New Fourier reconstruction algorithms for computerized tomography

D. Potts [a] and G. Steidl [b]

[a] Medical University of Lübeck, Institute of Mathematics, Wallstraße 40, Lübeck 23560, Germany

[b] University of Mannheim, Institute of Mathematics, D 7/29, Mannheim 68131, Germany

ABSTRACT

In this paper, we propose two new algorithms for high quality Fourier reconstructions of digital $N \times N$ images from their Radon transform. Both algorithms are based on fast Fourier transforms for nonequispaced data (NFFT) and require only $\mathcal{O}(N^2 \log N)$ arithmetic operations. While the first algorithm includes a bivariate NFFT on the polar grid, the second algorithm consists of several univariate NFFTs on the so–called linogram.

Keywords: Fast Fourier transform for nonequispaced data, Radon transform, computerized tomography, gridding, linogram

1. INTRODUCTION

We are interested in efficient and high quality reconstructions of digital $N \times N$ medical images from their Radon transform. The standard reconstruction algorithm, the filtered backprojection, ensures a good quality of the images at the expense of $O(N^3)$ arithmetic operations. Fourier reconstruction methods reduce the number of arithmetic operations to $\mathcal{O}(N^2 \log N)$, a feature which will be of particular interest for future three–dimensional image processing. Unfortunately, the straightforward Fourier reconstruction algorithm suffers from unacceptable artifacts so that it is useless in practice. A better quality of the reconstructed images can be achieved by linogram algorithms,[1,2] the gridding algorithm,[3,4] the unified Fourier reconstruction algorithm (UFR–algorithm)[5,6] or by a recently developed algorithm by K. Fourmont.[7]

In this paper, we propose two new Fourier reconstruction algorithms having better or at least the same quality as the above algorithms. Both algorithms are based on recent developments in connection with the efficient computation of discrete Fourier transforms for nonequispaced data.

The algorithms are designed for two different sampling geometries in the Fourier domain of the image, the polar grid and the linogram. As the gridding algorithm and the UFR–algorithm, our first algorithm incorporates a bivariate gridding step. The second algorithm requires only univariate NFFTs.

This paper is organized as follows: In Section 2, we provide fast approximative algorithms for the computation of discrete Fourier transforms for nonequispaced data recently developed in.[8–10] We apply these algorithms in different ways for the numerical inversion of the Radon transform in Section 3. Finally, Section 4 presents numerical results.

2. FAST FOURIER TRANSFORMS FOR NONEQUISPACED DATA

While gridding methods in connection with efficient computations of discrete Fourier transforms for nonequispaced data were applied in digital signal processing for a long time,[11,3] the theoretical foundations for these methods, in particular the relation between the speed of the algorithm and the approximation error were developed only recently.[8–10,12] On the other hand, the theoretical examinations lead to a couple of improved and modified fast Fourier transform algorithms for nonequispaced data.[13,14,7,15] In the following, we shortly describe their basic idea.

Let $\Pi^d := [-\frac{1}{2}, \frac{1}{2})^d$ and $I_N := \{k \in \boldsymbol{Z}^d : -\frac{N}{2} \le k < \frac{N}{2}\}$, where the inequalities hold componentwise. For $v_j \in N\Pi^d$, and $f_k \in \boldsymbol{C}$, we are interested in the fast and robust computation of the discrete Fourier transforms

$$f(v_j) = \sum_{k \in I_N} f_k \, \mathrm{e}^{-2\pi \mathrm{i} k v_j / N} \qquad (j \in I_M) \tag{1}$$

D. Potts: E-mail: potts@math.mu-luebeck.de
G. Steidl: E-mail: steidl@math.uni-mannheim.de

and

$$h(k) := \sum_{j \in I_M} f_j \, \mathrm{e}^{-2\pi \mathrm{i} k v_j / N} \qquad (k \in I_N)\,, \tag{2}$$

i.e. either the nodes in time or frequency domain are equispaced. It is easy to check that once we have an algorithm for the efficient computation of (1), we can simply design an efficient algorithm for (2) which we will call the "transposed" algorithm. Therefore, we restrict our attention to (1). For an algorithm with both nonequispaced nodes in time and frequency domain see.[16] Straightforward computation of (1) requires $\mathcal{O}(N^d M^d)$ arithmetic operations, too much for the applications we have in mind. Only in case of equispaced nodes $v_j := j$ $(j \in I_N)$, the above values can be evaluated by the well–known *fast Fourier transform* (FFT) with only $\mathcal{O}(N^d \log N)$ arithmetic operations. To speed up the computation of (1), we suggest the following approximate procedure:

Instead of evaluating the 1–periodic trigonometric polynomial

$$f(v) := \sum_{k \in I_N} f_k \, \mathrm{e}^{-2\pi \mathrm{i} k v} \tag{3}$$

at the nodes $w_j := v_j/N \in \Pi^d$ $(j \in I_M)$, we intend to evaluate a function of the form

$$s_1(v) := \sum_{l \in I_n} g_l \, \varphi(v - \frac{l}{n})\,.$$

Here φ is an 1–periodic function which we will specify later and $n := \alpha N$ with an *oversampling factor* $\alpha > 1$. Switching to the frequency domain, we obtain

$$s_1(v) = \sum_{k \in I_n} \hat{g}_k \, c_k(\varphi) \, \mathrm{e}^{-2\pi \mathrm{i} k v} + \sum_{r \in Z^d \setminus \{0\}} \sum_{k \in I_n} \hat{g}_k \, c_{k+nr}(\varphi) \, \mathrm{e}^{-2\pi \mathrm{i}(k+nr)v} \tag{4}$$

with

$$\hat{g}_k := \sum_{l \in I_n} g_l \, \mathrm{e}^{2\pi \mathrm{i} k l/n}, \tag{5}$$

$$c_k(\varphi) := \int_{\Pi^d} \varphi(v) \mathrm{e}^{2\pi \mathrm{i} k v} \, \mathrm{d}v \quad (k \in \boldsymbol{Z}^d).$$

Let $c_k(\varphi) \neq 0$ $(k \in I_N)$. Since s_1 should be a good approximation of f, we suggest by comparing (3) with (4) to set

$$\hat{g}_k := \begin{cases} f_k / c_k(\varphi) & k \in I_N, \\ 0 & k \in I_n \setminus I_N. \end{cases} \tag{6}$$

Then the values g_l can be obtained from (5) by the reduced inverse d–variate FFT of size n. If φ is well–localized in time domain such that it can be approximated by a 1–periodic function ψ with $\operatorname{supp} \psi \cap \Pi^d \subseteq \frac{m}{n} \Pi^d$ $(m \ll n)$, then

$$f(w_j) \approx s_1(w_j) \approx s(w_j) = \sum_{l \in I_{n,m}(w_j)} g_l \, \psi(w_j - \frac{l}{n})\,, \tag{7}$$

where $I_{n,m}(w_j) := \{l \in I_N : n w_j - m \leq l \leq n w_j + m\}$. For fixed $w_j \in \Pi^d$, the above sum contains at most $(2m+2)^d$ nonzero summands.

In summary, we obtain the following algorithm for the fast computation of (1) with
$\mathcal{O}((\alpha N)^d \log(\alpha N) + (2m+1)^d M^d)$ arithmetic operations:

Algorithm 2.1 (NFFT)

Input: $N \in \boldsymbol{N}$, $\alpha > 1$, $n := \alpha N$, $w_j \in \Pi^d$, $f_k \in \boldsymbol{C}$ $(j \in I_M,\, k \in I_N)$.

Precomputation: $c_k(\varphi)$ $(k \in I_N)$, $\psi(w_j - \frac{l}{n})$ $(j \in I_M,\, l \in I_{n,m}(w_j))$.

1. Form $\hat{g}_k := f_k / c_k(\varphi) \quad (k \in I_N)$.

2. Compute by d–variate FFT
$$g_l := n^{-d} \sum_{k \in I_N} \hat{g}_k \, \mathrm{e}^{-2\pi \mathrm{i} k l/n} \quad (l \in I_n).$$

3. Set
$$s(w_j) := \sum_{l \in I_{n,m}(w_j)} g_l \, \psi(w_j - \frac{l}{n}) \quad (j \in I_M)\,.$$

Output: $s(w_j)$ approximate value of $f(w_j)$ $(j \in I_M)$.

The corresponding "transposed" algorithm for the fast computation of (2) reads as follows:

Algorithm 2.2 (**NFFT$^{\mathbf{T}}$**)

Input: $N \in \boldsymbol{N}$, $\alpha > 1$, $n := \alpha N$, $w_j \in \Pi^d$, $f_k \in \boldsymbol{C}$ $(j \in I_M, k \in I_N)$.
Precomputation: $c_k(\varphi) \quad (k \in I_N)$, $\psi(w_j - \frac{l}{n}) \quad (l \in I_n, j \in J_{n,m}(l))$.

1. Set
$$\tilde{g}_l := \sum_{j \in J_{n,m}(l)} f_j \, \psi(w_j - \frac{l}{n}) \quad (l \in I_n)\,,$$
where $J_{m,n} := \{j \in I_M : l - m \le n w_j \le l + m\}$ $(l \in I_n)$.

2. Compute by d–variate FFT
$$\tilde{c}_k(g) := n^{-d} \sum_{l \in I_n} \tilde{g}_l \, \mathrm{e}^{-2\pi \mathrm{i} k l/n} \quad (k \in I_N).$$

3. Form $\tilde{h}(k) := \tilde{c}_k(g)/c_k(\varphi)$ $(k \in I_N)$.

Output: $\tilde{h}(k)$ approximate value of $h(k)$ $(k \in I_N)$.

We will refer to Step 3 of Algorithm 2.1 and to Step 1 of Algorithm 2.2 as "*gridding step*", respectively. Both algorithms introduce the same approximation error
$$E(w_j) := |f(w_j) - s(w_j)| \;\le\; E_a(w_j) + E_t(w_j)$$
which by (4) and (7) splits into the *aliasing error* $E_a(w_j) := |f(w_j) - s_1(w_j)|$ and the *truncation error* $E_t(w_j) := |s_1(w_j) - s(w_j)|$. By (4) and (6), the aliasing error can be estimated by
$$E_a(w_j) \le ||f||_1 \max_{k \in I_N} \sum_{r \in Z^d \setminus \{0\}} \left| \frac{c_{k+nr}(\varphi)}{c_k(\varphi)} \right| \,,$$
where $||f||_1 := \sum_{k \in I_N} |f_k|$. By (7), (6) and (5), the truncation error fulfills
$$E_t(w_j) \le ||f||_1 \, n^{-d} \, (\max_{k \in I_N} |c_k(\varphi)|^{-1}) \sum_{l \in I_n} |\varphi(w_j - \frac{l}{n}) - \psi(w_j - \frac{l}{n})|.$$
Note that the truncation error may be zero, i.e. $\varphi = \psi$, if φ has compact support.

Thus, the whole approximation error $E(w_j)$ depends on the localization of the function φ in time and frequency domain. Clearly, by Heisenberg's uncertainty principle, there doesn't exist a window function φ with arbitrary good localization in both time and frequency domain. However, for various functions φ, it was proved that the approximation error $E(w_j)$ decays exponentially as a function of the "support width" m of ψ. In particular, we refer to

- [8,10,13,12] for estimates with (tensor products of) Gaussian bells, Gaussian bells tapered with Hanning windows or with sinc–kernels,
- [9,17] for estimates with (tensor products of) B–splines and to[13] for estimates with three–directional box–splines,
- [18,7] for estimates with (tensor products of) Kaiser–Bessel functions.

Note that other candidates for φ with good localization in time and frequency domain as for example prolate spheroidal functions are not suited for our algorithms. See also.[12]

In the following, we apply the Algorithms 2.1 and 2.2 with the tensor product φ of dilated periodized Gaussian bells

$$(\pi b)^{-1/2} \sum_{r \in Z} \mathrm{e}^{-(n(v+r))^2/b} \tag{8}$$

and the tensor product ψ of its truncated version

$$(\pi b)^{-1/2} \sum_{r \in Z} \mathrm{e}^{-(n(v+r))^2/b} \, \chi_{[-m,m]}(n(v+r)) \,, \tag{9}$$

where $b := \frac{2\alpha m}{(2\alpha-1)\pi}$ and where $\chi_{[-m,m]}$ denotes the characteristic function of $[-m,m]$. By,[16] it is sufficient to choose $m = 5$ to obtain an approximation error $\leq 10^{-5}$ (single precision).

Finally, it is remarkable that similar to the classical FFT, the NFFT is more robust with respect to roundoff errors introduced by the finite arithmetic of the computer than the straightforward summation of (1) or (2).[17]

3. APPLICATION OF NFFT AND NFFT$^{\mathrm{T}}$ IN COMPUTERIZED TOMOGRAPHY

In this section, we propose two new Fourier reconstruction algorithms for computerized tomography, where we restrict our attention to the standard parallel scanning geometry. More precisely, we are interested in the inversion of the *Radon transform*
$R : L_2(\Omega) \to L_2([-1,1] \times S^1; (1-s^2)^{-1/2})$,

$$Rf(s,\varphi) := \int\limits_{(x,\theta)=s} f(x)\,\mathrm{d}x \qquad (\theta := (\cos\varphi, \sin\varphi)^T)$$

based on the *Fourier Slice Theorem*

$$\hat{f}(\sigma\theta) = \int\limits_{-\infty}^{\infty} Rf(s,\varphi)\mathrm{e}^{-2\pi \mathrm{i} s\sigma}\,\mathrm{d}s = \hat{Rf}(\sigma,\varphi). \tag{10}$$

We suppose that $\operatorname{supp} f \subseteq \Omega := \{x \in \boldsymbol{R}^2 : ||x|| \leq 1\}$. We want to reconstruct f on the grid

$$(x_j, y_k) := \left(j\frac{2}{N}, k\frac{2}{N}\right) \qquad (j,k = -\frac{N}{2}, \ldots \frac{N}{2} - 1).$$

Let Rf be given at the grid points

$$(s_r, \varphi_t) = \left(r\frac{2}{R}, t\frac{\pi}{T}\right) \qquad (t = 0, \ldots, T-1; r = -\frac{R}{2}, \ldots, \frac{R}{2} - 1),$$

where, by Shannon's sampling theorem,[19] $R \geq N$ and $T \geq \frac{\pi R}{2}$.

The standard Fourier reconstruction method follows directly from (10) and consists of the following three steps:

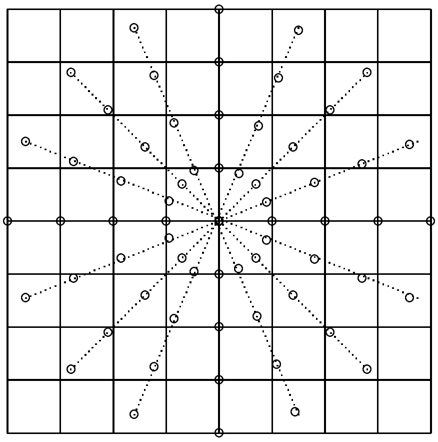
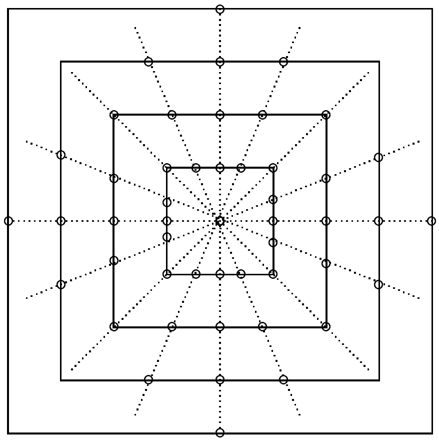
 

Figure 1. Polar grid of Algorithm 3.2 (left), Linogramm of Algorithm 3.3 (right).

Algorithm 3.1

1. Computation of

$$\hat{f}(\frac{m}{\gamma}\theta_t) = \hat{Rf}(\frac{m}{\gamma}, \varphi_t) := \frac{2}{R} \sum_{r=-\frac{R}{2}}^{\frac{R}{2}-1} Rf(r\frac{2}{R}, \varphi_t) \mathrm{e}^{-2\pi \mathrm{i} rm/(\frac{R\gamma}{2})} \quad (m = -\frac{R\gamma}{4}, \ldots, \frac{R\gamma}{4} - 1; t = 0, \ldots T-1)$$

   by T univariate FFT's of length $\frac{R\gamma}{2}$. Here $\frac{\gamma}{2} \geq 1$ is an oversampling factor.

2. Interpolation from the polar grid to the cartesian grid.

3. Computation of $f(x_j, y_k)$ $(j, k = -\frac{N}{2}, \ldots, \frac{N}{2} - 1)$ by bivariate FFT of size $\frac{\gamma}{2}N$.

The above algorithm produces essential artifacts so that it is useless in practice. In,[20] F. Natterer proved that most of these artifacts result from the interpolation in the radial direction in Step 2. This justifies a couple of higher quality Fourier reconstruction algorithms as:

- *Fourmont's algorithm*[7]
  By applying T univariate NFFTs of length N in Step 1, the algorithm requires only linear interpolations in angular directions in Step 2.

- *Gridding algorithm / UFR–algorithm*[3–6,2]
  Here Step 1 of Algorithm 3.1 is computed with oversampling factor $\frac{\gamma}{2} > 1$, i.e., $\frac{\gamma}{2} = 4$.[2] Instead of Step 2 a gridding step is performed which approximates the values of $\hat{f}$ on the cartesian grid. Step 3 coincides with the corresponding step of Algorithm 3.1.

- *Linogram algorithms*[1,2]
  By T chirp–z transforms of length $\frac{R\gamma}{2}$ in Step 1, the algorithm requires only linear interpolations in x–direction and in y–direction, i.e. in "nearly" angular directions in Step 2. Another version of the Linogram algorithm[1] computes linear interpolations in the Radon domain (sinogram). Then the linear interpolations in the Fourier domain in Step 2 can be avoided by using chirp–z transforms in Step 3, too.

In the following, we propose two Fourier reconstruction algorithms which *completely avoid linear interpolations.* Both algorithms are based on fast Fourier transforms for nonequispaced data as described in the previous section and require only $\mathcal{O}(N^2 \log N)$ arithmetic operations. We sketch the algorithms first and give some explanations later.

The first algorithm incorporates a bivariate NFFT$^{\mathrm{T}}$ and has therefore a bivariate gridding step as the gridding algorithm. An algorithm similar to Algorithm 3.2 is also in progress by K. Fourmont and F. Natterer.

Algorithm 3.2

1. Computation of
$$\hat{f}(\frac{m}{\gamma}\,\theta_t) = \hat{Rf}(\frac{m}{\gamma},\varphi_t) \quad (m = -\frac{R\gamma}{4},\ldots,\frac{R\gamma}{4}-1; t = 0,\ldots,T-1)$$
by T univariate FFT's of length $\frac{R\gamma}{2}$ ($\frac{\gamma}{2} \ge 1$).
2. Computation of
$$f(x_j, y_k) := \frac{\pi}{\gamma^2 T} \sum_{m=0}^{\frac{R\gamma}{4}-1} \sum_{t=-T}^{T-1} \nu_m\, \hat{f}(\frac{m}{\gamma}\theta_t)\, \mathrm{e}^{2\pi\mathrm{i}(jm\cos\varphi_t + km\sin\varphi_t)/(\frac{\gamma N}{2})} \qquad (j,k = -\frac{N}{2},\ldots,\frac{N}{2}-1)$$
by bivariate NFFT$^{\mathrm{T}}$, where
$$\nu_m := \begin{cases} \frac{1}{12} & m = 0\,, \\ m & m = 1,\ldots,\frac{R\gamma}{2}-1\,. \end{cases}$$

Step 1 coincides with the corresponding step of Algorithm 3.1. Further, $\frac{\gamma}{2}$ can be considered as oversampling factor. For images appearing in practice it is often sufficient to choose $\frac{\gamma}{2} = 1$.

The second algorithm is based on the linogram geometry. We combine T univariate NFFTs and $\frac{\gamma N}{2}$ univariate NFFT$^{\mathrm{T}}s$. Let T be divisible by 4.

Algorithm 3.3

1. Computation of
$$\hat{f}(\frac{m}{\gamma}\frac{1}{\cos\varphi_t}\theta_t) = \hat{Rf}(\frac{m}{\gamma}\frac{1}{\cos\varphi_t},\varphi_t) := \frac{2}{R}\sum_{r=-\frac{R}{2}}^{\frac{R}{2}-1} Rf(r\frac{2}{R},\varphi_t)\mathrm{e}^{-2\pi\mathrm{i}rm\frac{1}{\cos\varphi_t}/(\frac{R\gamma}{2})}$$
for $t = 0,\ldots,\frac{T}{4},\frac{3T}{4},\ldots,T-1$ and $m = \lceil -\frac{R\gamma}{4}\cos\varphi_t\rceil,\ldots,\lceil\frac{R\gamma}{4}\cos\varphi_t\rceil - 1$,
$$\hat{f}(\frac{m}{\gamma}\frac{1}{\sin\varphi_t}\theta_t) = \hat{Rf}(\frac{m}{\gamma}\frac{1}{\sin\varphi_t},\varphi_t) := \frac{2}{R}\sum_{r=-\frac{R}{2}}^{\frac{R}{2}-1} Rf(r\frac{2}{R},\varphi_t)\mathrm{e}^{-2\pi\mathrm{i}rm\frac{1}{\sin\varphi_t}/(\frac{R\gamma}{2})}$$
for $t = \frac{T}{4}+1,\ldots,\frac{3T}{4}-1$ and $m = \lceil -\frac{R\gamma}{4}\sin\varphi_t\rceil,\ldots,\lceil\frac{R\gamma}{4}\sin\varphi_t\rceil - 1$ by univariate NFFTs. Let the other values $\hat{f}(\frac{m}{\gamma}\frac{1}{\cos\varphi_t}\theta_t)$ and $\hat{f}(\frac{m}{\gamma}\frac{1}{\sin\varphi_t}\theta_t)$ ($m \in [-\frac{R\gamma}{4},\ldots,\frac{R\gamma}{4}-1]; m \in \boldsymbol{Z}$) be zero.
2. Computation of
$$\begin{aligned} f_1(x_j,y_k) &= \frac{\pi}{\gamma^2 T}\sum_{m=-\frac{R\gamma}{4}}^{\frac{R\gamma}{4}-1}\nu_m \sum_{t=-\frac{T}{4}}^{\frac{T}{4}-1}\frac{1}{\cos^2\varphi_t}\hat{f}(\frac{m}{\gamma},\frac{m}{\gamma}\frac{\sin\varphi_t}{\cos\varphi_t})\mathrm{e}^{2\pi\mathrm{i}\frac{\sin\varphi_t}{\cos\varphi_t}mk/(\frac{N\gamma}{2})}\mathrm{e}^{2\pi\mathrm{i}jm/(\frac{N\gamma}{2})} \\ f_2(x_j,y_k) &= \frac{\pi}{\gamma^2 T}\sum_{m=-\frac{R\gamma}{4}}^{\frac{R\gamma}{4}-1}\nu_m \sum_{t=-\frac{T}{4}+1}^{\frac{T}{4}}\frac{1}{\cos^2\varphi_t}\hat{f}(\frac{m}{\gamma}\frac{\sin\varphi_t}{\cos\varphi_t},\frac{m}{\gamma})\mathrm{e}^{2\pi\mathrm{i}\frac{\sin\varphi_t}{\cos\varphi_t}mj/(\frac{N\gamma}{2})}\mathrm{e}^{2\pi\mathrm{i}km/(\frac{N\gamma}{2})} \end{aligned}$$

$(j,k=-\frac{N}{2},\dots,\frac{N}{2}-1)$ by $\frac{R\gamma}{2}$ univariate NFFT$^{\mathrm{T}}$s of length $\frac{N\gamma}{2}$ for the inner sums and N univariate FFTs of length $\frac{N\gamma}{2}$ for the outer sum. Set

$$f(x_j,y_k)=\frac{1}{2}\operatorname{Re}(f_1(x_j,y_k)+f_2(x_j,y_k))\,.$$

Both algorithms use the fact that

$$\hat{f}(-\sigma\theta)=\overline{\hat{f}(\sigma\theta)}\,. \tag{11}$$

Based on (11) further improvements are possible which are incorporated in our implementation but will not be described in detail here. Further, as usual a “filter step” in the Fourier domain will be added after Step 1.

Let us give some more comments concerning our algorithms.

First step

For arbitrary fixed φ_t, let $h(s):=Rf(s,\varphi_t)$ and $\hat{h}(\sigma):=\hat{Rf}(\sigma,\varphi_t)$. In the first step of both algorithms, we discretize the integral

$$\hat{h}(\sigma)=\int_{-1}^{1}h(s)\mathrm{e}^{-2\pi\mathrm{i}s\sigma}\,\mathrm{d}s.$$

By Poisson's summation formula, we obtain

$$\hat{h}(\sigma)+\sum_{\substack{n\in Z\\ n\neq 0}}\hat{h}(\sigma+n\frac{R}{2})=\frac{2}{R}\sum_{r=-\frac{R}{2}}^{\frac{R}{2}-1}h(r\frac{R}{2})\mathrm{e}^{-2\pi\mathrm{i}r\sigma/(\frac{R}{2})}\,. \tag{12}$$

Since we want to reconstruct only details of f of size $\geq\frac{2}{N}$ and $R\geq N$, we can assume by Shannon's sampling theorem, that $\hat{h}(\sigma)$ is neglectable small for $|\sigma|>\frac{R}{4}$. Thus, the right–hand side of (12) is a good approximation of $\hat{h}(\sigma)$ for $\sigma\in[-\frac{R}{4},\frac{R}{4}]$.

Second step

In the second step of the algorithms, we compute a discretized form of the integral

$$\begin{aligned}f(x,y)&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\hat{f}(u,v)\mathrm{e}^{2\pi\mathrm{i}(ux+vy)}\,\mathrm{d}u\mathrm{d}v\\&=\int_{0}^{\infty}\sigma\int_{-\pi}^{\pi}\hat{f}(\sigma\cos\varphi,\sigma\sin\varphi)\mathrm{e}^{2\pi\mathrm{i}\sigma(\cos\varphi\,x+\sin\varphi\,y)}\,\mathrm{d}\varphi\mathrm{d}\sigma.\end{aligned}$$

Since the inner integral considered as function of σ is even, the above formula can be rewritten as

$$f(x,y)=\frac{1}{2}\int_{-\pi}^{\pi}\int_{-\infty}^{\infty}|\sigma|\hat{f}(\sigma\cos\varphi,\sigma\sin\varphi)\mathrm{e}^{2\pi\mathrm{i}\sigma(\cos\varphi\,x+\sin\varphi\,y)}\,\mathrm{d}\varphi\mathrm{d}\sigma\,. \tag{13}$$

We consider the inner integral. For arbitrary fixed $(x,y)\in[-1,1]^2$ and $\varphi\in[-\pi,\pi]$, we set

$$\begin{aligned}g(\sigma)&:=|\sigma|\hat{f}(\sigma\cos\varphi,\sigma\sin\varphi)\mathrm{e}^{2\pi\mathrm{i}\sigma(\cos\varphi\,x+\sin\varphi\,y)},\\\hat{g}(v)&:=\int_{-\infty}^{\infty}g(\sigma)\mathrm{e}^{2\pi\mathrm{i}v\sigma}\,\mathrm{d}\sigma.\end{aligned}$$

Then we obtain by Poisson's summation formula for the discretization on the polar grid

$$\hat{g}(0) + \sum_{\substack{n\in Z\\ n\neq 0}} \hat{g}(\gamma n) = \frac{1}{\gamma}\sum_{m\in Z} g\left(\frac{m}{\gamma}\right),$$

and on the linogram

$$\hat{g}(0) + \sum_{\substack{n\in Z\\ n\neq 0}} \hat{g}(\gamma\cos\varphi\, n) = \frac{1}{\gamma\cos\varphi}\sum_{m\in Z} g\left(\frac{m}{\varphi\cos\varphi}\right) \qquad (\varphi\in[-\frac{\pi}{4},\frac{\pi}{4}])$$

and since $\hat{f}(\sigma\cos\varphi,\sigma\sin\varphi)$ is neglectable small for $|\sigma| > \frac{R}{4}$,

$$\hat{g}(0) + \sum_{\substack{n\in Z\\ n\neq 0}} \hat{g}(\gamma n) \approx \frac{1}{\gamma}\sum_{m=-\frac{R\gamma}{4}}^{\frac{R\gamma}{4}-1} g\left(\frac{m}{\gamma}\right)$$

and

$$\hat{g}(0) + \sum_{\substack{n\in Z\\ n\neq 0}} \hat{g}(\gamma\cos\varphi\, n) \approx \frac{1}{\gamma\cos\varphi}\sum_{m=\lceil-\frac{R\gamma}{4}\cos\varphi\rceil}^{\lceil\frac{R\gamma}{4}\cos\varphi\rceil-1} g\left(\frac{m}{\varphi\cos\varphi}\right) \qquad (\varphi\in[-\frac{\pi}{4},\frac{\pi}{4}]),$$

respectively. The aliasing error on the left–hand side becomes smaller with increasing γ.

Further, $\cos\varphi \geq \frac{\sqrt{2}}{2}$ $(\varphi\in[-\frac{\pi}{4},\frac{\pi}{4}])$. Consequently, if we choose $\gamma=\gamma_p$ in the first discretization, we should choose $\gamma=\sqrt{2}\gamma_p$ in the second discretization to obtain a similar aliasing error.

Under the assumption that $\tilde{g}(\sigma) := \sigma\hat{f}(\sigma\cos\varphi,\sigma\sin\varphi)e^{2\pi i\sigma(\cos\varphi\, x+\sin\varphi\, y)}$ has "essential" bandwidth $\ll\gamma$, respectively $\ll\gamma\cos\varphi$, the aliasing error can be estimated as in[21] by

$$\sum_{\substack{n\in Z\\ n\neq 0}} \hat{g}(\gamma n) \approx \frac{1}{6\gamma^2}\hat{f}(0,0), \tag{14}$$

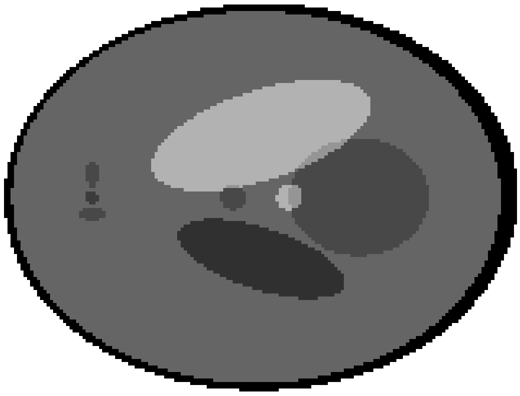
$$\sum_{\substack{n\in Z\\ n\neq 0}} \hat{g}(\gamma\cos\varphi n) \approx \frac{1}{6\gamma^2\cos^2\varphi}\hat{f}(0,0)\,. \tag{15}$$

Since we only want to reconstruct details of f of size $\geq \frac{2}{N}$, we can discretize the outer integral in (13) with small aliasing error by the trapezoidal rule at the nodes $\varphi_t = t\frac{\pi}{T}$ $\quad(t=-T,\ldots,T-1)$ if $T\geq\pi\frac{R}{2}$. This results in Step 2 of our Algorithms 3.2 and 3.3. In particular, (14) and (15) explain the coefficients of $\hat{f}(0,0)$ in Step 2 of our algorithms. Note that the coefficient $\frac{2\pi}{12\gamma^2}$ of $\hat{f}(0,0)$ in Algorithm 3.2 is nearly the area $\frac{2\pi}{8\gamma^2}$ of the circle with radius $\frac{1}{2\gamma}$ and that the coefficient of $\hat{f}(0,0)$ in Algorithm 3.3 is nearly the area $\frac{1}{\gamma^2}$ of the quadrat with length $\frac{1}{\gamma}$.

4. NUMERICAL EXAMPLES

A commonly examined model in computerized tomography is the Shepp–Logan Phantom of the brain. This model consists of several ellipses so that its Radon transform can be evaluated analytically. In order to get a sampled version of the phantom and its Radon transform we have use the software packages "RadonAna".[22] The original image (Figure 2 (left)) is of size $N\times N = 170\times 170$ and its sinogram of size $R\times T = 170\times 600$. Figure 2 (right) presents the reconstructed image obtained by the filtered backprojection. Here we have used the software package "iradon".[22]

The reconstructed images in Figure 3 were computed by the straightforward Algorithm 3.1 (Figure 3 (left)) and by the linogram algorithm (we used Algorithm 2.1) with linear interpolation in x and y directions[2] (Figure 3 (right)). We have used the filter sinc^3 in the Fourier domain. The speed of these algorithms is about 14 times faster than those of the filtered backprojection at the cost of the image quality.



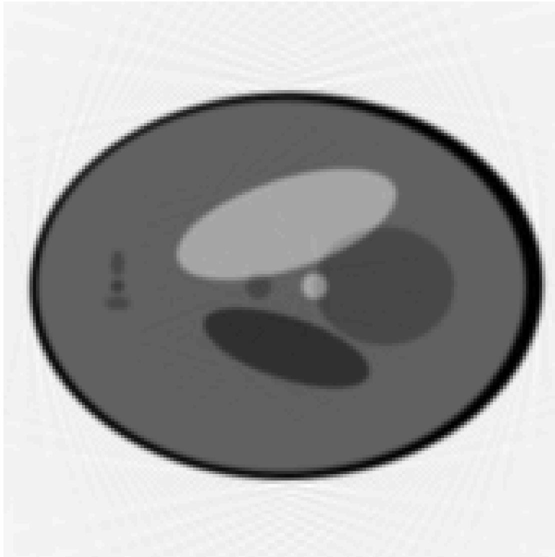
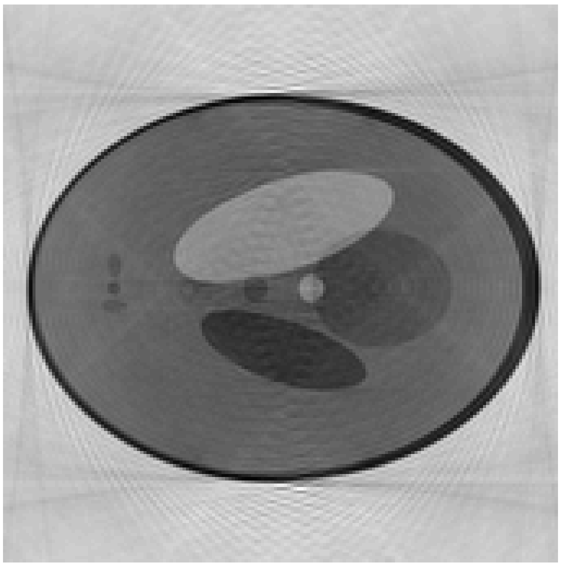


Figure 2. Shepp–Logan phantom, original (left), reconstructed image by filtered backprojection (right).



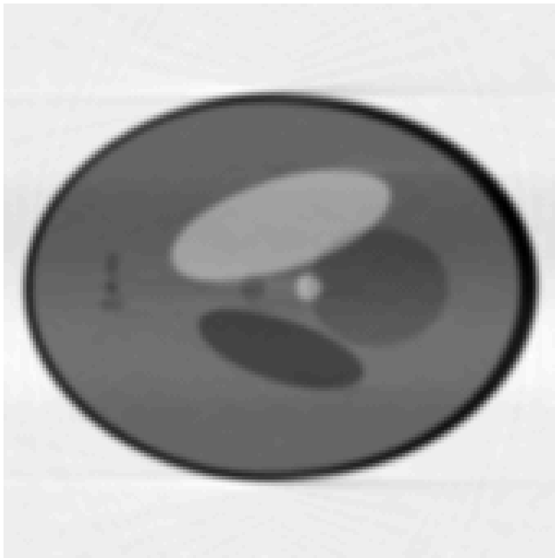


Figure 3. Reconstructed image by Algorithm 3.1 (left) and by the linogram algorithm (right).

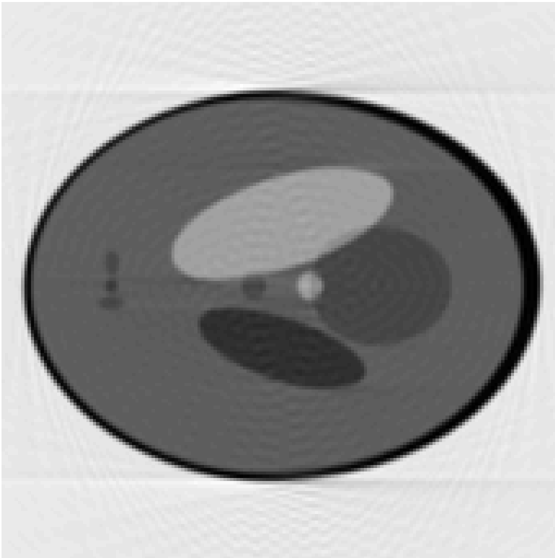
Our final numerical examples show that our Algorithms 3.2 and 3.3 lead to high quality images similar to the filtered backprojection. Both algorithms (as the above three algorithms) were applied with an oversampling factor $\frac{\gamma}{2} = \frac{256}{170}$ ($\approx \sqrt{2}$). Oversampling seems to be only necessary for artifical images containing high frequenies. The algorithms were implemented in C on a SGI O2.

We have computed Step 2 of Algorithm 3.2 by applying Algorithm 2.2 with $d = 2$, oversampling factor $\alpha = 2$, $m = 3$ and $b = \frac{2\alpha m}{(2\alpha-1)\pi} = \frac{4}{\pi}$. Further, we have chosen φ as the tensor product of dilated periodized Gaussian bells (8) and ψ as the tensor product of its truncated version (9).

Step 1 of Algorithm 3.3 was realized by Algorithm 2.1 with $d = 1$, oversampling factor $\alpha = 2$, $m = 5$ and

$b = \frac{20}{3\pi}$. Again, we have chosen φ as dilated periodized Gaussian bell (8) and ψ as its truncated version (9). Step 2 of Algorithm 3.3 was computed by Algorithm 2.2 with the same parameters as in Step 1. Further, we have used the sinc filter in the Fourier domain.

Figure 4 shows the reconstructed images obtained by our Algorithms 3.2 and 3.3, respectively.



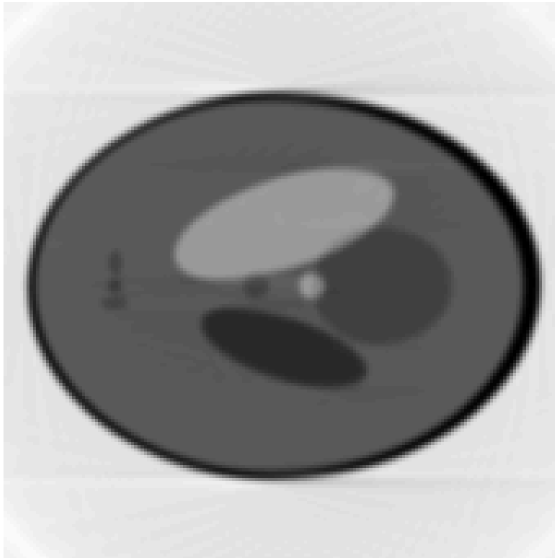


Figure 4. Reconstructed image by Algorithm 3.2 (left) and by Algorithm 3.3 (right).

Since the bivariate gridding step is rather expansive, Algorithm 3.2 is about three times slower than Algorithm 3.3 which requires nearly the same time as the linogram algorithm. A detailed comparison of the speed of the algorithms will be given in a forthcoming paper.

Finally, note that we can detect differences in the quality of the reconstructed images much better if we are given coloured images. For this we refer to http://www.math.mu-luebeck.de/potts/radon/spie.html.

REFERENCES

1. P. Edholm and G. Herman, "Linograms in image reconstruction from projections," IEEE *Trans. Med. Imag.* **6**, pp. 301 – 307, 1987.
2. J. Schulte, "Fourier–Rekonstruktionen in der Computer–Tomographie," Master's thesis, University of Münster, 1994. `http://wwwmath.uni-muenster.de/inst/num/Abschlussarbeiten/Schulte/`.
3. J. O'Sullivan, "A fast sinc function gridding algorithm for Fourier inversion in computer tomography," IEEE *Trans. Med. Imag.,* **4**, pp. 200 – 207, 1985.
4. H. Schomberg and J. Timmer, "The gridding method for image reconstruction by Fourier transformation," IEEE *Trans. Med. Imag.,* **MI–14**, pp. 596 – 607, 1995.
5. M. Kaveh and M. Soumekh, "Computer–assisted diffraction tomography," in *Image Recovery: Theory and Applications*, H. Stark, ed., pp. 369 – 413, (Academic Press, Orlando), 1987.
6. M. Kaveh, M. Soumekh, and J. Greenleaf, "Signal processing for diffraction tomography," IEEE *Transactions on Sonics and Ultrasonics,* **SU–31**, 1984.
7. K. Fourmont, *Schnelle Fourier–Transformation bei nichtäquidistanten Gittern und tomographische Anwendungen.* PhD thesis, University of Münster, 1999.
8. A. Dutt and V. Rokhlin, "Fast Fourier transforms for nonequispaced data," *SIAM J. Sci. Statist. Comput.* **14**, pp. 1368 – 1393, 1993.
9. G. Beylkin, "On the fast Fourier transform of functions with singularities," *Appl. Comput. Harmon. Anal.* **2**, pp. 363 – 381, 1995.

10. G. Steidl, "A note on fast Fourier transforms for nonequispaced grids," *Adv. Comp. Math* **9**, pp. 337 – 352, 1998.
11. R. A. Scramek and F. R. Schwab, "Imaging," in *Astronomical Society of the Pacific Conference, Vol 6*, F. R. S. R. Perley and A. Bridle, eds., pp. 117 – 138, 1988.
12. A. J. W. Duijndam and A. M. Schonewille, "Nonuniform fast Fourier transform," *Preprint* , 1997.
13. B. Elbel, "Mehrdimensionale Fouriertransformation für nichtäquidistante Daten," Master's thesis, Technical University Darmstadt, 1998.
14. J. Pelt, "Fast computation of trigonometric sums with application to frequency analysis of astronomical data," *Preprint, University of Delft* , 1998.
15. A. Ware, "Fast approximate Fourier transforms for irregularly spaced data," SIAM *Review* **40**, pp. 838 – 859, 1998.
16. B. Elbel and G. Steidl, "Fast Fourier transform for nonequispaced data," in *Approximation Theory IX*, C. K. Chui and L. L. Schumaker, eds., (Vanderbilt University Press, Nashville), 1998.
17. D. Potts, G. Steidl, and M. Tasche, "Fast Fourier transforms for nonequispaced data: A tutorial," in *Modern Sampling Theory: Mathematics and Application*, J. J. Benedetto and P. Ferreira, eds., ch. 12, pp. 253 – 274, Birkhäuser, 2000.
18. J. I. Jackson, "Selection of a convolution function for Fourier inversion using gridding," IEEE *Trans. Medical Imaging* **10**, pp. 473 – 478, 1991.
19. A. J. Jerry, "The Sannon sampling theorem – its various extensions and applications: a tutorial review," *Proc. IEEE* , 1965.
20. F. Natterer, "Fourier reconstruction in tomography," *Numer. Math.* **47**, pp. 343 – 353, 1985.
21. F. Natterer and F. Wübbeling, "Mathematical Methods in Image Reconstruction ," 2000.
22. P. Toft, *The Radon Transform - Theory and Implementation.* PhD thesis, Technical University of Denmark, 1996. `http://www.sslug.dk/~pto`.