$O(N^2 \log_2 N)$ Filtered Backprojection Reconstruction Algorithm for Tomography

Samit Basu, *Member, IEEE,* and Yoram Bresler, *Fellow, IEEE*

***Abstract*—We present a new fast reconstruction algorithm for parallel beam tomography. The new algorithm is an accelerated version of the standard filtered backprojection (FBP) reconstruction, and uses a hierarchical decomposition of the backprojection operation to reduce the computational cost from $O(N^3)$ to $O(N^2 \log_2 N)$. We discuss the choice of the various parameters that affect the behavior of the algorithm, and present numerical studies that demonstrate the cost versus distortion tradeoff. Comparisons with Fourier reconstruction algorithms and a recent multilevel inversion algorithm by Brandt *et al.*, both of which also have $O(N^2 \log_2 N)$ cost, suggest that the proposed hierarchical scheme has a superior cost versus distortion performance. It offers rms reconstruction errors comparable to the FBP with considerable speedup. For an example with a 512×512-pixel image and 1024 views, the speedup achieved with a particular implementation is over 40 fold, with reconstructions visually indistinguishable from the FBP.**

***Index Terms*—Fast algorithm, fast reconstruction, filtered backprojection, hierarchical, tomography.**

I. Introduction

THE reconstruction problem in two–dimensional (2-D) axial computed tomography (CT) is to recover an image from a set of its line-integral projections at different angles [1]. The type of algorithm used to address the reconstruction problem depends almost exclusively on how much of the line-integral information is available. When line-integrals are available from all possible directions, and the measurement noise is negligible, the filtered backprojection (FBP) reconstruction technique is the method of choice. Based on a discretization of an inverse formula for the Radon transform, the FBP consists of filtering each projection by a prespecified filter, followed by a backprojection operation. The filtering operation requires $O(N^2 \log N)$ operations when implemented by an FFT, or as little as $O(N^2)$ operations when implemented as a convolution with a fixed impulse response. In contrast, the backprojection operation, which computes for each pixel in the reconstructed image the sum of all line integrals that pass through that pixel, requires $O(N^3)$ operations, assuming a reconstruction of N^2 pixels, and projections at $O(N)$ angles. Backprojection, therefore, by far dominates the computational cost of FBP (or CBP) reconstruction.

Traditionally, for medical applications, backprojection has been accomplished by special hardware that exploits parallelism of the backprojection process to try and achieve near real-time reconstruction of a single slice (see, e.g., [2]). However, there still exists a lag in reconstruction time that is becoming increasingly important with the emergence of new technologies able to acquire data at ever faster rates. With the increasing data rates, the FBP becomes the bottleneck in the reconstruction and display process, and presents a barrier to real-time imaging.

Existing fast algorithms for reconstruction are based on either the Fourier slice theorem [3]–[8], or on a multiresolution resampling of the backprojection, as proposed in [9].[1] Algorithms based on the Fourier slice theorem use interpolations to transform the Fourier projection data from the polar to Cartesian grid, from which the reconstruction can be obtained by an inverse FFT. These fast algorithms, the most general of which appears to be gridding [6], [8], (which includes most other Fourier reconstruction algorithms or FRAs as special cases), generally have $O(N^2 \log N)$ complexity. Unfortunately, the interpolation step generally requires a large number of computations to avoid the introduction of artifacts into the reconstruction. Experimental evidence indicates that for reasonable image sizes $N \leq 10^3$, the realized performance gain of gridding over the more straightforward FBP is significantly less than the potential $N/\log N$ speedup. This also generally comes at a loss in reconstruction quality as well.

The multilevel inversion (MI) method described in [9] by Brandt *et al.* uses a sequence of nonuniformly sampled grids to efficiently represent the projections at different stages of the backprojection. The resulting algorithm has $O(N^2 \log N)$ complexity. The algorithm generates blurred reconstructions. A postprocessing step involving a deconvolution with a Gaussian approximation to the point spread function is necessary to counteract the blurring. Despite the theoretically $O(N^2 \log N)$ computational requirements of MI, we have found that our algorithm proposed in this paper consistently outperforms the MI method in terms of both reconstruction distortion and CPU time.

There are other fast reconstruction methods. These include the linogram method [10], and the "links" method [11], as well as related fast methods for reprojection, such as [12], [13], etc. Studying all of these algorithms is beyond the scope of this work, and for the purposes of comparison we will focus on the FRA, MI, and direct methods.

[1]Kindly provided to us by the authors.

Manuscript received April 28, 1999; revised April 18, 2000. This work supported in part by the National Science Foundation under Grant CCR 99-72980, a Fellowship from the Joint Services Electronics Program, and the Mac Van Valkenburg Memorial Fellowship. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. William Clem Karl.

S. Basu is with General Electric Corporate Research and Development Center, Niskayuna, NY 12309 USA (e-mail: basu@crd.ge.com).

Y. Bresler is with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: ybresler@uiuc.edu).

Publisher Item Identifier S 1057-7149(00)08053-2.

We present a class of fast tomographic reconstruction algorithms, which we call fast hierarchical backprojection (FHBP) algorithms, that address some of the shortcomings of existing fast algorithms. FHBP algorithms are based on a hierarchical decomposition of the Radon transform, similar to the one proposed in [14], and require $O(N^2 \log N)$ time for reconstruction. The algorithms do not suffer from the artifacts from which FRAs suffer (as there are no frequency-domain interpolations), and offer significant improvement in performance over the FRA and MI methods, with little or no loss in reconstruction quality compared to the direct FBP reconstruction. The potential speedup for images of size 10^3 square approaches and even exceeds the two orders of magnitude predicted, with small mean squared error (MSE). Thus, FBP-based reconstructions can be obtained in a small fraction of the time currently required. The proposed algorithms are also highly parallelizable, and can reuse existing hardware accelerations.

The basic idea behind the hierarchical decomposition is simple: the reconstruction of an $N \times N$ image can be written as the sum of shifted reconstructions of four smaller images, each of size $N/2 \times N/2$, centered at the origin. Because these images are half the (linear) size of the original image, they can be reconstructed from half the number of projections. Thus, if N projections were available for the reconstruction of the original image, then $N/2$ projections are need to reconstruct the smaller images. Each of the 4 reconstructions of the subimages will cost $(N/2)^2 N/2 = N^3/8$ operations, and 4 of these reconstructions must be performed for a total cost of $N^3/2$ operations to reconstruct the object, as opposed to N^3 operations for the original reconstruction. If we apply this decomposition recursively, then we get an $N^2 \log N$ cost, reducing the image size and projection count by a factor of two at each stage.

Although we describe the algorithm for the case $N = 2^K$, it can be easily be extended to other composite N. Also, we assume that $P = MN$ for some integer M. If this is not the case, then the projection data can be angularly interpolated prior to backprojection (such interpolation can be included as part of the filtering step in the FBP algorithm). Also, we describe the algorithm exclusively in terms of parallel beam projection data. It can be applied to fan beam data by appropriate rebinning.

We describe the hierarchical decomposition of the backprojection operation in Section II. In this section, we first construct an exact, but slow decomposition of the backprojection operator. We then show how angular decimation can be introduced to lead to an approximate decomposition which in turn leads to a fast hierarchical backprojection algorithm with $O(N^2 \log N)$ cost. Section III addresses the issues of angular and radial oversampling, and their effect on the fast algorithm. Section IV describes the two other $O(N^2 \log N)$ reconstruction algorithms to which we compare FHBP: the FRA and MI methods. In Section V, we present simulation results using the well established Shepp–Logan head phantom, and compare the proposed algorithm with other fast algorithms. Conclusions are presented in Section VI.

II. Hierarchical Decomposition of Backprojection

In this section, we first formally define the backprojection operation that forms the backbone of the FBP reconstruction algorithm. We then show that the backprojection operation admits an exact, hierarchical decomposition (with the same cost as the direct backprojection) in Section II-B. Then in Section II-C, we demonstrate how an approximate, faster decomposition can be constructed from this exact decomposition using the properties of backprojection. In Section II-D, we use both of these decompositions to construct the FHBP algorithm.

A. Direct Backprojection

We consider an $N \times N$ pixel reconstructed object, with pixel size normalized to one. We assume P view angles, uniformly spaced on $[0, 2\pi]$, although the algorithm can be easily generalized to other view angle distributions. In order to simplify the expressions in this section as much as possible, we will use an operator formulation. Operators between discrete spaces will be denoted using a bold Roman font; operators between discrete and continuous spaces, or between continuous and continuous spaces will be denoted using a script font. The Hilbert spaces that arise in the formulation are $L_2(\mathbb{R}^2)$, the space of square-integrable functions supported on $\mathbb{R}^2$, $L_2^P(\mathbb{R})$, the space of P-tuples of square-integrable functions on $\mathbb{R}$, $\ell_2(\mathbb{Z}^2)$, the space of square summable 2-D sequences, and $\ell_2^P(\mathbb{Z})$, the space of P-tuples of square summable sequences. All spaces are equipped with the standard, unweighted inner product. We will use a version of discrete backprojection that incorporates a modification that is required to construct the hierarchical decomposition. The *discrete backprojection* operator is a map $\boldsymbol{B}_{P,N}^{\boldsymbol{\tau}}$: $\ell_2^P(\mathbb{Z}) \rightarrow \ell_2(\mathbb{Z}^2)$ that produces a discrete image [element of $\ell_2(\mathbb{Z}^2)$] with $N \times N$ support from P discrete projections [each an element of $\ell_2(\mathbb{Z})$]. $\boldsymbol{B}_{P,N}^{\boldsymbol{\tau}}$ is affected by a further parameter $\boldsymbol{\tau} \in [-0.5, 0.5]^P$, a vector of radial shifts, whose purpose will be explained shortly. The dependence of $\boldsymbol{B}_{P,N}^{\boldsymbol{\tau}}$ on $\boldsymbol{\tau}$ will sometimes be suppressed for notational convenience.

To formulate the discrete backprojection operator, we decompose the backprojection process into four logical steps:

1) projections are radially interpolated to continuous functions;
2) continuous projections are backprojected onto the plane;
3) backprojected image is smoothed and sampled on a Cartesian grid;
4) samples are truncated to a region of interest.

Thus, the discrete backprojection operator is decomposed into a product of four simpler operators. Specifically,

$$\boldsymbol{B}_{P,N}^{\boldsymbol{\tau}} = \boldsymbol{K}_N \mathcal{S} \mathcal{B}_P \mathcal{I}_{\boldsymbol{\tau}} \tag{1}$$

where $\mathcal{I}_{\boldsymbol{\tau}}$: $\ell_2^P(\mathbb{Z}) \rightarrow L_2^P(\mathbb{R})$ is the radial interpolation operator of step 1, with shifting by $\boldsymbol{\tau}$

$$\mathcal{I}_{\boldsymbol{\tau}} g(r, p) = \sum_k g(k, p) \phi(r - (k + \tau_p)T) \tag{2}$$

where T is the radial sampling period, and ϕ is the convolutional interpolation kernel used. This operator interpolates each of the projections, shifting the pth projection by τ_p. For the case of simple linear interpolation, $\phi(x) = \Lambda(x/T)$, where

$$\Lambda(x) = \begin{cases} 1 - |x|, & |x| \leq 1 \\ 0, & \text{else.} \end{cases}$$

In the standard FBP reconstruction problem, we choose $\boldsymbol{\tau} = \mathbf{0}$.

The operator $\mathcal{B}_P$: $L_2^P(\mathbb{R}) \rightarrow L_2(\mathbb{R}^2)$ is the backprojection operator for radially unsampled sinograms (step 2), defined by

$$\mathcal{B}_P g_c(x, y) = \sum_{p=1}^{P} g_c(x\cos\theta_p + y\sin\theta_p, p) \tag{3}$$

where $\boldsymbol{\theta} \in [0, 2\pi)^P$ is the vector[2] of view angles. We will, for simplicity, assume that

$$\theta_p = \frac{2\pi p}{P} \tag{4}$$

but the algorithm we describe can be easily generalized to other view angle distributions.

The next operator $\mathcal{S}$: $L_2(\mathbb{R}^2) \rightarrow \ell_2(\mathbb{Z}^2)$ is the mollification, or generalized sampling operator of step 3 that represents the backprojected function on a discrete grid. We will assume that this operation is shift invariant, and that the coordinates have been scaled so that $\mathcal{S}$ can be written as

$$\mathcal{S} f_c(m, n) = \int\int f_c(x, y) b(x - m, y - n)\, dx\, dy. \tag{5}$$

The function $b(x, y)$ may be an indicator function for a pixel, or some smoother function. We will assume only that $b(x, y)$ is chosen to have small support. Suggestions for smoother choices of $b(x, y)$ include tensor splines, truncated Gaussians, and smooth, rotationally symmetric functions.

The final operator that makes up $\boldsymbol{B}$ is a truncation operator for step 4. The operator $\boldsymbol{K}_N$: $\ell_2(\mathbb{Z}^2) \rightarrow \ell_2(\mathbb{Z}^2)$ windows the argument 2-D sequence to a square of size $N \times N$, approximately centered at the origin (depending on whether N is odd or even)

$$\boldsymbol{K}_N f(m, n) = \begin{cases} f(m, n), & m, n \in \mathbb{Z}, -\dfrac{N}{2} \le m, n < \dfrac{N}{2} \\ 0, & \text{else.} \end{cases} \tag{6}$$

Of course, in implementation, the breakdown of the backprojection into 4 steps is impossible, since this would involve the manipulation of continuous quantities. However, (1)–(3), (5) and (6) can be combined into the following alternate formula for the discrete backprojection:

$$\boldsymbol{B}^{\boldsymbol{\tau}}_{P,N} g(m, n) = \begin{cases} \displaystyle\sum_p \sum_k g(k, p) \eta^{\boldsymbol{\tau}}_{k,p,m,n}, & -\dfrac{N}{2} \le m, n < \dfrac{N}{2} \\ 0, & \text{else} \end{cases} \tag{7}$$

where

$$\eta^{\boldsymbol{\tau}}_{k,p,m,n} = \int\int \phi(x\cos\theta_p + y\sin\theta_p - (k + \tau_p)T) \cdot b(x - m, y - n)\, dx\, dy. \tag{8}$$

B. Exact Hierarchical Decomposition of Backprojection

We now derive an exact hierarchical decomposition of the backprojection operation, in which the backprojection of P projections onto an $N \times N$ image is decomposed into four backprojections of P projections, each onto a $N/2 \times N/2$-sized image, centered at the origin.

To derive the exact hierarchical decomposition, we are interested in the interaction of $\boldsymbol{B}$ with several operations: 2-D shifts (translations) of the output, and truncations of its input and output. To examine this interaction in detail, we will present a series of simple properties, each of which explicitly describe the interaction of various shift and truncation operations with each of the operators defined above. Proofs of these properties are omitted, as they follow directly from the appropriate definitions. We first define two spatial shift operators. The first is a discrete shifter $\boldsymbol{Z}_{\boldsymbol{\delta}}$: $\ell_2(\mathbb{Z}^2) \rightarrow \ell_2(\mathbb{Z}^2)$, with $\boldsymbol{\delta} \in \mathbb{Z}^2$, defined by

$$\boldsymbol{Z}_{\boldsymbol{\delta}} f(m, n) = f(m - \delta_1, n - \delta_2). \tag{9}$$

The second operator is a continuous shifter $\mathcal{M}_{\boldsymbol{\delta}}$: $L_2(\mathbb{R}^2) \rightarrow L_2(\mathbb{R}^2)$, with $\boldsymbol{\delta} \in \mathbb{R}^2$, defined by

$$\mathcal{M}_{\boldsymbol{\delta}} f_c(x, y) = f_c(x - \delta_1, y - \delta_2). \tag{10}$$

We also define two sinogram-domain shift operators, one discrete and one continuous. The first $\hat{\boldsymbol{Z}}_{\boldsymbol{\delta}}$: $\ell_2^P(\mathbb{Z}) \rightarrow \ell_2^P(\mathbb{Z})$ for $\boldsymbol{\delta} \in \mathbb{Z}^P$, defined by

$$\hat{\boldsymbol{Z}}_{\boldsymbol{\delta}} g(k, p) = g(k - \delta_p, p). \tag{11}$$

The second is a continuous shifter $\hat{\mathcal{M}}_{\boldsymbol{\delta}}$: $L_2^P(\mathbb{R}) \rightarrow L_2^P(\mathbb{R})$, defined by

$$\hat{\mathcal{M}}_{\boldsymbol{\delta}} g_c(r, p) = g_c(r - \delta_p, p) \tag{12}$$

where $\boldsymbol{\delta} \in \mathbb{R}^P$.

Property 1: (Shift property of $\mathcal{B}_P$)

$$\mathcal{M}_{\boldsymbol{\delta}} \mathcal{B}_P = \mathcal{B}_P \hat{\mathcal{M}}_{\boldsymbol{\delta}\cdot\boldsymbol{\theta}} \tag{13}$$

where $\boldsymbol{\delta} \in \mathbb{R}^2$, and $\boldsymbol{\delta}\cdot\boldsymbol{\theta} \in \mathbb{R}^P$ is defined by $(\boldsymbol{\delta}\cdot\boldsymbol{\theta})_p = \delta_1 \cos\theta_p + \delta_2 \sin\theta_p$.

Property 2: (Shift property of $\mathcal{S}$)

$$\boldsymbol{Z}_{\boldsymbol{\delta}} \mathcal{S} = \mathcal{S} \mathcal{M}_{\boldsymbol{\delta}} \tag{14}$$

for $\boldsymbol{\delta} \in \mathbb{Z}^2$.

Property 3: (Shift property of $\mathcal{I}_{\boldsymbol{\tau}}$)

$$\hat{\mathcal{M}}_{\boldsymbol{\delta}} \mathcal{I}_{\boldsymbol{\tau}} = \mathcal{I}_{\langle \boldsymbol{\tau} + \boldsymbol{\delta}/T \rangle} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\tau} + \boldsymbol{\delta}/T]} \tag{15}$$

for $\boldsymbol{\delta} \in \mathbb{R}^2$, where for any vector $\boldsymbol{x} \in \mathbb{R}^P$, $[\boldsymbol{x}] \in \mathbb{Z}^P$, $\langle \boldsymbol{x} \rangle \in [-0.5, 0.5]^P$ where $[\boldsymbol{x}]_i$ is x_i rounded to the nearest integer, and $\langle \boldsymbol{x} \rangle_i = x_i - [\boldsymbol{x}]_i$.

The last property that we require relates to the fact that the discrete backprojection we defined in (1) is *local* is a certain sense. In particular, suppose that R_1 is the radius of support of b, and R_2 is the radius of support of ϕ. Then from geometric arguments and algebraic simplifications, it follows from (8) that $\eta^{\boldsymbol{\tau}}_{k,p,m,n} = 0$ for all k, p, m, n satisfying

$$|m\cos(\theta_p) + n\sin(\theta_p) - T(k + \tau_p)| \ge R_1 + R_2. \tag{16}$$

If m, n are restricted to a $N \times N$ region, centered at the origin, then we can replace (16) with the looser sufficient condition

$$|k| \ge \frac{R_1 + R_2 + N/\sqrt{2}}{T} + \frac{1}{2} \Longrightarrow \eta^{\boldsymbol{\tau}}_{k,p,m,n} = 0 \quad \forall\, 0 \le p < P, \quad -\frac{N}{2} \le m, n < \frac{N}{2}. \tag{17}$$

To derive (17) from (16), we have used the triangle inequality, the inequality $|a - b| \ge ||a| - |b||$, and the fact that $|\tau_p| \le 1/2$. Thus, it follows that the sum over k in (7), can be replaced by a sum over only those k that fail[3] to satisfy (17). To express this

[2]In our implementation, we assume that the symmetry properties (see, e.g., [15]) of the Radon transform are used so that $\boldsymbol{\theta} \in [0, \pi)^P$. Note that if θ_p are equally spaced on $[0, 2\pi)$ and P is even, then this symmetry can be used to reduce the complexity of the backprojection by a factor of 2.

[3]Note that (16) is equivalent to truncating the sinogram to the support of the square, while (17) truncates to the circumscribing circle. A slight gain in efficiency can be obtained by using (16), at the cost of increased complexity in the algorithm's implementation.

in terms of operators, we define $\hat{\boldsymbol{K}}_N$: $\ell_2^P(\mathbb{Z}) \rightarrow \ell_2^P(\mathbb{Z})$, which radially truncates its argument as follows:

$$\hat{\boldsymbol{K}}_N g(k, p) = \begin{cases} g(k, p), & |k| \leq \dfrac{R_1 + R_2 + N/\sqrt{2}}{T} + \dfrac{1}{2} \\ 0, & \text{else.} \end{cases} \tag{18}$$

We then have the following property.

Property 4: (Truncation of $\boldsymbol{B}_{P,N}$ input)

$$\boldsymbol{B}_{P,N} = \boldsymbol{B}_{P,N}\hat{\boldsymbol{K}}_N. \tag{19}$$

The hierarchical decomposition of the backprojection operator is the main result of this subsection, and we present it as the following theorem.

Theorem 1: Let $N = 2K$ for $K \in \mathbb{Z}$, $K \geq 1$. Then

$$\boldsymbol{B}^{\boldsymbol{\tau}}_{P,N} = \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{B}^{\langle \boldsymbol{\nu}_i \rangle}_{P,N/2} \hat{\boldsymbol{K}}_{N/2} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]} \tag{20}$$

where $\boldsymbol{\nu}_i = \boldsymbol{\tau} - \boldsymbol{\delta}_i \cdot \boldsymbol{\theta}/T$, and

$$\boldsymbol{\delta}_1 = (N/4,\ N/4)', \quad \boldsymbol{\delta}_2 = (-N/4,\ N/4)'$$
$$\boldsymbol{\delta}_3 = (N/4,\ -N/4)', \quad \boldsymbol{\delta}_4 = (-N/4,\ -N/4)'$$

for $N > 2$, and

$$\boldsymbol{\delta}_1 = (1, 0)', \quad \boldsymbol{\delta}_2 = (0, 0)'$$
$$\boldsymbol{\delta}_3 = (1, 1)' \quad \boldsymbol{\delta}_4 = (0, 1)'$$

for $N = 2$.

Proof: First, the following can trivially be shown

$$\boldsymbol{K}_N = \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{K}_{N/2} \boldsymbol{Z}_{-\boldsymbol{\delta}_i}. \tag{21}$$

Next, from (1) and the fact that $\boldsymbol{K}_N = \boldsymbol{K}_N \boldsymbol{K}_N$, it follows that

$$\boldsymbol{B}^{\boldsymbol{\tau}}_{P,N} = \boldsymbol{K}_N \boldsymbol{B}^{\boldsymbol{\tau}}_{P,N}.$$

Hence

$$\begin{aligned} \boldsymbol{B}^{\boldsymbol{\tau}}_{P,N} &= \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{K}_{N/2} \boldsymbol{Z}_{-\boldsymbol{\delta}_i} \boldsymbol{B}^{\boldsymbol{\tau}}_{P,N} \\ &= \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{K}_{N/2} \boldsymbol{Z}_{-\boldsymbol{\delta}_i} \mathcal{S}\mathcal{B}_P \mathcal{I}_{\boldsymbol{\tau}} \\ &= \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{K}_{N/2} \mathcal{S}\mathcal{M}_{-\boldsymbol{\delta}_i} \mathcal{B}_P \mathcal{I}_{\boldsymbol{\tau}} \\ &= \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{K}_{N/2} \mathcal{S}\mathcal{B}_P \hat{\mathcal{M}}_{-\boldsymbol{\delta}\cdot\boldsymbol{\theta}} \mathcal{I}_{\boldsymbol{\tau}} \\ &= \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{K}_{N/2} \mathcal{S}\mathcal{B}_P \mathcal{I}_{\langle \boldsymbol{\nu}_i \rangle} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]} \\ &= \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{B}^{\langle \boldsymbol{\nu}_i \rangle}_{P,N/2} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]} \\ &= \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{B}^{\langle \boldsymbol{\nu}_i \rangle}_{P,N/2} \hat{\boldsymbol{K}}_{N/2} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]}. \end{aligned}$$

■

The basic idea is to note that the decomposition of Theorem 1 consists of terms of the form $\boldsymbol{B}^{\langle \boldsymbol{\nu}_i \rangle}_{P,N/2} \hat{\boldsymbol{K}}_{N/2} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]}$, which is essentially the reconstruction of an $N/2 \times N/2$ quadrant of the original image centered at the origin, using the same number of projections, P. The decomposition consists of the following steps:

1) shift the projections so that they correspond to the ith quadrant of the reconstruction being centered at the origin, and then radially truncate them to the width used by the backprojection $\boldsymbol{B}^{\langle \boldsymbol{\nu}_i \rangle}_{P,N/2}$ (cf. Property 4);
2) backproject the shifted projections, and truncate the result to a $N/2 \times N/2$ region;
3) shift the reconstructed quadrant back to the appropriate location, and add quadrants together.

The resulting decomposition is illustrated in Fig. 1.

Theorem 1 does *not* lead to a fast algorithm. To see why, let us assume that computing $\boldsymbol{B}_{P,N} g$ requires $c_B N^2 P$ operations. Then Theorem 1 replaces $\boldsymbol{B}_{P,N}$ by four terms of the form $\boldsymbol{B}_{P,N/2}$. Ignoring the shifting operations (which can be affected by reindexing), the cost of each of the four quadrant backprojections is $c_B N^2 P/4$, for a total of $c_B N^2/P$ operations. Thus, computing the backprojection using Theorem 1 does not lead to a fast algorithm.

C. Angularly Decimated Hierarchical Backprojection

In this subsection, we derive a different, approximate decomposition starting from Theorem 1, which does lead to an $O(N^2 \log N)$ algorithm for backprojection. Instead of reconstructing the smaller $N/2 \times N/2$ subimages using all P projections, as prescribed in Theorem 1, we will angularly decimate the projections from P to $P/2$ projections before backprojecting, so that each term in the sum involves a backprojection to an $N/2 \times N/2$ size object using $P/2$ projections. In this section, we develop this basic idea into the fast algorithm, as well as argue why such a decimation step is reasonable.

At this point, we introduce our first key assumption

A1) Radial interpolation $\mathcal{I}$ is an ideal bandlimited interpolation with bandlimit B.

This assumption is consistent with a bandlimited ramp filter used in the reconstruction algorithm to eliminate aliasing due to angular sampling of the projections, and to limit the effects of noise [15], [16]. Hence, the continuous filtered projections g_c can be assumed to be bandlimited to frequency B.

Next, we define an angular decimation operator $\mathcal{O}$: $L_2^P(\mathbb{R}) \rightarrow L_2^{P/2}(\mathbb{R})$ for functions $g_c \in L_2^P(\mathbb{R})$, which angularly decimates g_c by a factor of two. The operator can be written as

$$\mathcal{O}_P g_c(r, q) = \sum_{p=0}^{P-1} g_c(r, p) \psi(2q - p) \tag{22}$$

where ψ is the angular smoothing kernel, applied to avoid aliasing prior to downsampling.

Now, in Theorem 1, we wish to replace the terms $\boldsymbol{B}_{P,N/2}$ with terms of the form $\boldsymbol{B}_{P/2,N/2}$ with some suitable preprocessing. Because the radial interpolation (by Assumption A1) results in sinograms g_c bandlimited to radial bandwidth B, the equivalent problem is to replace the term

$$\boldsymbol{K}_{N/2} \mathcal{S}\mathcal{B}_P g_c \tag{23}$$

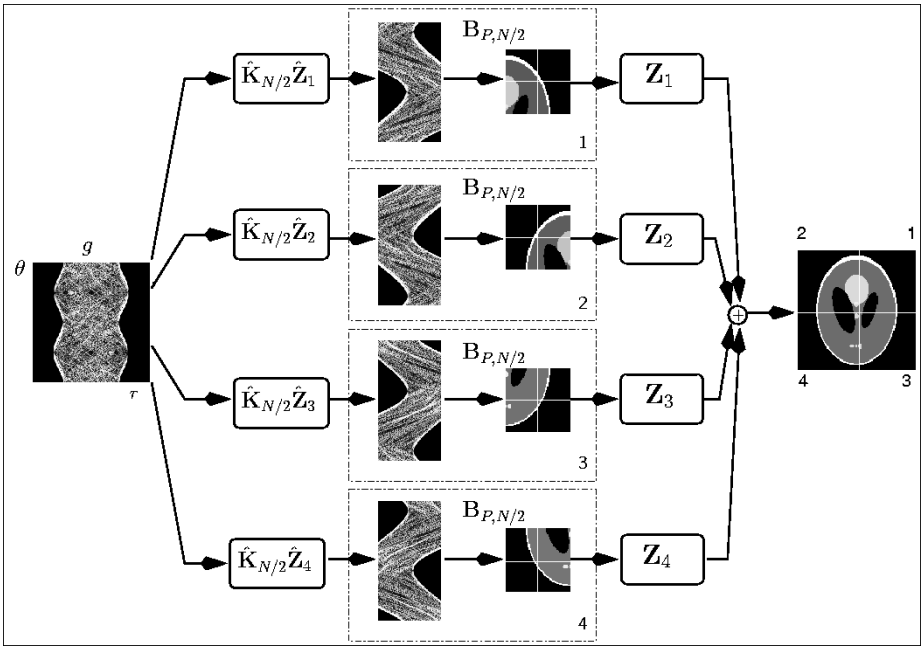


Fig. 1. Decomposition of Theorem 1 applied to a filtered sinogram. Here, $\hat{\boldsymbol{Z}}_i = \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]}$, and $\boldsymbol{Z}_i = \boldsymbol{Z}_{\boldsymbol{\delta}_i}$. Each of the smaller sinograms is computed from g by radially shifting each projection by a different (integer) amount, and then truncating to a width of $O(N\sqrt{2})$. The decomposition can be reapplied to the boxed sections.

which is the backprojection of P (radially continuous) projections onto a $N/2 \times N/2$ pixel grid, by the following, which first angularly decimates the data prior to backprojecting:

$$\boldsymbol{K}_{N/2}\mathcal{S}\mathcal{B}_{P/2}\mathcal{O}_P g_c \tag{24}$$

where g_c is radially bandlimited, with bandlimit B.

We will show that if $P > BN\sqrt{2}$, then for an appropriately chosen kernel ψ in (22), and for any sinogram g_c radially bandlimited to bandwidth B

$$\boldsymbol{K}_{N/2}\mathcal{S}\mathcal{B}_P \approx \boldsymbol{K}_{N/2}\mathcal{S}\mathcal{B}_{P/2}\mathcal{O}_P \tag{25}$$

where the approximation is in the L_2 sense, i.e., the difference has negligible energy. Thus, if the projections are radially bandlimited with bandwidth B, and P, N, B satisfy $P > BN\sqrt{2}$, then the backprojection of P projections onto an image of size $N/2 \times N/2$ can be well approximated by the following sequence of steps:

1) first angularly decimate the projection data via $\mathcal{O}_P$;
2) then backproject it at half the number of view angles.

To verify this claim, consider the L_2 approximation error

$$\begin{aligned}\varepsilon &\triangleq \|(\boldsymbol{K}_{N/2}\mathcal{S}\mathcal{B}_P - \boldsymbol{K}_{N/2}\mathcal{S}\mathcal{B}_{P/2}\mathcal{O}_P)g_c\|_2 \\ &= \|(\boldsymbol{K}_{N/2}\mathcal{S}\mathcal{B}_P - \boldsymbol{K}_{N/2}\mathcal{S}\mathcal{B}_{P/2}\mathcal{O}_P)\Pi_B g_c\|_2 \\ &\le \|\boldsymbol{K}_{N/2}\mathcal{S}(\mathcal{B}_P - \mathcal{B}_{P/2}\mathcal{O}_P)\Pi_B\|_2 \|g_c\|_2 \end{aligned} \tag{26}$$

where Π_B: $L_2^P(\mathbb{R}) \to L_2^P(\mathbb{R})$ is an ideal radial lowpass filter with cutoff B, and equality holds because g_c is bandlimited to the same bandwidth B. The first norm in the right hand side of the inequality in (26) is the (induced) operator norm. We would now like to rewrite ε in terms of errors in the projection domain. To do so, we need to consider the adjoints[4] of the various operators. Recall that for a bounded linear operator $\mathcal{X}$: $S_1 \to S_2$, the adjoint $\mathcal{X}^\star$: $S_2 \to S_1$ is the operator which satisfies

$$\langle \mathcal{X}g,\, h\rangle_{S_2} = \langle g,\, \mathcal{X}^\star h\rangle_{S_1} \tag{27}$$

for all $g \in S_1$, $h \in S_2$, where $\langle \cdot, \cdot \rangle_{S_i}$ denotes the inner product on the space S_i.

It is readily verified that the adjoint of $\mathcal{O}_P$ is an angular interpolation operator $\mathcal{O}_P^\star$: $L_2^{P/2}(\mathbb{R}) \to L_2^P(\mathbb{R})$, defined by

$$\begin{aligned}\mathcal{O}_P^\star g_c(r,\, q) = &\sum_{p=0}^{P/2-1} g_c(r,\, p)\psi(2p-q) \\ &q \in \{0,\, \cdots,\, P-1\}\end{aligned} \tag{28}$$

assuming ψ is real-valued. Equation (28) can be derived as follows:

$$\langle h,\, \mathcal{O}_P g\rangle_{L_2^P(\mathbb{R})} = \int_r \sum_q h(r,\, q) \sum_p g(r,\, p)\psi(2q-p)\, dr \tag{29}$$

$$= \int_r \sum_p g(r,\, p) \sum_q h(r,\, q)\psi(2q-p)\, dr \tag{30}$$

$$= \int_r \sum_p g(r,\, p) \sum_q h(r,\, q)\psi(2q-p)\, dr \tag{31}$$

$$= \langle \mathcal{O}_P^\star h,\, g\rangle_{L_2^{P/2}(\mathbb{R})}. \tag{32}$$

[4]The reader who is unfamiliar with adjoints may wish to peruse [17], [18], both of which discuss properties of adjoints and their use in applications. However, adjoints in an infinite dimensional setting share many of their important properties with the matrix case, where the adjoint of a matrix is simply its transpose. What we will thus argue is that if $\boldsymbol{M}$ is a matrix representation of the Radon transform, and $\bar{\boldsymbol{M}}$ is an approximation to $\boldsymbol{M}$ in the sense that $\|M - \bar{M}\| \le \varepsilon$, then it is also true that $\|\boldsymbol{M}' - \bar{\boldsymbol{M}}'\| \le \varepsilon$. Furthermore, a classic result in tomography is that M' is a discrete backprojection operation.

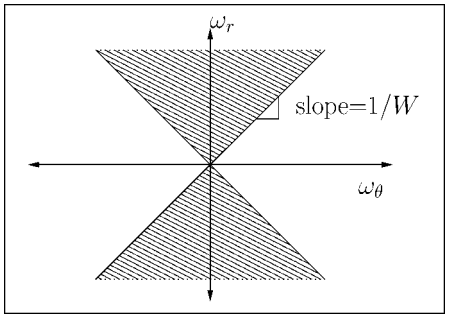


Fig. 2. Spectral support of the Radon transform of an object with radius of support W.

We also note that $\boldsymbol{K}_N^\star = \boldsymbol{K}_N$, $\mathcal{B}_P^\star = \mathcal{R}_P$, where $\mathcal{R}_P$: $L_2(\mathbb{R}^2) \rightarrow L_2^P(\mathbb{R})$ is the standard, discrete-angle Radon transform operator, defined by

$$\mathcal{R}_P f_c(r, p) = \int_{\mathbb{R}} f_c(r\cos\theta_p - s\sin\theta_p,\, r\sin\theta_p + s\cos\theta_p)\, ds,$$

and $\mathcal{S}^\star$: $\ell_2(\mathbb{Z}^2) \rightarrow L_2(\mathbb{R}^2)$ is an interpolation defined by

$$\mathcal{S}^\star f(x, y) = \sum_{m,n} f(m, n) b(x - m, y - n).$$

Next, we use the fact that $\|\mathcal{C}\|_2 = \|\mathcal{C}^\star\|_2$ for any linear, bounded operator $\mathcal{C}$ with adjoint $\mathcal{C}^\star$, along with properties of adjoints (and $\Pi_B^\star = \Pi_B$) to get

$$\begin{aligned}\varepsilon &= \|\boldsymbol{K}_{N/2}\mathcal{S}(\mathcal{B}_P - \mathcal{B}_{P/2}\mathcal{O}_P)\Pi_B\|_2 \\ &= \|\Pi_B(\mathcal{R}_P - \mathcal{O}_P^\star\mathcal{R}_{P/2})\mathcal{S}^\star\boldsymbol{K}_{N/2}\|_2 \\ &= \|\Pi_B\mathcal{R}_P\mathcal{S}^\star\boldsymbol{K}_{N/2} - \mathcal{O}_P^\star\Pi_B\mathcal{R}_{P/2}\mathcal{S}^\star\boldsymbol{K}_{N/2}\|_2 \quad (33)\end{aligned}$$

where the last equality follows because radial and angular filtering commute. Hence, the rms error ε in replacing (23) by (24) is the same as the rms error in replacing $\Pi_B\mathcal{R}_P\mathcal{S}^\star\boldsymbol{K}_{N/2}$ by $\mathcal{O}_P^\star\Pi_B\mathcal{R}_{P/2}\mathcal{S}^\star\boldsymbol{K}_{N/2}$. The advantage of this reformulation is that we can use known results about the spectrum of the projections to analyze the effect of angular sampling and interpolation.

To continue, let us first recall the *bow-tie* result of [19]. This result states that for an object f_c with radius of support W, the spectral support of the 2-D Fourier transform of its continuous-angle Radon transform is essentially supported on a "bow-tie" shaped region depicted in Fig. 2. From this result, we can infer that the Fourier transform of $\Pi_B\mathcal{R}_P f_c(r, p)$ (continuous with respect to r, and discrete with respect to p), has spectral support shown in Fig. 3. To avoid angular aliasing of the copies, the cut-off bandwidth B must be chosen so that $P > 2BW$. For f supported on an $N \times N$ grid centered at the origin, this is equivalent to

$$P > BN\sqrt{2}. \quad (34)$$

Now, assuming that P, B, and N satisfy (34) let us consider the spectral support of $\Pi_B\mathcal{R}_P\mathcal{S}^\star\boldsymbol{K}_{N/2}f$ for any $f \in \ell_2(\mathbb{Z}^2)$, shown in Fig. 4. The sinogram is angularly oversampled by a factor of two. It follows from Shannon's sampling theorem that the sinogram can be angularly downsampled by a factor of two with no loss of information. The sinogram can then be angularly upsampled by a factor of two, and lowpass filtered with no error

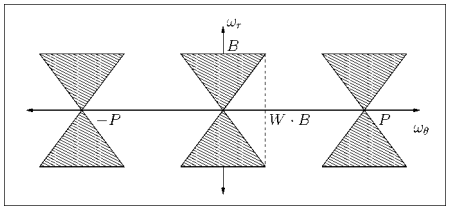


Fig. 3. Spectral support of the Radon transform of an object with radius of support W, sampled at P view angles, and radially bandlimited with bandlimit B.

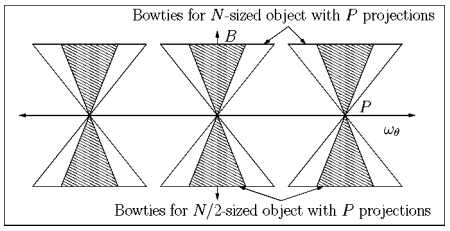


Fig. 4. Spectral support of the Radon transform of an object of size $N/2$, sampled at P view angles, and radially bandlimited with bandlimit B.

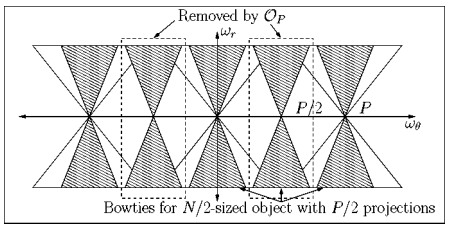


Fig. 5. Spectral support of the Radon transform of an object of size $N/2$, sampled at $P/2$ view angles, and radially bandlimited with bandlimit B. The copies that are boxed are removed by the ideal resampling operation.

(if ideal filtering is used). To that end, let $\mathcal{Y}$: $L_2^P(\mathbb{R}) \rightarrow L_2^{P/2}(\mathbb{R})$ denote an angular downsampler, i.e.,

$$\mathcal{Y}g_c(r, p) = g_c(r, 2p). \quad (35)$$

Then from the above arguments, it follows that for the appropriate choice of an angular filtering operator $\mathcal{H}$: $L_2^P(\mathbb{R}) \rightarrow L_2^P(\mathbb{R})$ defined by

$$\mathcal{H}g_c(r, p) = \sum_q g_c(r, q)h(p - q) \quad (36)$$

we have

$$\Pi_B\mathcal{R}_P\mathcal{S}^\star\boldsymbol{K}_{N/2} = \mathcal{H}\mathcal{Y}^\star\mathcal{Y}\Pi_B\mathcal{R}_P\mathcal{S}^\star\boldsymbol{K}_{N/2} \quad (37)$$

where $\mathcal{H}\mathcal{Y}^\star\mathcal{Y}$ corresponds to the following steps:

1) angular downsampling by a factor of two;
2) angular upsampling by a factor of two;
3) angular filtering with kernel $h(k) = \psi(k)$.

Fig. 5 illustrates this process in the spectral domain. Note that (34) ensures that no aliasing occurs after the downsampling step. Also note that (37) relies on ψ (or equivalently h) being an "ideal" bandlimited resampling kernel (i.e., a Dirichlet kernel for uniform angle distribution). Thus, we make our second assumption, which is

A2) $\mathcal{O}_P^\star$ is an ideal bandlimited angular interpolator.

Now, we note that $\mathcal{H}\mathcal{Y}^\star = \mathcal{O}_P^\star$ if $h(k) = \psi(-k)$, and that

$$\mathcal{Y}\Pi_B\mathcal{R}_P = \Pi_B\mathcal{Y}\mathcal{R}_P = \Pi_B\mathcal{R}_{P/2}$$

(because angular and radial filtering commute) so that (37) becomes

$$\Pi_B \mathcal{R}_P \mathcal{S}^\star \boldsymbol{K}_{N/2} = \mathcal{O}_P^\star \Pi_B \mathcal{R}_{P/2} \mathcal{S}^\star \boldsymbol{K}_{N/2}. \tag{38}$$

Thus, it follows from (38), (33), and (26) that $\varepsilon = 0$, establishing (25) as an equality (not an approximation).

Hence, we see that under ideal circumstances (i.e., when the bow-tie is the precise support of the spectrum, and Assumptions A1 and A2 hold), we have $\varepsilon = 0$. If we take into account the fact that Fig. 2 is only an approximation, then we have ε nonzero. But the bow-tie approximation can be made arbitrarily close (see [15] for rigorous estimates) by either increasing W, or increasing P for a fixed B, N. This latter approach amounts to increasing the amount of angular oversampling, and results in the bow-tie copies being spaced further apart.

Applying approximation (25) to the backprojections in Theorem 1, and invoking A1, it follows that

$$\boldsymbol{B}_{P,N}^{\boldsymbol{\tau}} \approx \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{K}_{N/2} \mathcal{S} \mathcal{B}_{P/2} \mathcal{O}_P \mathcal{I}_{\langle \boldsymbol{\nu}_i \rangle} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]}. \tag{39}$$

The next step in the derivation involves the effective interchange of $\mathcal{O}_P$ and $\mathcal{I}_{\langle \boldsymbol{\nu}_i \rangle}$. These two operators do not commute, because $\mathcal{I}_{\langle \boldsymbol{\nu}_i \rangle}$ involves a different shift $\nu_i(p)$ for $p = 0, \cdots, P-1$ for each projection. We therefore take a different approach.

Let us define a downsampler $\cdot \downarrow 2$: $\mathbb{R}^P \rightarrow \mathbb{R}^{P/2}$ as $(\boldsymbol{x} \downarrow 2)_p = x_{2p}$. In view of A1, and the fact that angular decimation does not affect radial bandwidth

$$\mathcal{O}_P \mathcal{I}_{\langle \boldsymbol{\nu}_i \rangle} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]} g(r, k)$$

is bandlimited in r to bandlimit B for any sequence $g \in \ell_2^P(\mathbb{Z})$. This function of r (for fixed k), can therefore be exactly interpolated from its radial samples at rate $1/T > 2B$ using bandlimited interpolation. It follows that for each $i = 1, \cdots, 4$

$$\begin{aligned} &\mathcal{O}_P \mathcal{I}_{\langle \boldsymbol{\nu}_i \rangle} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]} g(r, p) \\ &\quad = \mathcal{I}_{\langle \boldsymbol{\nu}_i \downarrow 2 \rangle} \mathcal{G}_{\langle \boldsymbol{\nu}_i \downarrow 2 \rangle} \mathcal{O}_P \mathcal{I}_{\langle \boldsymbol{\nu}_i \rangle} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]} g(r, p) \end{aligned} \tag{40}$$

where $\mathcal{G}_{\boldsymbol{\tau}}$: $L_2^{P/2}(\mathbb{R}) \rightarrow \ell_2^{P/2}(\mathbb{Z})$ with $\boldsymbol{\tau} \in [-0.5, 0.5]^{P/2}$ is a generalized resampling operator, defined by

$$\mathcal{G}_{\boldsymbol{\tau}} g_c(k, p) = g_c((k + \tau_p)T, p). \tag{41}$$

Thus, it follows that (39) can be rewritten as

$$\boldsymbol{B}_{P,N}^{\boldsymbol{\tau}} \approx \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{K}_{N/2} \mathcal{S} \mathcal{B}_{P/2} \mathcal{I}_{\langle \boldsymbol{\nu}_i \downarrow 2 \rangle} \mathcal{G}_{\langle \boldsymbol{\nu}_i \downarrow 2 \rangle} \mathcal{O}_P \mathcal{I}_{\langle \boldsymbol{\nu}_i \rangle} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]}. \tag{42}$$

The operation $\boldsymbol{A}_{P,i}$: $\ell^{P_2(\mathbb{Z})} \rightarrow \ell_2^{P/2}(\mathbb{Z})$, defined by $\boldsymbol{A}_{P,i} = \mathcal{G}_{\langle \boldsymbol{\nu}_i \downarrow 2 \rangle} \mathcal{O}_P \mathcal{I}_{\langle \boldsymbol{\nu}_i \rangle}$ is a discrete to discrete mapping that incorporates the radial interpolation, angular smoothing and downsampling, and radial sampling, all in one step. It can be represented explicitly as

$$\begin{aligned} &\boldsymbol{A}_{P,i} g(m, n) \\ &\quad = \sum_p \sum_k g(k, p) \phi(T(m - k + \nu_{2n} - \nu_p)) \cdot \psi(2n - p). \end{aligned} \tag{43}$$

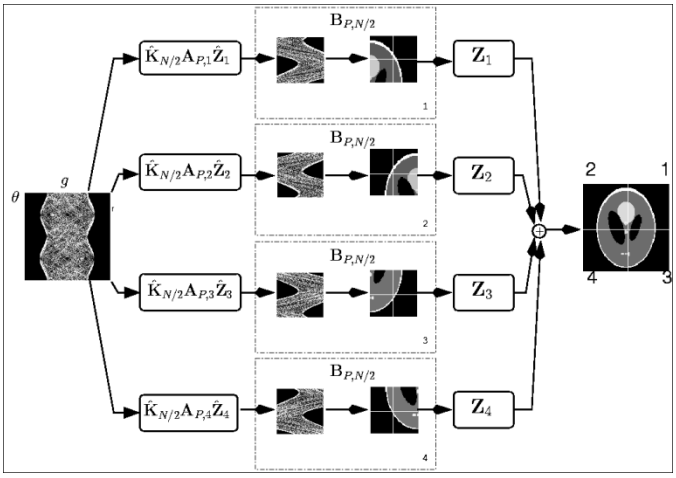


Fig. 6. Decomposition of (45) applied to a filtered sinogram, for comparison with Fig. 1. Here, $\hat{\boldsymbol{Z}}_i = \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]}$, and $\boldsymbol{Z}_i = \boldsymbol{Z}_{\boldsymbol{\delta}_i}$. Each of the smaller sinograms is computed from g by radially shifting each projection by a different (integer) amount, angularly decimating them by a factor of two, and then truncating to a width of $O(N\sqrt{2})$. The decomposition can be reapplied to the boxed steps.

For each point (m, n), the summation extends over the values of p and k for which the summand does not vanish. Using this definition in (42), we have the following form for the approximate decomposition of the backprojection operator

$$\begin{aligned} \boldsymbol{B}_{P,N}^{\boldsymbol{\tau}} &\approx \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{K}_{N/2} \mathcal{S} \mathcal{B}_{P/2} \mathcal{I}_{\langle \boldsymbol{\nu}_i \rangle \downarrow 2} \boldsymbol{A}_{P,i} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]} \\ &= \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{B}_{P/2,N/2}^{\langle \boldsymbol{\nu}_i \rangle \downarrow 2} \boldsymbol{A}_{P,i} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]}. \end{aligned} \tag{44}$$

Using Property 4 to reincorporate the radial truncation $\hat{\boldsymbol{K}}_{N/2}$, the final decomposition of $\boldsymbol{B}_{P,N}$ is

$$\boldsymbol{B}_{P,N}^{\boldsymbol{\tau}} \approx \sum_{i=1}^{4} \boldsymbol{Z}_{\boldsymbol{\delta}_i} \boldsymbol{B}_{P/2,N/2}^{\langle \boldsymbol{\nu}_i \rangle \downarrow 2} \hat{\boldsymbol{K}}_{N/2} \boldsymbol{A}_{P,i} \hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]}. \tag{45}$$

The decomposition of (45) is illustrated in Fig. 6 for comparison with Fig. 1. Note that the subsinograms depicted in Fig. 1 are each $P \times O(N/2)$ in size, and the backprojections onto the quadrants each involve P projections. In Fig. 6, on the other hand, the subsinograms have been angularly decimated to size $P/2 \times O(N/2)$, and the backprojections onto the quadrants each involve $P/2$ projections. In the next subsection, we will exploit this difference to construct the fast algorithm.

D. Fast Hierarchical Backprojection

Equation (45) is the source of the $O(N^2 \log N)$ fast algorithm. To see why, suppose that we can compute $\hat{\boldsymbol{K}}_{N/2} \boldsymbol{A}_{P,i}$ in $c_A PN/(2\sqrt{2})$ operations. The operation count on the right hand side is dominated by the cost of the backprojections $\boldsymbol{B}_{P/2,N/2}$, each of which requires $c_B PN^2/8$ operations. All four backprojections require a total of $c_B PN^2/2$ operations, as opposed to the left hand side of (45), which requires $c_B PN^2$ operations.

Although this improvement seems marginal, the true advantage of (45) is that it can be applied *recursively*, replacing each of the backprojections of $P/2$ projections onto an $N/2 \times N/2$ image ($\boldsymbol{B}_{P/2,N/2}$) by four backprojections of $P/4$ projections onto an $N/4 \times N/4$ image ($\boldsymbol{B}_{P/4,N/4}$), and so on. Ultimately, if

we continue to the single pixel level, the backprojection $\boldsymbol{B}_{P,N}$ will be decomposed into the sum of N^2 backprojections, each of P/N projections onto a single pixel. The dominant cost is now the computation of $\hat{\boldsymbol{K}}_{N_k}\boldsymbol{A}_{P_k,i}$ at each stage k, and not the computation of the backprojections [we assume that $\boldsymbol{B}_{P/N,1}g$ can be computed with $O(1)$ operations].

To find the overall operation count, then, let us examine the first stage of the decomposition, from which we can infer the total operation count. We must compute $h(m, n) = \hat{\boldsymbol{K}}_{N/2}\boldsymbol{A}_{P,i}g(m, n)$ for some g. However, $h(m, n)$ is nonzero for only $PN/(2\sqrt{2})$ values of (m, n), since $\boldsymbol{A}_{P,i}$ angularly decimates its input to $P/2$ projections, and $\hat{\boldsymbol{K}}_{N/2}$ radially truncates its argument to width $N\sqrt{2}/2$. Examining (43), we note that if ϕ and ψ have small support, then the cost of computing $h(m, n)$ for one pair of (m, n) is c_A, and the cost for all values of m, n is $c_APN/(2\sqrt{2})$. A total of four such h_i have to be computed, so the total cost is $c_APN\sqrt{2}$. Now, consider the kth level in the decomposition. There are now 4^k images to be reconstructed, each of size $N/2^k \times N/2^k$. We must compute $\hat{\boldsymbol{K}}_{N/2^k}\boldsymbol{A}_{P/2^{k-1},i}$ for each of these images, which requires $c_A\sqrt{2}PN/4^k$ operations for each sinogram. Thus, the total work at the kth stage is simply $c_APN\sqrt{2}$. Hence, the number of operations per stage is constant, and the number of stages is $\log_2 N$, leading to a total operation count of $O(PN\log_2 N)$. Assuming that $P = O(N)$, we get a total operation count of $O(N^2\log_2 N)$. Including the shifts $\hat{\boldsymbol{Z}}_{-[\boldsymbol{\nu}_i]}$ into the computations does not change the order of the operation count, as they also require $O(PN)$ operations. What remains, is to examine the possibility of choosing ϕ and ψ of small support, as required for the validity of the foregoing argument. Clearly, to satisfy A1 and A2 requires ϕ of infinite support, and ψ of support $P/2^k$ (the number of projections in the kth stage). In practice, however, accurate interpolations can be obtained with short interpolators if radial and angular oversampling is used. That is, $1/T = 2E_TB$ and $P = E_\theta BN\sqrt{2}$, with $E_T, E_\theta > 1$. With such oversampling, it is possible to choose ϕ and ψ so that the interpolation error decays exponentially with the length of ϕ and ψ [20]. We will therefore use such oversampling with short interpolators, in which case A1 and A2 become approximations, rather than assumptions. The approximation error in (25), however, can be controlled (made arbitrarily small) by the appropriate choice of oversampling and interpolator length.

III. Oversampling

As mentioned in the previous section, with short interpolators, the validity of approximations A1) and A2), and ultimately the key approximation (25) of the fast algorithm, depends on the amount of radial and angular oversampling involved. In practice, radial and angular oversampling are rarely part of the acquisition. In order to avoid excessive artifacts from the fast reconstruction process, we will discuss means of artificially increasing the sampling rate prior to reconstruction. In this section, we will examine how the amount of radial and angular oversampling can be controlled to tradeoff complexity for accuracy of the reconstruction.

A. Angular Oversampling

It may at first seem impossible to control the amount of angular oversampling, given that g is measured by some physical system. However, we note that there are two recursive decompositions to work with. The first is that of Theorem 1, which *exactly* decomposes a backprojection of P projections onto a $N \times N$ grid into four backprojections of P projections onto a $N/2 \times N/2$ grid. As we argued previously, (see Fig. 4, in particular), each of these smaller backprojections is thus effectively angularly oversampled by twice the oversampling factor of the original backprojection. We can apply Theorem 1 a total of $Q < \log N$ times, resulting in backprojections that are oversampled by 2^Q times the amount of angular oversampling of the original backprojection. When the backprojections are sufficiently oversampled, we can switch over to using (45), so that subsequent decompositions are angularly decimated. For all further decompositions, the backprojections will be oversampled by the same factor.

As Q is increased, the bow-tie approximations become more accurate, as increasing Q is equivalent to fixing N and B at each stage in the decomposition, but increasing P by a factor of 2^Q. Thus, we expect the error ε will decrease, and the total distortion in using the fast algorithm will similarly decrease. However, there is clearly a performance tradeoff in taking this approach, because we now have effectively replaced $\boldsymbol{A}_{P,i}$ by $\boldsymbol{A}_{P2^Q,i}$ throughout the decomposition, which implies that the total cost will increase in the $\log_2 N - Q$ fast stages by the factor 2^Q. In fact, the operation count can easily be shown to be

$$O(2^QPN(\log_2 N - Q + 1))$$

which reduces to $PN\log_2 N$ for $Q = 0$, and to PN^2 for $Q = \log_2 N$. Furthermore, we see that for Q fixed, the algorithm is still $N^2\log_2 N$ provided N is large.

Remark: An alternative to fixing the interpolation kernel and changing the amount of oversampling is to fix the amount of oversampling, and then change the interpolation kernel to achieve a desired accuracy/cost tradeoff. Also, different orderings of the exact and approximate decomposition steps will lead to different algorithms, all theoretically equivalent, but with slightly different properties than the one examined in detail in this work.

B. Radial Oversampling

Unlike the case of angular oversampling, the decomposition (45) does not incorporate any radial decimation, and hence, no radial oversampling tradeoff can be gained by using Theorem 1 instead of (45) for a certain number of iterations. Thus, in the case of radial oversampling, we must "synthesize" more radial data from the measurements g in such a way as to improve the approximation (40), while not introducing any new artifacts or anomalies into the reconstruction. The simplest way to synthesize this new data is to note that by Assumption A1, the operator $\mathcal{I}_{\boldsymbol{\tau}}$ exactly interpolates g for all radial coordinates. Hence, we can generate additional data by resampling the continuous functions $\mathcal{I}_{\boldsymbol{\tau}}g(r, p)$ at a higher rate, and changing the interpolation kernel for $\mathcal{I}_{\boldsymbol{\tau}}$ in subsequent levels of the decomposition. This process is similar to that of an oversampling D/A converter,

TABLE I
PARAMETERS USED IN IMPLEMENTATION OF GRIDDING

Parameter	Meaning	Value Used
$\hat{w}$	Gridding window	Kaiser-Bessel window
β_{rad}	Padding Factor for projection FFTs	2
α	Controls shape of Kaiser-Bessel window	1.25
K	Size of support of 2D interpolation window ($K \times K$)	variable

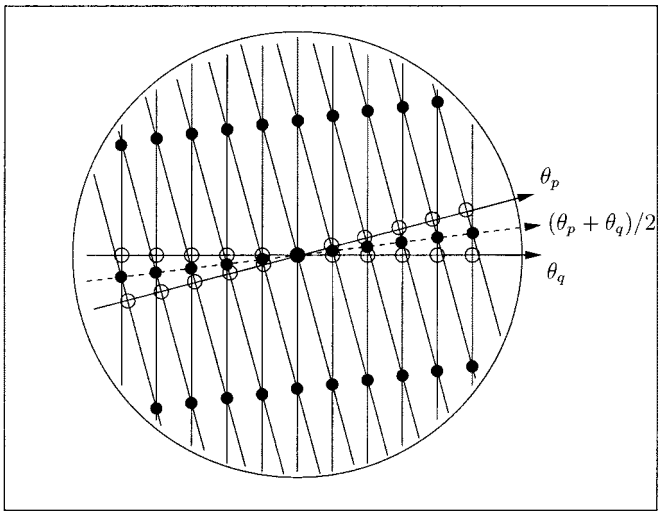


Fig. 7. First stage grid merge in Multilevel Inversion. Shown are two backprojected projections (one at direction θ_p and the other at direction θ_q). Because the backprojected functions are constant in the orthogonal direction, they can be represented by the "$\circ$"-shaped samples. When they are added, the result can be represented by the samples located at the solid-circle points, which are dense in the direction $(\theta_p + \theta_q)/2$ (the "fast" direction), and sparse in the orthogonal (or "slow") direction.

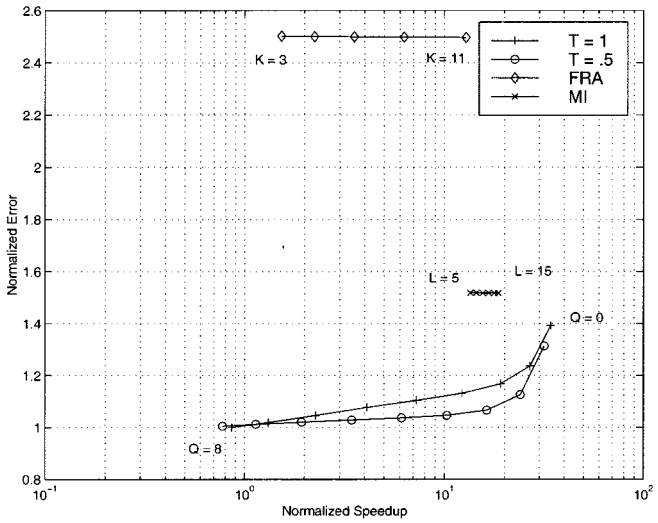


Fig. 8. Comparison of CPU time versus rms error for our implementations of the FRA, MI and FHBP methods on a $N = 256$ pixel Shepp–Logan head phantom, using $P = 512$ projections on $[0, \pi)$. We normalized both the CPU time and the rms error to that of the FBP. The parameter K controls the interpolation window in the FRA, and the parameter L controls the minimum number of slow-direction samples in the MI method.

which boosts the sampling rate so that a cheap antialiasing filter can be used. Oversampling radially by a factor of M (not necessarily an integer) implies that the cost of the algorithm increases by the factor M. Furthermore, if the fast backprojection is to be used with ramp-filtered projections, and the ramp filtering is implemented with FFTs, then the radial interpolation step can be efficiently accomplished simultaneously with the ramp filtering, for a total cost of $O(MPN \log_2(NM))$.

IV. OTHER $O(N^2 \log N)$ RECONSTRUCTION ALGORITHMS

Although the proposed algorithm is interesting in its own right, other fast $O(N^2 \log_2 N)$ reconstruction algorithms have been proposed, which warrant comparison. In this section, we first describe (briefly) the class of $O(N^2 \log_2 N)$ reconstruction algorithms known collectively as *Fourier reconstruction algorithms* (FRA), and then *multilevel inversion* (MI).

A. Fourier Reconstruction Algorithms

FRAs are based on the well-known Fourier Slice Theorem [15], which states that the Fourier transform of a projection at angle θ is a radial slice through the 2-D Fourier transform of the object at direction θ. A reconstruction algorithm is then formed by the following sequence of steps:

1) take the (padded) FFT of the projections;
2) interpolate from the polar grid to a 2-D Cartesian FFT grid;
3) take a 2-D Inverse FFT to recover the image.

While FRAs have been thoroughly researched, the difficulty with using them is in step 2). The interpolation step introduces distortions into the reconstruction, and must be done carefully.

As an improvement on FRAs, the method of gridding for reconstruction was proposed in [8]. Gridding is almost the same as the FRA described above, but instead of reconstructing the object $f(n, m)$, gridding calls for the reconstruction of $f(n, m)w(n, m)$, where $w(n, m)$ is a window sequence bounded away from zero. This has the effect of convolving the Fourier transform of f with the FT of w, which is the interpolator in step 2). We have implemented gridding as described in [8]. The various parameters that define the algorithm were set to the recommended values, and are shown in Table I. The only parameter that was left free was the parameter K, which was the linear size of the support of the 2-D interpolator. Thus, the cost of step 2) is αNPK^2. The algorithm's total cost is dominated by the FFTs, and is $N^2 \log_2 N$ assuming N is a power of two for simplicity (this is not a requirement

TABLE II
PARAMETERS USED IN IMPLEMENTATION OF FHBP ALGORITHM

Parameter	Meaning	Value Used
ψ	Angular decimation filter	[0.5,1,0.5]
$\phi(t)$	Radial interpolation kernel	$\text{sinc}\left(\frac{t}{T}\right)\cos\left(\frac{\pi t}{6T}\right)$ for $\lvert t\rvert < 3T$
$b(t)$	Spatial sampling kernel	$b(x,y) = 1$ for $\sqrt{x^2+y^2} \leq 1/2$
ρ	Ramp filter for FBP-type reconstructions	Hamming windowed ideal ramp
T	Radial sampling period	1.0

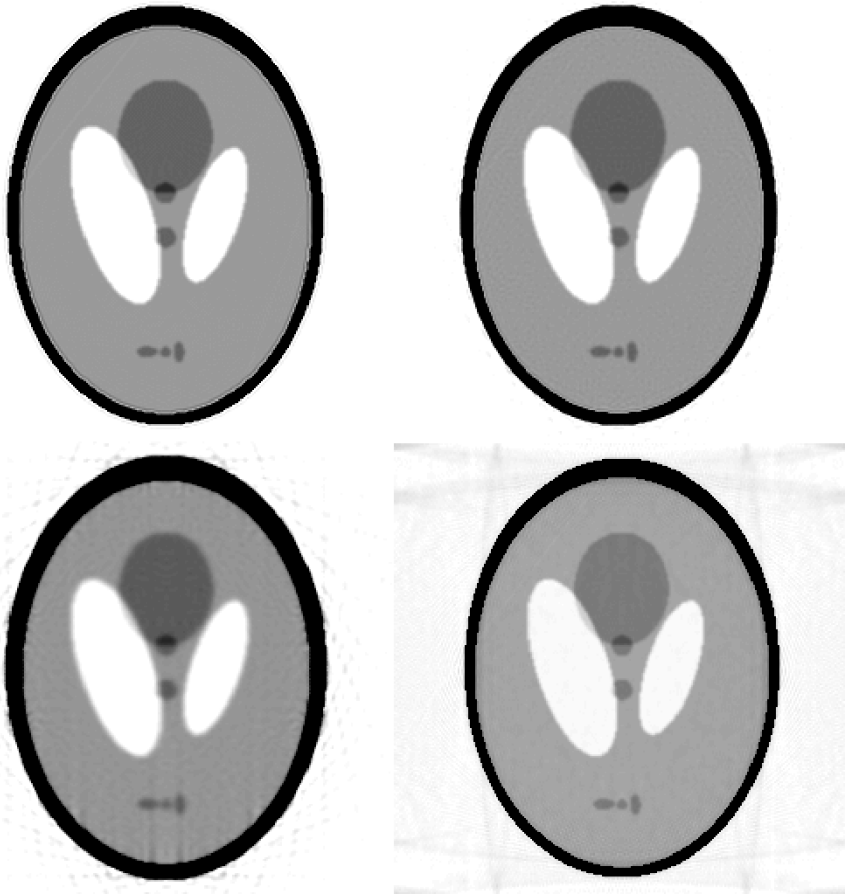


Fig. 9. Reconstructions of the Shepp–Logan head phantom on a 256×256 pixel grid, using the FBP, FHBP, FRA, and MI algorithms. The parameters for all algorithms were chosen to yield a speedup of roughly one order of magnitude. Reconstructions are windowed to [0, 0.05] to show the details. Clockwise, from upper left, are the reconstructions using the FBP, the FHBP with $\times 2$ radial oversampling and $Q = 3$, the FRA with $K = 3$, and MI with the minimum number of samples in the slow direction set to 15.

of gridding). Dividing out the weight $w(n, m)$ requires a negligible N^2 computations.[5]

B. Multilevel Inversion

The method of Brandt *et al.* [9] is based on the following observation. A backprojection is the sum of backprojected single projections, each at direction θ_p. Now, the backprojected projection varies only in the θ_p direction, and is constant in the orthogonal direction (see Fig. 7). Hence, a backprojected projection can be efficiently represented on a grid that is sampled with period T in the θ_p direction (called the *fast direction*), and has only one sample in the orthogonal direction (called the *slow direction*). Now, if we add two adjacent backprojected projections at directions θ_p and θ_{p+1} together, the resulting function can be efficiently represented on a grid that is sampled with period T in the direction $(1/2)(\theta_p + \theta_{p+1})$, which is the fast direction of the sum of the two grids, and with a small number of samples in the slow direction. The process of taking two of these adjacent, nonuniformly sampled grids, and computing a new grid of samples at the intermediate angle with a higher sampling rate in the slow direction is known as a *grid merge*. Furthermore, formulae and guidelines are given in [9] to determine the sampling rate

[5]Note that if the projection data and image are both real, then the number of operations for the FRA can be decreased by a factor of two by exploiting the symmetries in the Fourier transform. We did not exploit this fact in our simulations.

in the slow direction, as well as the types of interpolation and extrapolation that must be performed to compute a grid merge. Thus, MI consists of the following steps:

1) initialize $P = 2^Q$ grids with sampling rate $1/T$ in the fast direction and a single sample in the slow direction;
2) merge adjacent pairs of grids, to yield half the number of grids, each with rate $1/T$ in the fast direction, and progressively higher rates in the slow direction;
3) repeat step 2), until only a single grid remains.

If the sampling rates are chosen in accordance with the formulae of [9], then this final grid will be a uniformly sampled representation of the backprojected function. A set of grid merges, as required for step 2), can be shown to require $O(NP)$ computations, and a total of $\log_2 P$ iterations of step 2) are required to recover the backprojected function. Thus, the total cost is $O(NP \log_2 P)$. If we assume P is order N, then the algorithm is an $O(N^2 \log N)$ algorithm, like the other fast algorithms described in this paper.

Due to the interpolations and approximations made in the grid merging process, the resulting backprojected function is generally degraded by a blur. To overcome this blur, Brandt *et al.* propose a deconvolution step using an estimate of the PSF of the algorithm. Note that the reconstruction algorithm is not shift invariant, and so an approximation to the PSF is obtained by averaging point reconstructions at different spatial locations. The process is described in detail in [9]. Also, the algorithm can be parameterized in a number of ways to trade complexity for performance. One of these ways, as discussed in [9], is to place a lower limit on the number of samples used in the slow direction of any grid. This effectively makes the grids at lower levels in the merging process more densely sampled, and thus increases the computational complexity. However, the oversampled functions are then easier to interpolate using linear interpolation. The results of this tradeoff and the algorithm's performance relative to the other methods is discussed in the next section.

V. Simulations and Performance Evaluation

We now present some numerical results demonstrating the performance of the proposed algorithm. Our simulations are divided into two parts. First, we compare the performance of our algorithm with the FRA and MI algorithms on analytical projections of the Shepp–Logan head phantom[6] (see [23, p. 439]). In the second part of our simulations, we demonstrate the performance of the FHBP algorithm on synthetic projections of CT reconstructions from the visual human dataset (VHD) [24], and present a characterization of the point spread function (PSF) for the algorithm.

For both sets of simulations, the FHBP parameter choices (apart from the choice of Q and T) were fixed at the values indicated in Table II. Fig. 8 shows the results of comparisons between the rms reconstruction error (relative to a digitized version of the phantom) and actual CPU time for our implementations of the FHBP, MI, and FRA algorithms. All algorithms reconstructed $N = 256$ pixel images from $P = 512$ projections, uniformly spaced on $[0, \pi)$, with a detector spacing of $T = 1.0$. The projections were computed analytically, with integrating

[6]This phantom is not the original phantom designed by Shepp and Logan in [21]. Instead, it has been adjusted by Rowland in [22] to represent densities relative to water. As such, it has a high skull/soft tissue contrast with the skull density 100 times greater than the interior tissue. For this reason, it is a challenging phantom to reconstruct. The phantom in [23] is Rowland's version, with one of the larger ellipses removed.

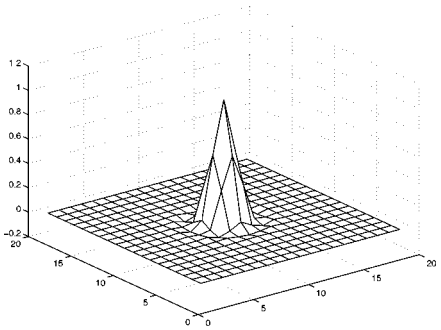


Fig. 10. Center pixel PSF for the FBP.

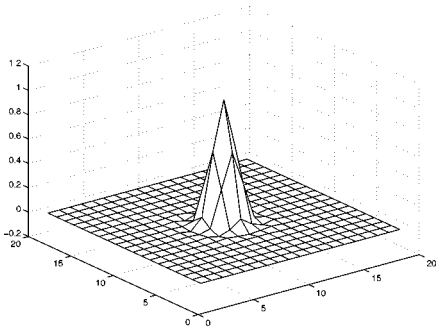


Fig. 11. Center pixel PSF for the FHBP with $\times 2$ radial upsampling and $Q = 3$.

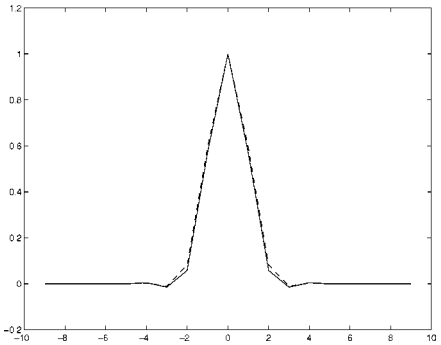


Fig. 12. Slice through the x-axis of the PSF for the FBP (solid line) and FHBP (dashed line).

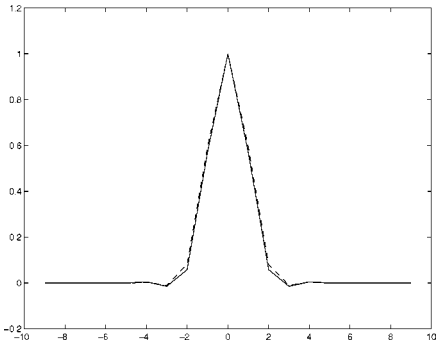


Fig. 13. Slice through the y-axis of the PSF for the FBP (solid line) and FHBP (dashed line).

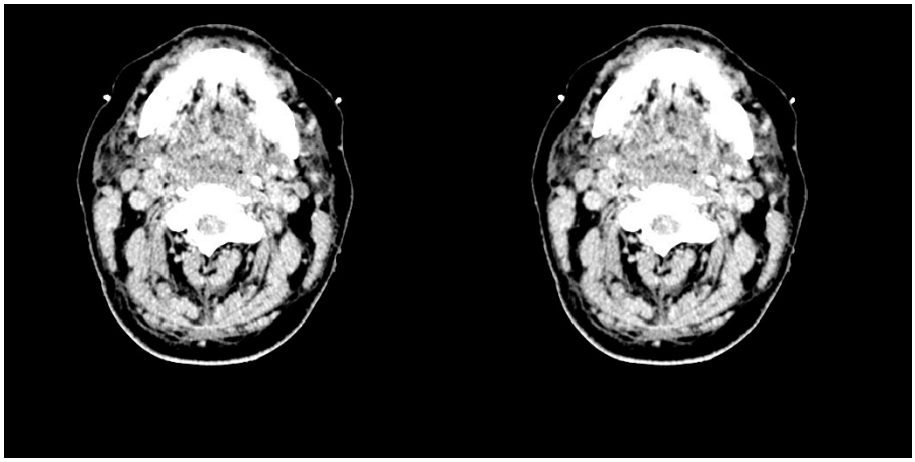


Fig. 14. Left: Reconstruction from synthetic projections of a CT slice from the Visual Human Data set using the FBP. Original image was 512×512, and $P = 1024$ projections on $[0, \pi)$ were computed with $T = 1.0$. Image is shown windowed to [1000, 1100] HU. Right: Reconstruction using FHBP with $\times 2$ radial upsampling, and $Q = 3$. Speedup of FHBP over FBP (excluding filtering time), was $40\times$.

detectors of width 1.0 used in the radial sampling step. The parameters for the FRA are given in Table I. The computational cost was traded off for accuracy by changing K, which controls the number of points to use in the interpolation step. For MI, the cost was traded off for accuracy by changing the minimum number of samples in the slow direction. For the FHBP algorithm, the cost and performance tradeoff was accomplished by varying two parameters: i) Q, the number of times the exact decomposition is used; and ii) the radial upsampling factor, applied prior to the FHBP algorithm. All CPU times and reconstruction errors were normalized relative to an implementation of the direct FBP [implemented via (7)], and radial ramp filtering times were included in the run times for the appropriate algorithms (the FRA does not require a separate radial filtering step). Note that the fastest speedup plotted in Fig. 8 for the FHBP algorithm using no radial oversampling and $Q = 0$, is roughly $\times 20$, which compared to $N/\log_2 N = 32$, indicates that the implementation-dependent constants for the FHBP are close to the relevant constants for the FBP. In examining Fig. 8, the most striking aspect is the almost complete lack of improvement in reconstruction performance with increased computational cost for the MI and FRA algorithms. We will return to this point shortly.

In Fig. 9, we show sample reconstructions for the four algorithms under consideration, all of which have been windowed to the same grayscale level of [0, 0.05] to show the finer details. Apart from the FBP, the parameters for each of the FHBP, MI, and FRA algorithms were chosen so as to yield a speedup of roughly $\times 10$ over the FBP. For the MI, this meant taking the maximum of $L = 15$ points in the slow direction. For the FRA, the interpolation window size was set to the minimum $K = 3$. For the FHBP, we chose to radially upsample by a factor of 2, and set $Q = 3$. The visual reconstruction quality for the FRA, FHBP and FBP algorithms are quite similar. Although we have not studied the phenomenon in detail, the reason for the large MSE errors in the FRA appears to be due to a combination of spatial aliasing ("ghost" images of lower intensity that overlap the desired image, and are due to truncation in the ideal interpolation) and a slight geometric distortion, also a result of the interpolation step.

For the MI, the errors are more obvious. The streaking and blurring artifacts are definitely due to the postprocessing step. The deconvolution of the Gaussian approximation to the point spread function causes artifacts in the reconstruction that only show up at the lower level of densities. We note that the authors of the MI method claim to obtain reconstructions that are sharper than the FBP by postprocessing using deconvolution with a Gaussian approximation to the PSF [9]. However, we note that none of the images in [9] appear to have the contrast requirements of the Shepp–Logan head phantom, and that in at least one case, an image in [9] shows significant ringing due to the deconvolution step. We also note that the authors of the MI method claim that these problems can be overcome, but we were unable to obtain further details of the deconvolution step. Finally, we find that the MI algorithm's postprocessing step is difficult to implement in practice, as it is the equivalent of solving an image restoration problem that is inherently difficult. Indeed, in [25], it is shown that Gaussian PSFs lead to the most ill-posed deconvolution problems.

Next, we characterized the PSF of the FHBP algorithm, using the same parameters as specified in Table II. First, we generated projection data corresponding to ten randomly placed point masses in the plane. These projections were then filtered, and backprojected using the FBP and FHBP algorithms. The resulting reconstructed masses were qualitatively similar for all points in the plane, suggesting a nearly spatially invariant PSF. We thus computed PSFs for the center point for both the FBP and FHBP algorithms. The resulting PSF for the FBP and FHBP algorithms are shown in Figs. 10 and 11, respectively. Comparisons of slices through the x and y axis of the PSFs are shown in Figs. 12 and 13, respectively. These figures were all obtained using the same parameters as were used in Fig. 9, i.e., $Q = 3$, and $\times 2$ radial upsampling. We note that the PSFs for the FBP and FHBP algorithms are extremely close, reinforcing our conclusions from the phantom study.

As a final subjective comparison, we took a CT reconstruction[7] from the female visual human dataset, which was a 512×512 image through the base of the skull, showing both the spinal cord and surrounding soft tissue. We then computed discrete projections using a discrete approximation to the Radon transform, which used circular pixels and integrating

[7]Slice number 1194, to be precise.

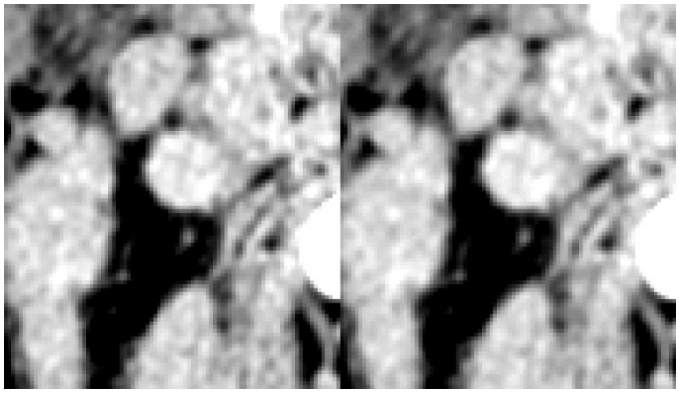


Fig. 15. Enlargements of a region from the images in Fig. 14

detectors. A total of $P = 1024$ projections on $[0, \pi)$ were used, with a radial sampling interval of $T = 1.0$. The FHBP algorithm was run with $\times 2$ radial upsampling, and $Q = 2$. The speedup (excluding ramp filtering time) measured was $\times 40$ for the FHBP versus the FBP. The resulting reconstructions are shown in Fig. 14, windowed to [0, 100] Hounsfield units (HU) so that the soft tissue can be seen. The reconstructions are visually indistinguishable. In Fig. 15, a 100×100 pixel region near the base of the spinal column is shown enlarged, also windowed to [0, 100] HU. Greater performance gains should be obtainable through the obvious parallel implementation of the algorithm and optimization of the code. In comparison, in [8], the authors reported that gridding achieved a speedup of roughly ten under similar circumstances.

On the basis of this phantom and repeated experiments for different sized pixel grids (up to 1024×1024), we would then suggest an empirical speedup of $\approx N/\log_2 N$ for reconstructions of visual quality similar to the FBP via the FHBP algorithm. Extrapolating this speedup suggests that for high resolution medical imaging, with 4096×4096 pixel images, speedups of greater than two orders of magnitude should be achievable with reconstructions of visual quality comparable to the FBP.

VI. Conclusions

We have constructed a class of fast algorithms for fast reconstruction from 2-D tomographic data. These algorithms provide orders of magnitude speedup in reconstruction time with little or no added distortion. The proposed algorithms are parallelizable, simple, and, in our experiments, outperform Fourier Reconstruction Algorithms as well as the MI method in terms of reconstruction distortion and CPU time.[8] We are examining the possibility of constructing fast algorithms for other imaging geometries in both 2-D and 3-D.

References

[1] S. R. Deans, *The Radon Transform and Some of Its Applications*. New York: Wiley, 1983.

[2] I. Agi, P. Hurst, and K. Current, "An image processing IC for backprojection and spatial histogramming in a pipelined array," *IEEE J. Solid-State Circuits*, vol. 28, pp. 210–221, Mar. 1993.

[3] R. M. Mersereau, "Direct Fourier transform techniques in 3-D image reconstruction," *Comput. Biol. Med.*, vol. 6, pp. 247–258, 1976.

[4] H. Stark, J. Woods, P. Indraneel, and P. Hingorani, "An investigation of computerized tomography by direct Fourier inversion and optimum interpolation," *IEEE Trans. Biomed. Eng.*, vol. 28, no. 7, pp. 496–505, 1981.

[5] R. M. Lewitt, "Reconstruction algorithms: Transform methods," *Proc. IEEE*, vol. 71, pp. 390–408, Mar. 1983.

[6] J. O'Sullivan, "A fast sinc function gridding algorithm for Fourier inversion in computer tomography," *IEEE Trans. Med. Imag.*, vol. 4, no. 4, pp. 200–207, 1985.

[7] S. Matej and I. Bajla, "A high-speed reconstruction from projections using direct Fourier method with optimized parameters—An experimental analysis," *IEEE Trans. Med. Imag.*, vol. 9, no. 4, pp. 421–429, 1990.

[8] H. Schomberg and J. Timmer, "The gridding method for image reconstruction by Fourier transformation," *IEEE Trans. Med. Imag.*, vol. 14, Sept. 1995.

[9] A. Brandt, J. Mann, M. Brodski, and M. Galun, "A fast and accurate multilevel inversion of the radon transform," SIAM J. Appl. Math., to be published.

[10] P. Edholm, G. Herman, and D. Roberts, "Image reconstruction from linograms: Implementation and evaluation," *IEEE Trans. Med. Imag.*, vol. 7, pp. 239–246, Sept. 1988.

[11] S. Nilsson, "Fast backprojection," Dept. of Electrical Engineering, Linköping University, S-58 183 Linköping, Sweden, Tech. Rep. LiTH-ISY-R-1865, ISSN 1400-3902, July 1996.

[12] A. Brandt and J. Dym, "Fast computation of multiple line integrals," *SIAM J. Sci. Comput.*, vol. 20, pp. 1417–1429, 1999.

[13] M. L. Brady, "A fast discrete approximation algorithm for the Radon transform," *SIAM J. Comput.*, vol. 27, pp. 107–119, Feb. 1998.

[14] A. Boag, Y. Bresler, and E. Michielssen, "A multilevel domain decomposition algorithm for fast reprojection of tomographic images," *IEEE Trans. Image Processing*, to be published.

[15] F. Natterer, *The Mathematics of Computerized Tomography*. New York: Wiley, 1986.

[16] A. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. New York: IEEE Press, 1988.

[17] T. Moon and W. Sterling, *Mathematical Methods and Algorithms for Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 2000.

[18] D. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1969.

[19] P. A. Rattey and A. G. Lindgren, "Sampling the 2-D radon transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, no. 5, pp. 994–1002, 1981.

[20] F. Natterer, "Efficient evaluation of oversampled functions," *J. Comput. Appl. Math.*, vol. 14, pp. 303–309, 1986.

[21] L. A. Shepp and B. F. Logan, "The Fourier reconstruction of head section," *IEEE Trans. Nucl. Sci.*, vol. 21, no. 1, pp. 21–43, 1974.

[22] G. T. Herman, Ed., *Image Reconstruction from Projections*. Berlin, Germany: Springer-Verlag, 1979, vol. 32.

[23] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

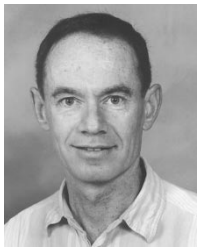
[24] Nat. Inst. Health, "Visual human dataset," Nat. Lib. Med., Bethesda, MD.

[25] J. Philip, "The most ill-posed nonnegative kernels in discrete deconvolution," *Inv. Prob.*, vol. 3, 1987.

[8] We note that CPU time comparisons depend, to some extent, on the architecture of the machine, as well as the level of optimization of the codes. Thus, in practice, changes in the run time on the order 2–3 times can be expected, due to these factors.

Samit Basu (S'97–M'00) received the B.E.E. degree (magna cum laude) in 1995 from the University of Delaware, Newark, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign, in 1998 and 2000, respectively. While studying, he was supported by the Eugene DuPont Scholarship, a graduate fellowship from the Electrical and Computer Engineering Department, University of Illinois, a graduate fellowship from the Joint Services Electronics Program, and the Mac VanValkenburg Memorial Fellowship.

In 2000, he joined the General Electric Corporate Research and Development Center, Niskayuna, NY, where he is an Electrical Engineer working on problems in tomographics reconstruction and signal processing.



Yoram Bresler (S'82–M'85–SM'91–F'99) received the B.Sc. (cum laude) and M.Sc. degrees from the Technion—Israel Institute of Technology, Haifa, in 1974 and 1981, respectively, and the Ph.D. degree from Stanford University, Stanford, CA, in 1985, all in electrical engineering.

From 1974 to 1979, he was an Electronics Engineer with the Israeli Defense Force. From 1979 to 1981, he was a Consultant with the Flight Control Laboratory, Technion, developing algorithms for autonomous TV aircraft guidance. From 1985 to 1987 he was a Research Associate with the Information Systems Laboratory, Stanford University, working on sensor array processing and medical imaging. In 1987, he joined the University of Illinois at Urbana-Champaign, where he is currently a Professor in the Department of Electrical and Computer Engineering and the Bioengineering Program, and Research Professor with the Coordinated Science Laboratory. In 1995–1996, he spent a sabbatical leave at the Technion. His current research interests include multidimensional and statistical signal processing and their applications to inverse problems in imaging and sensor arrays. He is currently on the editorial board of *Machine Vision and Applications*.

Dr. Bresler was an Associate Editor for the IEEE TRANSACTIONS FOR IMAGE PROCESSING (1992–1993), and a Member of the IEEE Image and Multidimensional Signal Processing Technical Committee (1994–1998). In 1988 and 1989, he received the Senior Paper Awards from the IEEE Acoustics, Speech, and Signal Processing Society. He is the recipient of a 1991 NSF Presidential Young Investigator Award, the Technion Fellowship in 1995, and the Xerox Senior Award for Faculty Research in 1998. He was named a "Universtity of Illinois Scholar" in 1999.