}_{\theta_l}\left(\frac{\omega}{\cos\theta_l}\right). \tag{13}$$

Recall that $R_{\theta_l}(t_k)$ corresponds to a discrete set of measurements. In the direct summation algorithm, we use interpolation to define $R_{\theta_l}(t)$ for any t. To incorporate the linear interpolation, let us introduce the "hat" function, or the linear spline

$$\beta(t) = \begin{cases} 1 - |t|, & -1 < t < 1, \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

with its Fourier transform given by

$$\hat{\beta}(\omega) = \left(\frac{\sin(\pi\omega)}{\pi\omega}\right)^2.$$

We express the piecewise linear interpolation of the discrete data using $\beta(t)$ by defining

$$R_{\theta_l}(t) = \sum_{k=0}^{M-1} \beta(t - k + x_{\mathrm{s}}) r_{kl}. \tag{15}$$

It is easily verified that the function $R_{\theta_l}(t)$ is continuous with respect to t. Using (15) we have

$$\begin{aligned} \hat{R}_{\theta_l}(\omega) &= \sum_{k=0}^{M-1} r_{kl} \int_{-\infty}^{\infty} \beta(t - k + x_{\mathrm{s}})\, \mathrm{e}^{2\pi \mathrm{i} t\omega}\, \mathrm{d}t \\ &= \sum_{k=0}^{M-1} r_{kl}\, \mathrm{e}^{2\pi \mathrm{i}(k - x_{\mathrm{s}})\omega} \int_{-\infty}^{\infty} \beta(s)\, \mathrm{e}^{2\pi \mathrm{i} s\omega}\, \mathrm{d}s \\ &= \mathrm{e}^{-2\pi \mathrm{i} x_{\mathrm{s}}\omega} \hat{\beta}(\omega) \sum_{k=0}^{M-1} r_{kl}\, \mathrm{e}^{2\pi \mathrm{i} k\omega}. \end{aligned} \tag{16}$$

Combining (13) and (16) yields

$$\begin{aligned} v_l(\omega) &= \frac{w_l}{\cos\theta_l}\, \mathrm{e}^{-2\pi \mathrm{i} x_{\mathrm{s}} \frac{\omega}{\cos\theta_l}} \hat{\rho}\left(\frac{\omega}{\cos\theta_l}\right) \hat{\beta}\left(\frac{\omega}{\cos\theta_l}\right) \\ &\quad \times \sum_{k=0}^{M-1} r_{kl}\, \mathrm{e}^{2\pi \mathrm{i} k \frac{\omega}{\cos\theta_l}} \\ &= F_l(\omega) \hat{r}_l\left(\frac{\omega}{\cos\theta_l}\right), \end{aligned} \tag{17}$$

where

$$F_l(\omega) = \frac{w_l}{\cos\theta_l}\, \mathrm{e}^{-2\pi \mathrm{i} x_{\mathrm{s}} \frac{\omega}{\cos\theta_l}} \hat{\rho}\left(\frac{\omega}{\cos\theta_l}\right) \hat{\beta}\left(\frac{\omega}{\cos\theta_l}\right), \tag{18}$$

and

$$\hat{r}_l\left(\frac{\omega}{\cos\theta_l}\right) = \sum_{k=0}^{M-1} r_{kl}\, \mathrm{e}^{2\pi \mathrm{i} k \frac{\omega}{\cos\theta_l}}. \tag{19}$$

We note that the factor $F_l(\omega)$ is independent of the measured data once the angles θ_l are known.

The final step is computing $g_n(x)$ from $\hat{g}_n(\omega)$. By taking the inverse Fourier transform of (10) we arrive at

$$g_n(x) = \int_{-\infty}^{\infty} \left(\sum_{l=1}^{N_\theta} v_l(\omega)\, \mathrm{e}^{-2\pi \mathrm{i} \xi_l(\omega) z_n} \right) \mathrm{e}^{-2\pi \mathrm{i} \omega x}\, \mathrm{d}\omega, \tag{20}$$

where $\xi_l(\omega) = \omega \tan\theta_l$ and $v_l(\omega)$ are defined in (17).

Let us consider a smooth bandlimited filter ρ with its Fourier transform $\hat{\rho}$ defined by (6). Since $\hat{\rho}$ is zero for $|\omega| > \frac{1}{2}$ we observe that from (17) it follows that $v_l(\omega) = 0$ for $|\omega| > \frac{1}{2}$. Hence (20) reduces to

$$g_n(x) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\sum_{l=1}^{N_\theta} v_l(\omega)\, \mathrm{e}^{-2\pi \mathrm{i} \xi_l(\omega) z_n} \right) \mathrm{e}^{-2\pi \mathrm{i} \omega x}\, \mathrm{d}\omega. \tag{21}$$

The equation (21) is equivalent to the sum (9) used in the direct summation algorithm. We can obtain a fast reconstruction algorithm by computing the sums in (19) and (21) using the USFFT described in the Appendix A.

4. Implementation

4.1. Discretization

Let us evaluate (21) at pixel locations in the final image $g_n(x_m)$ where x_m is given by

$$x_m = -x_{\mathrm{s}} + m, \quad m = 1, 2, \ldots, M_{\mathrm{f}},$$

and x_{s} is a shift parameter due to possible deformations of the specimen. In order to avoid aliasing artifacts in the FFS algorithm, we construct an $M_{\mathrm{f}} \times N$-image where $M_{\mathrm{f}} > M$. The number of pixels M_{f} needed to avoid aliasing will be selected in Section 4.3. By discretizing ω as

$$\omega_k = -\frac{1}{2} + \frac{k}{M_{\mathrm{f}}}, \quad k = 1, 2, \ldots, M_{\mathrm{f}},$$

we approximate (21) by using the trapezoidal rule

$$\begin{aligned} g_n(x_m) &\simeq \frac{1}{M_{\mathrm{f}}} \sum_{k=-\frac{M_{\mathrm{f}}}{2}+1}^{\frac{M_{\mathrm{f}}}{2}} \left(\sum_{l=1}^{N_\theta} v_l\left(\frac{k}{M_{\mathrm{f}}}\right) \mathrm{e}^{-2\pi \mathrm{i} \xi_l(\frac{k}{M_{\mathrm{f}}}) z_n} \right) \mathrm{e}^{-2\pi \mathrm{i} \frac{k}{M_{\mathrm{f}}} x_m} \\ &= \frac{1}{M_{\mathrm{f}}} \sum_{k=-\frac{M_{\mathrm{f}}}{2}+1}^{\frac{M_{\mathrm{f}}}{2}} \left(\sum_{l=1}^{N_\theta} v_l\left(\frac{k}{M_{\mathrm{f}}}\right) \mathrm{e}^{-2\pi \mathrm{i} \xi_l(\frac{k}{M_{\mathrm{f}}}) z_n} \right) \mathrm{e}^{2\pi \mathrm{i} \frac{k}{M_{\mathrm{f}}} x_{\mathrm{s}}}\, \mathrm{e}^{-2\pi \mathrm{i} \frac{k}{M_{\mathrm{f}}} m} \\ &= \frac{1}{M_{\mathrm{f}}} \sum_{k=-\frac{M_{\mathrm{f}}}{2}+1}^{\frac{M_{\mathrm{f}}}{2}} \hat{g}_{nk}\, \mathrm{e}^{-2\pi \mathrm{i} k \frac{m}{M_{\mathrm{f}}}}, \end{aligned} \tag{22}$$

where

$$\hat{g}_{nk} = \mathrm{e}^{2\pi \mathrm{i} \frac{k}{M_{\mathrm{f}}} x_{\mathrm{s}}} \sum_{l=1}^{N_\theta} v_l\left(\frac{k}{M_{\mathrm{f}}}\right) \mathrm{e}^{-2\pi \mathrm{i} \xi_l(\frac{k}{M_{\mathrm{f}}}) z_n}. \tag{23}$$

Due to the construction of the filter ρ in (6), the integrand in (21) can be considered to be smooth and periodic in ω with period 1. Under this condition, the trapezoidal rule is rapidly convergent and we can achieve any desired accuracy by choosing M_{f} large enough. We have found that for our applications setting M_{f} to $1.5 - 2$ times the number of projections M usually suffices. We will demonstrate the accuracy of (22) in Section 5.2.1.

We note that the reconstruction formula (22) allows the grid in the z-variable to be shifted by a constant z_{s} by scaling v_l. Such shifts are essential in some cases when reconstructing three-dimensional volumes as discussed in Section 5.1.

4.2. Numerical algorithm

Our first goal in designing the FFS algorithm is to match the results with those of the direct summation algorithm. We do it for two reasons. First, since the direct summation algorithm has been used for a long time in TEM and significant experience has been accumulated for interpretation of the images, we avoid the issue of acceptance. Second, we demonstrate the flexibility of the FFS algorithm. As it turns out by changing parameters we can achieve a higher order interpolation in the input data in comparison with the linear interpolation used within the direct summation.

In order to match the direct summation algorithm, we consider linear interpolation and use a bandlimited version of the filter $\hat{\rho}(\omega) = |\omega|$. The discretization of the radial weighting filter is given by

$$\hat{\rho}(\omega_k) = \begin{cases} |\omega_k|, & |\omega_k| \leqslant \omega_\mathrm{c}, \\ \omega_\mathrm{c}\,\mathrm{e}^{-\frac{(\omega_k-\omega_\mathrm{c})^2}{2\omega_\mathrm{s}^2}}, & \omega_\mathrm{c} < |\omega_k| \leqslant 1/2, \\ 0, & |\omega_k| > 1/2, \end{cases}$$

where ω_c is a user specified cutoff frequency and ω_s is a user specified parameter for the Gaussian roundoff. This filter matches the filtering routinely used for direct summation, but other choices of filters are also possible. By an appropriate choice of the Gaussian roundoff, the filter is (approximately) continuous with respect to ω. The discretization of the linear interpolation filter is given by

$$\hat{\beta}(\omega_k) = \left(\frac{\sin(\pi\omega_k)}{\pi\omega_k}\right)^2.$$

We discuss other choices of this filter in Section 4.4 below.

Next we summarize the main steps of the FFS algorithm. To estimate the number of operations, let us consider N_θ projection angles with M samples each to reconstruct an image with $M \times N$ pixels. We consider a three-dimensional volume consisting of N_y slices. In what follows M_f (to be estimated in Section 4.3 below) is the number of spatial frequency modes in the x-direction.

Algorithm

1. Precomputation: For each angle θ_l and each frequency ω_m, compute $F_l(\omega_m)$ defined by (18).
2. For each image, evaluate (22):
   2.1. For each angle θ_l and each frequency ω_m, compute the sum (19) using the USFFT and multiply the result by $F_l(\omega_m)$ to obtain $v_l(\omega_m)$ in (17). See the Appendix A for details. Computational cost: $\mathrm{O}(N_\theta M_\mathrm{f}) + \mathrm{O}(N_\theta M \log M)$.
   2.2. For each frequency ω_m, compute the sum in (23) using the USFFT. See the Appendix A for details. Computational cost: $\mathrm{O}(M_\mathrm{f} N_\theta) + \mathrm{O}(M_\mathrm{f} N \log N)$.
   2.3. Compute the sum in (22) using the FFT. Computational cost: $\mathrm{O}(N M_\mathrm{f} \log M_\mathrm{f})$.

The steps are illustrated in Fig. 2. Using the symmetry of the Fourier transform $\hat{f}(\omega)$ of the real data $f(t)$, we double the speed of the algorithm by using $\hat{f}(-\omega) = \overline{\hat{f}(\omega)}$.

The algorithm assumes a limited range of projection angles. For the applications in this paper the angles are typically between ±70°. We note that the algorithm can be modified for the full angular range by first computing the density $g(x,z)$ based on the projection angles satisfying $|\theta_l| \leqslant 45°$, then interchanging the role of x and z, and computing g based on the projection angles satisfying $|\theta_l| > 45°$. Finally, we add the two reconstructions. A similar procedure was used by Potts and Steidl (2001).

In most applications, the angles θ_l and filters are the same for all slices, which means that the precomputation step in the algorithm only needs to be done once while steps 2.1–2.3 are repeated for each slice.

4.3. Oversampling

Applying the backprojection operator, we note that the collected data will produce non-zero intensity not only within but also *outside* the shaded portion of the specimen in Fig. 1. We refer to the shaded portion in Fig. 1 as "the region of interest." The extension of the support outside the region of interest does not cause any problems in the direct summation algorithm. However, by using (22) we see that since $g(x + M_\mathrm{f}, z) = g(x,z)$, we reconstruct a periodic image. If M_f is not large enough, the reconstructed area outside the region of interest will wrap around and overlap with the region of interest causing undesirable artifacts. By knowing the full extent of the reconstruction, M_ext in Fig. 1, we can choose the number of frequencies M_f large enough. Since the size of M_f affects the computational speed of the algorithm, it is important to choose M_f as small as possible.

From Fig. 1 we observe that

$$\Delta M = \frac{M_\mathrm{ext} - M}{2} = N \tan\theta_\mathrm{max},$$

where $\theta_\mathrm{max} = \max\{|\theta_l|\}_{l=1}^{N_\theta}$. This is illustrated in Fig. 3, where there is no wrap-around between the left and right sides of the reconstruction.

If M_f is smaller than the spatial support $M_\mathrm{ext} = M + 2\Delta M$, we will observe aliasing but as long as the overlapping region is outside the region of interest, no harm is done to the reconstruction. Hence, in order to avoid aliasing artifacts, we must choose

$$M_\mathrm{f} \geqslant M + N \tan\theta_\mathrm{max}. \tag{24}$$

This is illustrated in Fig. 4, where the wrap-around does not overlap the region of interest. For our applications, we note that choosing M_f according to (24) also is sufficient to achieve the desired accuracy in discretizing the integral in (21).

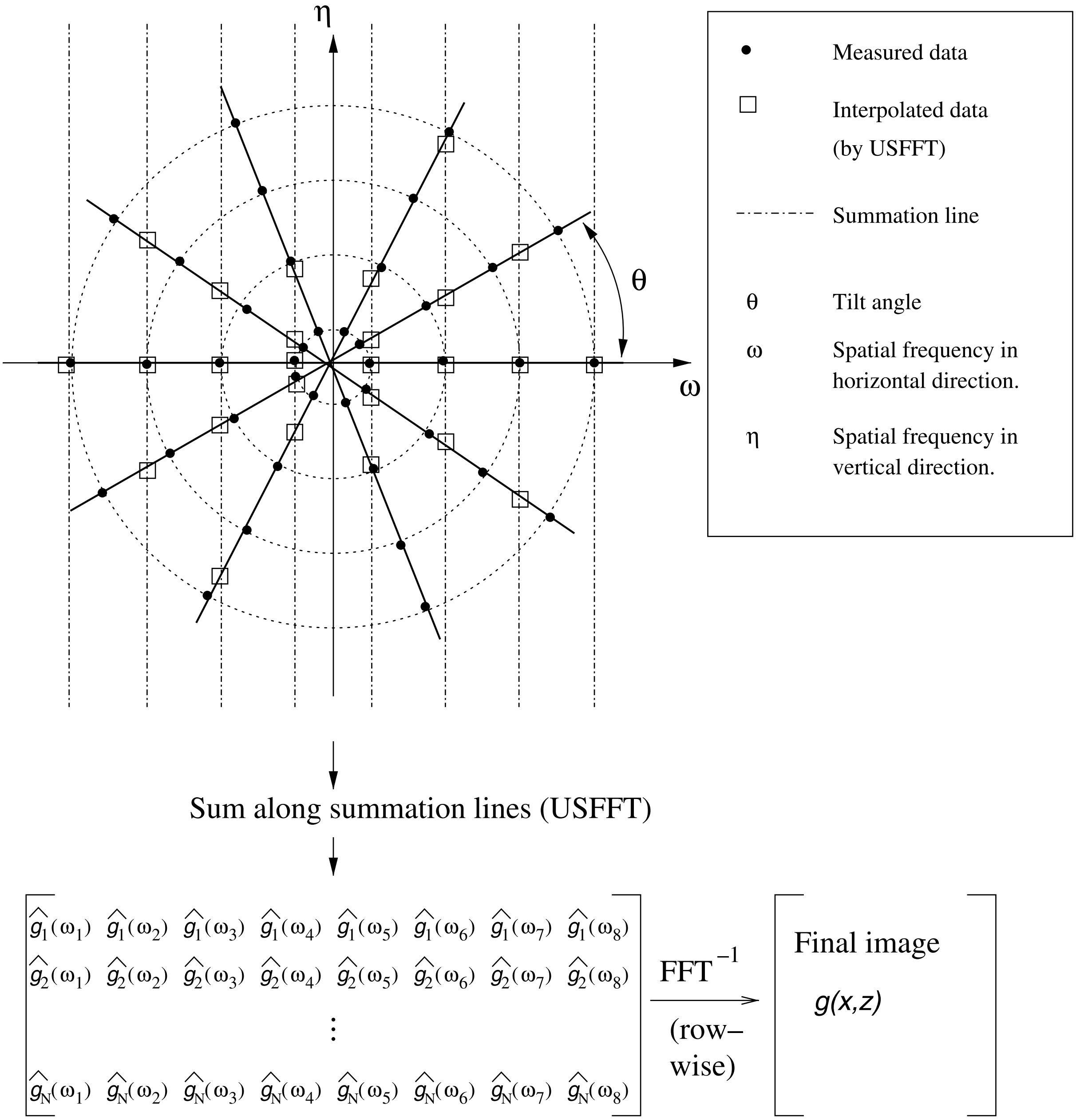
η
ω
θ
● Measured data
□ Interpolated data (by USFFT)
-·-·-·- Summation line
θ Tilt angle
ω Spatial frequency in horizontal direction.
η Spatial frequency in vertical direction.
Sum along summation lines (USFFT)
$\hat{g}_1(\omega_1)$ $\hat{g}_1(\omega_2)$ $\hat{g}_1(\omega_3)$ $\hat{g}_1(\omega_4)$ $\hat{g}_1(\omega_5)$ $\hat{g}_1(\omega_6)$ $\hat{g}_1(\omega_7)$ $\hat{g}_1(\omega_8)$
$\hat{g}_2(\omega_1)$ $\hat{g}_2(\omega_2)$ $\hat{g}_2(\omega_3)$ $\hat{g}_2(\omega_4)$ $\hat{g}_2(\omega_5)$ $\hat{g}_2(\omega_6)$ $\hat{g}_2(\omega_7)$ $\hat{g}_2(\omega_8)$
⋮
$\hat{g}_N(\omega_1)$ $\hat{g}_N(\omega_2)$ $\hat{g}_N(\omega_3)$ $\hat{g}_N(\omega_4)$ $\hat{g}_N(\omega_5)$ $\hat{g}_N(\omega_6)$ $\hat{g}_N(\omega_7)$ $\hat{g}_N(\omega_8)$
FFT^{-1}
(row–wise)
Final image
g(x,z)

Fig. 2. Interpretation of the FFS in the Fourier domain. The measurement given by the data acquisition is represented by the filled dots. Step 2.1 in the algorithm amounts to interpolating the data to the positions indicated by the squares. Step 2.2 of the algorithm computes the matrix $\hat{g}_n(\omega_k)$ column-wise by adding data along the summation lines. The final step of the algorithm computes the image by applying the inverse FFT row-wise on $\hat{g}_n(\omega_k)$.

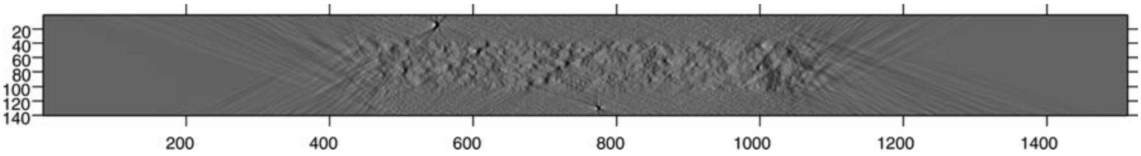

Fig. 3. For this reconstruction $M = 572$, $N = 140$, and $\theta_{\max} = 73.31°$. For this reconstruction, we chose $M_f = 1512$. We observe that for this choice of M_f, the entire support of the image fits in the reconstructed image.

Choosing M_f larger than M can be thought of as oversampling the image. The oversampling factor is given by the ratio M_f/M and it is desirable to make this factor as close to one as possible. For typical data sets, we have found that this oversampling factor ranges between 1.5 and 2. In (24), the size of M_f depends on

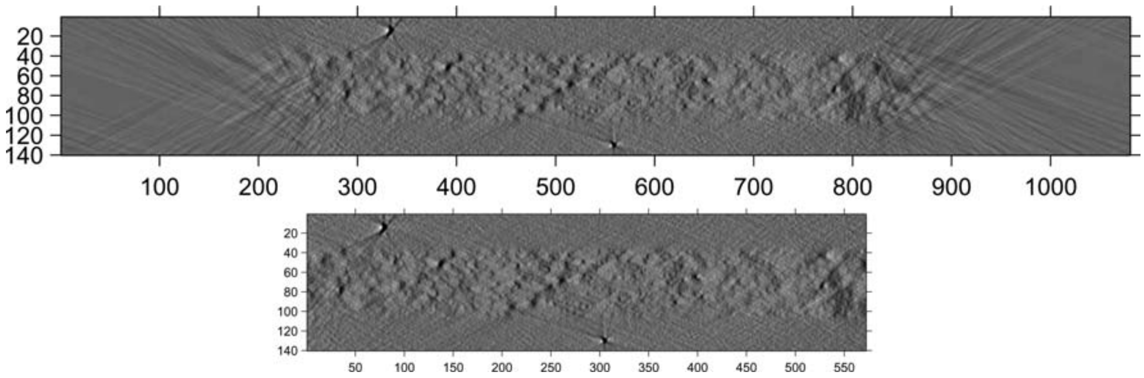

Fig. 4. We construct the same data set as in Fig. 3, but here we chose $M_{\mathrm{f}} = 1080$ which satisfies (24). Here, we do see some wrap-around in the left and right part of the upper image, but it does not overlap the region of interest cropped out in the lower image.

$N \tan \theta_{\max}$. We note that for physical reasons, large $\theta_{\max}$ requires thin specimens, which implies a small N. Hence, the oversampling factor is bounded in most applications and we have found it is typically sufficient to choose $M_{\mathrm{f}} \leqslant 2M$. Therefore, the total computational cost of the full three-dimensional reconstruction algorithm is given by $\mathrm{O}(N_y N_\theta M \log M) + \mathrm{O}(N_y N_\theta M_{\mathrm{f}} \log M_{\mathrm{f}})$. This should be compared to $\mathrm{O}(N_y M N N_\theta)$ for the direct summation. Actual speed comparisons are given in Section 5.2.2.

4.4. Interpolation

In Sections 3 and 4, we matched the FFS to reproduce the result of the direct summation algorithm that uses piecewise linear interpolation of the data. In direct summation, the interpolation is applied in step 2.2 of the algorithm in Section 2.3. We now show that higher order interpolation schemes can be easily incorporated into the FFS without additional computational costs.

As in the case of linear interpolation, the piecewise interpolation of higher order can be described in the space domain by using the B-splines, (Schoenberg, 1973). For the linear interpolation, the linear B-spline is given by the hat function in (14). Although higher order interpolation schemes may be cumbersome to implement in the space domain, in Fourier based algorithms the interpolation is implemented in the Fourier domain where it is simply a multiplication by an appropriate filter. From the definition of the B-splines as a repeated convolution of the characteristic function, it follows that the Fourier transform of the B-spline of odd order k is given by

$$\hat{\beta}^{(k)}(\omega) = \left(\frac{\sin(\pi\omega)}{\pi\omega}\right)^{k+1}, \quad k = 1, 3, 5, \ldots \tag{25}$$

Thus, all we need to use higher order interpolation in our reconstruction algorithm, is to apply (25) when computing the factor $F_l(\omega)$ given by (18). This is done in step 1 in the algorithm in Section 4.2.

Since the higher order interpolation filter (25) decays faster with order k, increasing the order of interpolation effectively low-passes the data. We leave the choice of the order as a parameter in the algorithm. Such choice should be guided by practical considerations and experiences. Since using high-order interpolation has a bandlimiting effect, it also implies that we need fewer frequencies for the reconstruction which, in turn, provides a marginal speed improvement.

5. Results

5.1. Incorporation of the FFS algorithm into IMOD

The FFS algorithm was incorporated into the Tilt program of the IMOD package (http://www.bio3d.colorado.edu/imod; Kremer et al., 1996). The basic framework of this program was originally written by Mike Lawrence while at the Medical Research Council in Cambridge. Before the speed comparisons reported here, the direct summation procedures in Tilt were optimized in two ways. First, all boundary checks were moved outside of the inner summation loops. Second, each line of input data was stretched by the cosine of the

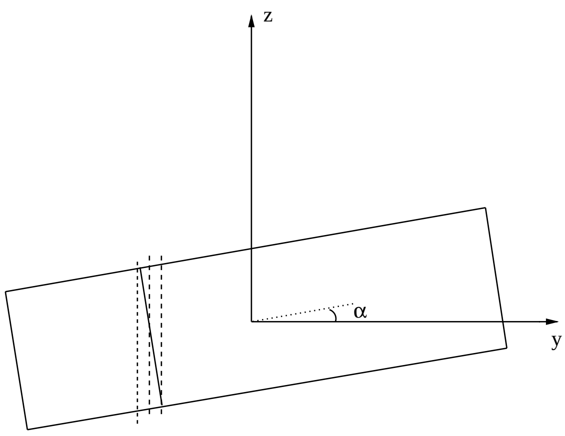

Fig. 5. x-axis tilting. Side view of section to be reconstructed where the section is tilted by the angle α around the x-axis (pointing out from the paper). Dashed lines are slices computed by the algorithm and the solid line is the output slice interpolated from vertical slices.

tilt angle, with the stretched data oversampled by a factor of two to minimize the filtering effect of interpolating twice. The result is that a line of stretched input data can be added into a line of the tomographic slice by stepping through the input line at a fixed interval and with fixed interpolation factors. The advantages of FFS would have been considerably higher than described here without these improvements to the original code.

Further developments in the Tilt program and in the FFS algorithm were spurred by the desire to correct for the plane of the specimen not being parallel to the tilt axis. When the specimen is tilted in the plane of the reconstruction, the thickness of the reconstruction must be increased to contain the material of interest, sometimes by as much as 50% for large data sets. To avoid this, the Tilt program can apply a tilt around the *x*-axis when computing the reconstruction by direct summation. However, this means that one output slice is based on multiple lines of input data, making fast backprojection unusable for this computation. To solve this problem, the FFS algorithm was modified to shift the output slice vertically (in the *z* dimension) by a chosen amount.

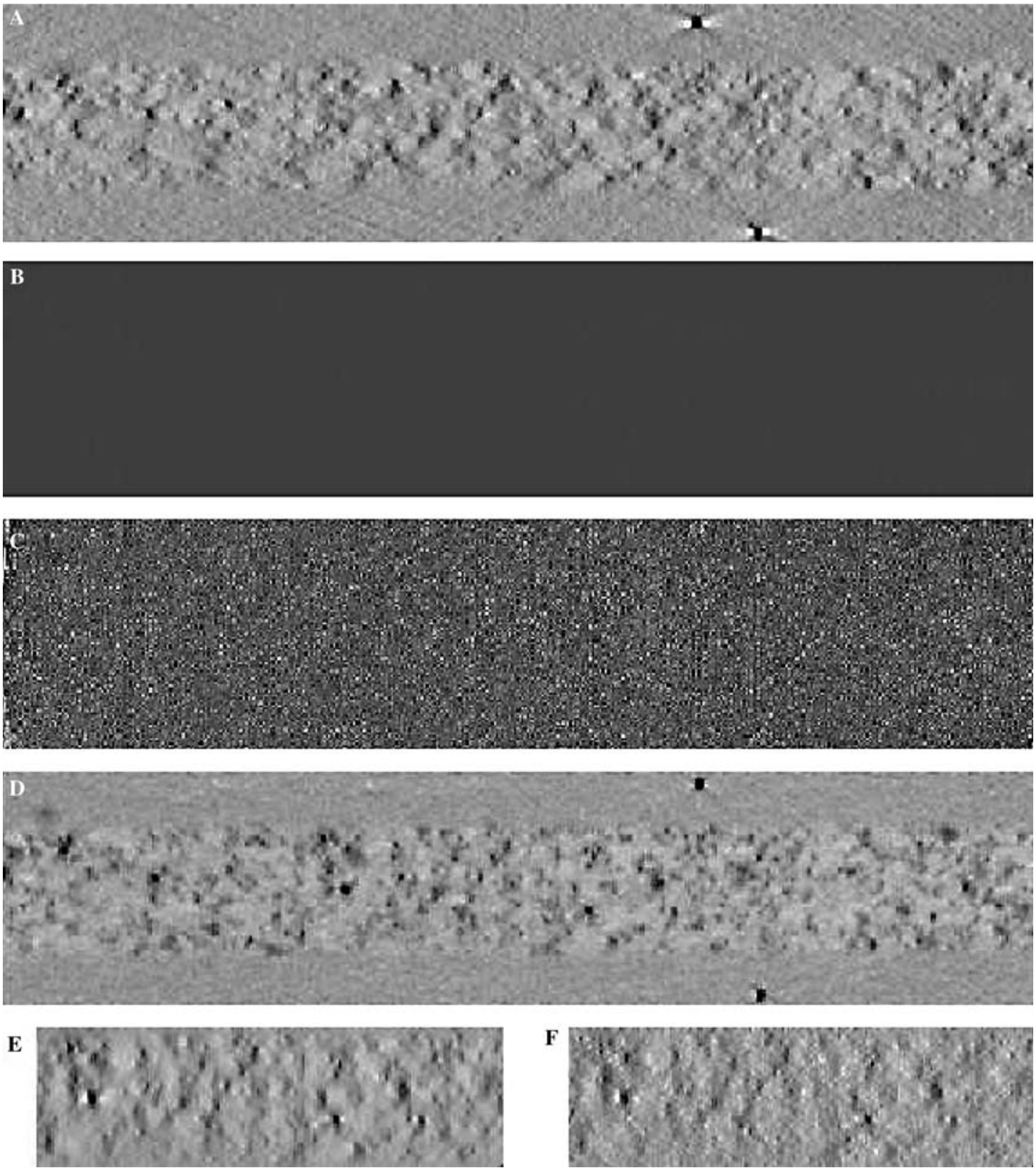

Fig. 6. Image reconstructions. (A) A test data set computed with the FFS algorithm. (B) The difference between the image in (A) and the corresponding slice reconstructed by direct summation with the same contrast as in (A). (C) The same dataset as in (B), but with the contrast amplified 29 times. (D) Test data set for the Fourier ring correlation test. (E) Portion of reconstruction using the FFS of the data set in (D) without noise added. (F) Portion of reconstruction using the FFS of the data set in (D) with noise added.

Tilt now uses FFS to compute slices perpendicular to the tilt axis, (dashed vertical line in Fig. 5), each shifted appropriately in *z* so that it covers the region needed for the reconstructed output slice. It then interpolates between these perpendicular slices to obtain an output slice tilted around the *x*-axis (tilted solid line in Fig. 5).

5.2. Tests

5.2.1. Accuracy

The data set shown here is based on images from the mitotic spindle of a dividing cell from the PtK cell line. The cell was high-pressure frozen, freeze-substituted, embedded in epon-araldite, and sectioned at 300 nm. The section was tilted between ±70 at 1.5° intervals and images were recorded on film in a JEOL microscope operating at 1000 keV. The grid was then rotated by 90° around the *z*-axis in the specimen holder and a second, similar tilt series was taken. Data were digitized at a pixel size of 2.3 nm using a CCD camera. The resolution of both the film and the CCD camera were good enough to ensure that the images have substantial information out to the Nyquist frequency. The overall modular transfer function (MTF) is estimated to be 30% at the Nyquist frequency. The single axis and combined tomograms were computed as described previously (Mastronarde, 1997).

Fig. 6 shows that the FFS algorithm produces essentially the same reconstruction as direct summation. One slice from the reconstruction of the test data set computed with FFS appears in Fig. 6A. The two densest features, one above and one below the sectioned material, are colloidal gold particles placed on the surface of the support film as fiducial markers for alignment. The difference between this image and the corresponding slice reconstructed by direct summation appears in Fig. 6B, at the same contrast as in Fig. 6A, and again in Fig. 6C with the contrast amplified 29 times. Aside from differences in the edge artifacts produced by the two procedures, the most prominent difference is at the gold particles, where the difference is about 1% of the density of the particles relative to the background. The differences within the section are smaller and are less than 2% of the range of densities found there.

For a quantitative assessment of the fidelity of reconstruction by the two methods, a sample volume was reprojected, then tomograms were built from the reprojections and compared with the original volume by Fourier ring correlation (Saxton and Baumeister, 1982). The combined dual-axis tomogram of our test data set was used as the sample volume; Fig. 6D shows the corresponding slice from this volume. This volume was considered suitable because the characteristic artifacts from single-axis tomography, namely the dark rays at the terminal angles of the tilt series and white shadows to the sides of densities, are much reduced there (Mastronarde, 1997). The volume was reprojected at 1.5° intervals between ±66°, either with no added noise or with added Poisson noise. The latter images were filtered by the estimated MTF of the film and digitizing apparatus. Noise levels equivalent to 1000, 3000 or 9000 electrons/pixel were explored, to represent both low dose and standard exposure situations. Several different high frequency cutoffs were applied in each of these situations. Figs. 6D and E show a central portion of the reconstruction from the noise-free data and the data with 3000 electrons/pixel of noise, respectively.

Fig. 7 shows the correlation between the Fourier transforms of the reconstruction and of the test volume, as a function of spatial frequency, averaged over 190 slices of each reconstruction. Clearly the two methods are equivalently good at reconstructing the test volume, regardless of the amount of noise added. From all of the cases tested, there was a tendency for the FFS to be slightly worse at low frequencies and slightly better at higher frequencies except near the Nyquist frequency; however, in practical terms these differences are insignificant.

5.2.2. Speed

The speed of the FFS algorithm was explored with a variety of data sizes and under several computer architectures. The width of the input data was varied from 256 to 4096 pixels; the number of projections was varied from 21 to 375, with an extreme angle of either 60° or 70°; and the thickness of the reconstruction was varied from 25 to 400 pixels. For a particular data size, CPU time was measured for the computation of one

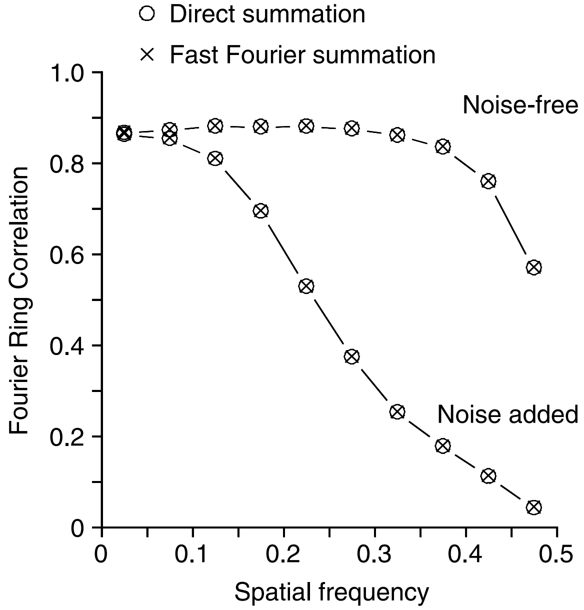

Fig. 7. Quantitative comparison between direct summation and FFS. The graph shows the Fourier ring correlation averaged over 190 slices of each reconstruction.

slice and ten slices, and the incremental time to compute nine slices were used to compare FFS and direct summation. Comparisons were done on SGI Octane computers with R10000 and R12000 processors, on a Sun Sparc Ultra-60, and on Intel-architecture computers with Pentium 3, Pentium 4, Athlon Thunderbird, and Athlon MP processors. For the SGI and Sun tests, programs were compiled with the native compilers; the Intel-architecture tests were done both with programs compiled with GNU compilers and with Intel compilers.

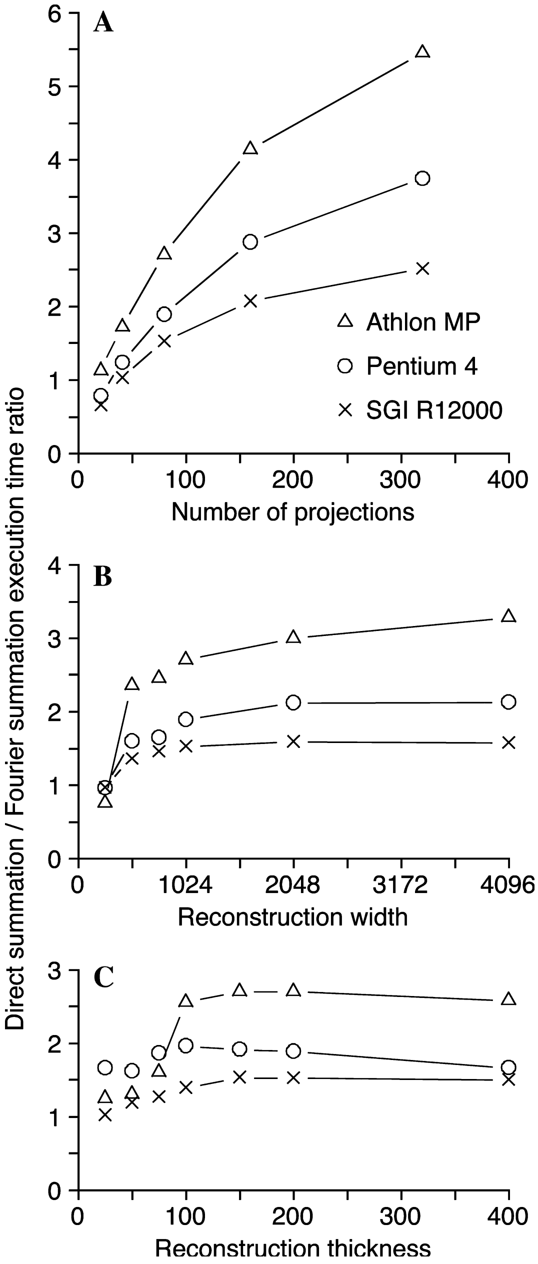

Fig. 8. Speed comparison. (Direct summation)/FFS execution time ratio for three computer architectures. (A) Dependence on number of projections with width 1024 and thickness 200. (B) Dependence on reconstruction width with 80 projections and thickness 200. (C) Dependence on thickness with 80 projections and width 1024.

The results in Fig. 8, from SGI, Pentium 4, and Athlon processors, illustrate the range of performance benefits found with FFS for typical data sizes. These graphs are not intended to demonstrate the order $O(N_\theta M \log M) + O(MN \log N)$ of the FFS algorithm since they show the dependence on each parameter separately with the other parameters fixed at typical values. The main point of Fig. 8 is to show performance gains for current typical sizes.

The strongest dependence is on the number of projections (Fig. 8A), where the speed benefit climbs 5-fold with an increase from 20 to 320 projections. This initial rise was most abrupt and pronounced with Athlon processors (e.g., Fig. 8C) and it reflects predominantly a slowing down of the direct summation per unit of computation rather than a speedup of FFS. Our interpretation is that the architecture of the Athlons is particularly favorable to the direct summation for small data sizes, but at some point a limit in cache or pipeline size is reached and the performance falls abruptly for direct summation.

The FFS is actually slower than direct summation for small data sets (Figs. 8A and B). To avoid using a slower algorithm, the Tilt program switches to direct summation when the width, thickness, or number of projections falls below a specified limit for the given computer architecture. Overall, the typical benefit from FFS is about 1.5- to 2.5-fold, with greater benefits available on some computers and with larger data sets.

Acknowledgments

This research was supported in part by DARPA Grants F49620-98-1-0491 and F30602-98-1-0154, University of Virginia Subcontract MDA-972-00-1-0016, and NSF/ITR Grant ACI-0082982 (G.B), NIH/NCRR Grant RR00592 and NIH Program Project Grant P01GM61306 to J. Richard McIntosh (D.N.M), and DARPA Grant F49620-98-1-0491, NSF/ITR Grant ACI-0082982, and NIH Program Project Grant P01GM61306 to J. Richard McIntosh (K.S.). The authors thank the anonymous reviewers and Martin Mohlenkamp for their useful and valuable comments.

Appendix A. Unequally spaced fast Fourier transform (USFFT)

As is well known, the discrete Fourier transform

$$\hat{u}_n = \sum_{k=0}^{N-1} u_k \mathrm{e}^{\pm 2\pi i k \frac{n}{N}}, \quad n = 0, 1, \ldots, N-1, \qquad \text{(A.1)}$$

can be computed in $\mathrm{O}(N\log N)$ operations using the fast Fourier transform (FFT) (Cooley and Tukey, 1965).

Since our algorithm uses the sums (23) and (19), we need a fast algorithm to compute the sums

$$\hat{u}_n = \sum_{k=1}^{M} u_k\, \mathrm{e}^{\pm 2\pi \mathrm{i}\xi_k n}, \quad n = -\frac{N}{2}, -\frac{N}{2}+1, \ldots, \frac{N}{2}-1, \tag{A.2}$$

and

$$\hat{u}(\xi_k) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} u_n\, \mathrm{e}^{\pm 2\pi \mathrm{i} n\xi_k}, \quad k = 1, 2, \ldots, M, \tag{A.3}$$

for a given real set of points $\{\xi_k\}_{k=0}^{M}$, where $|\xi_k| < 1/2$ for each k. We note that M may be different from N. The requirement $|\xi_k| < 1/2$ is not a constraint since by appropriate scaling and shifting it is always possible to satisfy this condition.

The sums in (A.2) and (A.3) can be computed using either the NFFT (Dutt and Rokhlin, 1993) or the USFFT (Beylkin, 1995). These algorithms effectively contain interpolation that guarantees the accuracy of the result. We use the algorithm in Beylkin (1995) which requires $C_1M + C_2N\log N$ operations and produces a prescribed accuracy. For this reason, we can match our results with those obtained via the direct summation.

Let us rewrite the sum (19) to match the form in (A.3). By a change of the summation index, we have

$$\hat{r}\left(\frac{\omega}{\cos\theta_l}\right) = \mathrm{e}^{\pi \mathrm{i}\omega\frac{M}{\cos\theta_l}} \sum_{k=-\frac{M}{2}}^{\frac{M}{2}-1} r_{k+\frac{M}{2},l}\, \mathrm{e}^{2\pi \mathrm{i} k\frac{\omega}{\cos\theta_l}}.$$

We now rewrite the sum (23) to match the form in (A.2). If $z_n = z_s + n$, where z_s is a constant, we have

$$\sum_{l=1}^{N_\theta} v_l\, \mathrm{e}^{-2\pi \mathrm{i}\xi_l z_n} = \sum_{l=1}^{N_\theta} \left(v_l\, \mathrm{e}^{-2\pi \mathrm{i}\xi_l z_s}\right) \mathrm{e}^{-2\pi \mathrm{i}\xi_l n},$$

where we have omitted the dependence of v_l and ξ_l on k/M_f.

References

Basu, S., Bresler, Y., 2000. $\mathrm{O}(N^2\log_2 N))$ filtered backprojection reconstruction algorithm for tomography. IEEE Trans. Med. Imag. 9 (10), 1760–1773.

Beylkin, G., 1995. On the fast fourier transform of functions with singularities. Appl. Comput. Harmon. Anal. 2, 363–381.

Bracewell, R.N., 1956. Strip integration in astronomy. Aust. J. Phys. 9, 198–217.

Brandt, A., Mann, J., Brodski, M., Galun, M., 2000. A fast and accurate multilevel inversion of the Radon transform. SIAM J. Appl. Math. 60 (2), 437–462.

Cooley, J.W., Tukey, J.W., 1965. An algorithm for the machine computation of complex Fourier series. Math. Comp. 19, 297–301.

Cormack, A.M., 1964. Representation of a function by its line integrals, with some radiological applications II. J. Appl. Phys. 35, 2908–2913.

Deans, S.R., 1993. The Radon Transform and Some of Its Applications. Krieger Publishing Company, Malabar.

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

Dutt, A., Rokhlin, V., 1993. Fast Fourier transforms for nonequispaced data. SIAM J. Sci. Stat. Comp. 14 (6), 1368–1393.

Edholm, P., Herman, G., 1987. Linograms in image reconstruction from projections. IEEE Trans. Med. Imag. 6, 301–307.

Frank, J. (Ed.), 1992. Electron Tomography. Plenum Press, New York.

Gilbert, P.F.C., 1972. The reconstruction of a three-dimensional structure from projections and its applications to electron microscopy II. Direct methods. Proc. R. Soc. Lond. B. 182, 89–102.

Kremer, J.R., Mastronarde, D.N., McIntosh, J.R., 1996. Computer visualization of three-dimensional image data using IMOD. J. Struct. Biol. 136, 110–118.

Lanzavecchia, S., Bellon, P.L., 1998. Fast computation of 3D Radon transform via a direct Fourier method. Bioinformatics 14 (2), 212–216.

Mastronarde, D.N., 1997. Dual-axis tomography: an approach with alignment methods that preserve resolution. J. Struct. Biol. 120, 343–353.

Natterer, F., 1986. The Mathematics of Computerized Tomography. Wiley, New York.

Natterer, F., Wübbeling, F., 2001. Mathematical Methods in Image Reconstruction. SIAM, Philadelphia, PA.

O'Sullivan, J.D., 1985. A fast sinc function gridding algorithm for Fourier inversion in computerized tomography. IEEE Trans. Med. Imag. 4 (4), 200–207.

Potts, D., Steidl, G., 2001. A new linogram algorithm for computerized tomography. IMA J. Numer. Anal. 21 (3), 769–782.

Radon, J., 1917. Über the bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. Berichte Sächsische Akademie der Wissenschaften Leipzig Math.-Phys. Kl. 69, 262–267.

Reimer, L., 1997. Transmission electron microscopy: physics of image formation and analysis (fourth Ed.), Springer Series in Optical Sciences, vol. 36.

Saxton, W.O., Baumeister, W., 1982. The correlation averaging of a regularly arranged bacterial cell envelope protein. J. Micros. 127, 127–138.

Schoenberg, I.J., 1973. Cardinal spline interpolation, SIAM Conference Board of the Mathematical Sciences Regional Conference Series in Applied Mathematics No. 12, Philadelphia, PA.

Schomberg, H., Timmer, J., 1995. The gridding method for image reconstruction by Fourier transformation. IEEE Trans. Med. Imag. 14 (3), 596–607.

Waldén, J., 2000. Analysis of the direct Fourier method for computer tomography. IEEE Trans. Med. Imag. 9, 211–222.