An $O(N^2 \log N)$ Multilevel Backprojection Method

Achi Brandt
Jordan Mann
Matvei Brodski
Weizmann Institute of Science
Rehovot, Israel

Abstract

A number of imaging technologies reconstruct an image function from its Radon projection using the convolution backprojection method. The convolution is an $O(N^2 \log N)$ algorithm, where the image consists of $N \times N$ pixels, while the backprojection is an $O(N^3)$ algorithm, thus constituting the major computational burden of the convolution backprojection method. An $O(N^2 \log N)$ multilevel backprojection method is presented here. When implemented with a Fourier-domain postprocessing technique, also presented here, the resulting image quality is similar to or superior than the image quality of the classical backprojection technique.

1 Background

Reconstruction of a function of two or three variables from its Radon transform has proven vital in computed tomography (CT), nuclear magnetic resonance imaging, astronomy, geophysics, and a number of other fields[3]. One of the best known reconstruction algorithms is the convolution backprojection method (CB), which is widely used in commercial CT devices[3] (with rebinning for divergent-beam projections[6]). Recently, it has been applied to spotlight-mode synthetic aperture radar image reconstruction[4], in which the conventional method is the direct Fourier method (DF), i.e., Fourier-domain interpolation followed by 2-D FFT[9].

Originally, CB was preferred to DF since the former provided better images[6, 8]. However, since the backprojection part of CB raises the computational complexity of the method to $O(N^3)$, while DF's complexity is $O(N^2 \log N)$, there has been interest in finding effective implementation of DF[1, 5, 7, 11, 12, 13, 14, 15]. We present here an $O(N^2 \log N)$ backprojection algorithm, based on a multiscale approach, which, when used as the second half of CB, reduces the complexity of CB to $O(N^2 \log N)$. Empirical results indicate that the multiscale backprojection, together with the postprocessing step described in the second half of this paper, produces images of quality equal to or better than that of of the classical backprojection algorithm and better than the quality of DF. Furthermore, multilevel methods can generally be applied under weaker regularity requirements than Fourier methods can. For example, the algorithm presented here could be adjusted to provide different resolutions for different parts of the reconstruction, even if the Radon data are equally spaced.

The multilevel approach described here is structurally similar to the fast Radon transform described in [2].

2 Mathematical Preliminaries

Let f be an absolutely integrable function of two variables. $P_\theta f(t)$, the Radon transform of the function at angle θ, is then defined by

$$P_\theta f(t) = \int_{-\infty}^{\infty} f(t\cos\theta - \tau\sin\theta, t\sin\theta + \tau\cos\theta)\, d\tau. \tag{1}$$

This is the integral of f along the line

$$x\cos\theta + y\sin\theta = t$$

in the x-y plane. (See Figure 1). In many imaging technologies, such as CT, the challenge is to compute f, or an approximation to it, given samples of $P_\theta f(t)$ for finitely many values of θ and t.

To understand CB, which is meant to solve this problem, we will need the Fourier transform. For $f \in L^1(\mathbb{R}^d)$, the Fourier transform $\hat{f}$ of f is defined by

$$\hat{f}(u) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} f(x)\, e^{ix\cdot u} dx,$$

for all $u \in \mathbb{R}^d$. We note that for any θ, $f \in L^1(\mathbb{R}^2)$ implies that $P_\theta f$ is well defined and belongs to $L^1(\mathbb{R}^1)$, and thus the Fourier transform of $P_\theta f$ is also well defined. Many methods for reconstructing a function from its Radon transform, including CB, rely on the Radon Slice Theorem [6], which states that

$$\widehat{P_\theta f}(\rho) = \sqrt{2\pi}\hat{f}(\rho\cos\theta, \rho\sin\theta). \tag{2}$$

Thus, if $f(x, y)$ is the image function, the 2-D Fourier transform of the image can be sampled by sampling the Fourier transform of the Radon transform of the image for different values of ρ and θ.

3 Convolution Backprojection method

3.1 Theory

One way to reconstruct f from its Radon transform is to compute $\hat{f}$ using (2) for as many values of ρ and θ as possible and then perform a 2-D inverse Fourier transform. Algorithms based on this idea are called direct Fourier methods (DF). CB, however, uses the following approach. The Fourier transform inversion formula [3] can be written in polar coordinates as

$$f(x, y) = \frac{1}{2\pi} \int_0^\pi \int_{-\infty}^{\infty} \hat{f}(\rho\cos\theta, \rho\sin\theta)\, e^{i\rho(x\cos\theta + y\sin\theta)} |\rho|\, d\rho\, d\theta.$$

By the Radon Slice Theorem (2), this equals

$$\frac{1}{(2\pi)^{3/2}} \int_0^\pi \int_{-\infty}^{\infty} \widehat{P_\theta f}(\rho)\, e^{i\rho(x\cos\theta + y\sin\theta)} |\rho|\, d\rho\, d\theta.$$

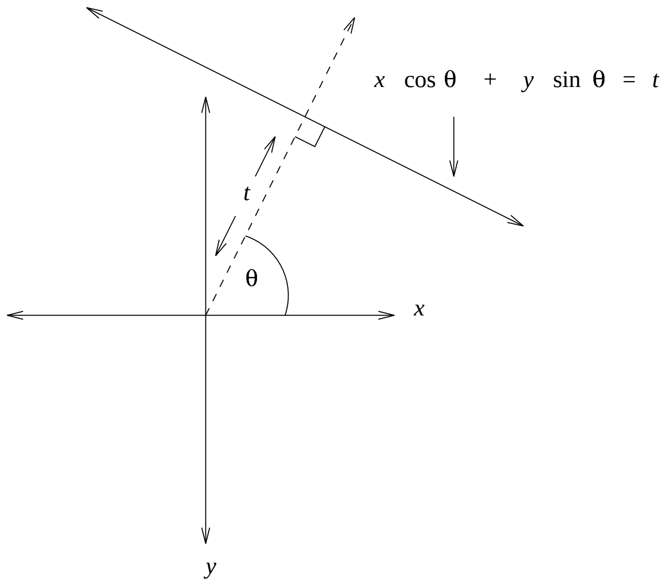

Figure 1: The Radon transform of f at angle θ and position t, denoted $P_\theta f(t)$. This is the integral of f on the line $x\cos\theta + y\sin\theta = t$.

In most applications, values of $P_\theta f$ or $\widehat{P_\theta f}$ are obtained from the physical data. Using the $\widehat{P_\theta f}$ values, we may compute a function $\tilde{P}_\theta f$, defined by

$$\tilde{P}_\theta f(u) = \int_{-\infty}^{\infty} \widehat{P_\theta f}(\rho)\, e^{i\rho u}|\rho|\, d\rho, \tag{3}$$

and then

$$f(x,y) = \frac{1}{(2\pi)^{3/2}} \int_0^{\pi} \tilde{P}_\theta f(x\cos\theta + y\sin\theta)\, d\theta, \tag{4}$$

which means that $f(x,y)$ can be reconstructed from the $\tilde{P}_\theta f$ values, given these values for all θ.

3.2 Implementation and computational complexity

Of course, in actual applications, $P_\theta f(t)$ is known for only a finite number of values of θ and t, and f is to be computed at a finite set of points, or pixels, (x,y). We will assume here that the pixels (x_i, y_i) to be computed lie inside a circle inscribed in an $N \times N$ square grid, with a distance d between adjacent pixel centers in both the horizontal and vertical directions, that there are Q angles θ_j at which $P_\theta f$ is known, and that at each such angle θ_j, $P_\theta f(t)$ is known at N evenly spaced values t_k of t, the difference between consecutive values t_k, t_{k+1} being d as well. Since we limit the region of reconstruction to the inscribed circle, N samples of the Radon transform in any direction are enough to cover the region. We will further assume that Q is a power of 2, though the algorithm can clearly be generalized to the case of arbitrary Q.

Computation of $\tilde{P}_\theta f$ is called the convolution step because multiplying two functions together in the Fourier domain and taking the inverse Fourier transform of the result is associated with convolution [3]. The Fourier transform and inverse Fourier transform required in (3) for this step are approximated in practice using the discrete Fourier transform (DFT) and its inverse. Since there are N values of $P_\theta f(t)$ for each value of θ, the computation of $\tilde{P}_\theta f$ for a single value of θ is $O(N \log N)$ using the Fast Fourier Transform (FFT), and since there are Q values of θ, the entire convolution step is $O(QN \log N)$. If, as is usually assumed, $Q \approx N$, then the order of the convolution step is $N^2 \log N$.

The integral with respect to θ in (4) is replaced in practice by summation over all available values of θ. Since there are $O(N^2)$ pixels at which summation is required, and Q values of θ, this final step of computing the right side of (4) is clearly $O(QN^2)$, or, if $Q \approx N$, $O(N^3)$. Since the contribution of $\tilde{P}_\theta f$ to a pixel (x, y) depends only on the quantity $x \cos\theta + y \sin\theta$, $\tilde{P}_\theta f(u)$ is said to be "smeared" or "backprojected" along the line $x \cos\theta + y \sin\theta = u$ for each value of u. That is, it is added to every pixel lying on that line. Therefore, this final step is called the backprojection step. Although the concept of backprojection suggests an algorithm with a different loop structure than the one suggested by (4) — namely, (4) suggests an outer loop to go through all the pixels and an inner loop to go through the values of θ, whereas the backprojection concept suggests the reverse — the backprojection concept does not change the computational complexity. Although DF is $O(N^2 \log N)$, commercial CT scanners have traditionally used CB, despite its higher computational complexity, in part because of the superior quality of the images it produces[6].

4 Multiscale Backprojection

It follows from the analysis in subsection (3.2) that an $O(N^2 \log N)$ implementation of the backprojection step of CB would lower the complexity of the entire CB algorithm to $O(N^2 \log N)$. Such an implementation is presented here.

4.1 Basic concept

The multilevel backprojection method relies on the following reasoning. In the standard backprojection algorithm, a single $N \times N$ grid is used to sum the appropriate values of $\tilde{P}_{\theta_j} f$, for $j = 1, \ldots, Q$, for each pixel. One can, however, start with Q grids g_j^0, $j = 1, \ldots, Q$, and for each j project $\tilde{P}_{\theta_j} f$ only onto g_j. Later, the Q different grids can be added together pixelwise to produce the final image. This approach can be used to save computation in the following way.

In what follows, for all i and j, f_j^i will be a function of two continuous variables, and g_j^i will be a "grid" containing a finite set of samples of f_j^i. For all j, let

$$f_j^0(x, y) = \tilde{P}_{\theta_j} f(x \cos\theta + y \sin\theta).$$

I.e., f_j^0 is the function of two variables resulting from the backprojection of $\tilde{P}_{\theta_j} f$ along the lines $x \cos\theta + y \sin\theta = t$, for different values of t, in the x-y plane. Clearly, f_j^0 does not vary along such lines, and therefore, when collecting samples of f_j^0 to form the grid g_j^0, it is sufficient to compute and store one point value in the grid for each of the N values of t at which $\tilde{P}_{\theta_j} f(t)$ has been computed, rather than computing and storing N^2 point values. See Figure 2. In fact, g_j^0 is merely a 1-D array containing all the computed values of $\tilde{P}_{\theta_j} f$, and is compiled by the convolution part of the CB algorithm. Thus, although we are beginning the backprojection with Q grids instead

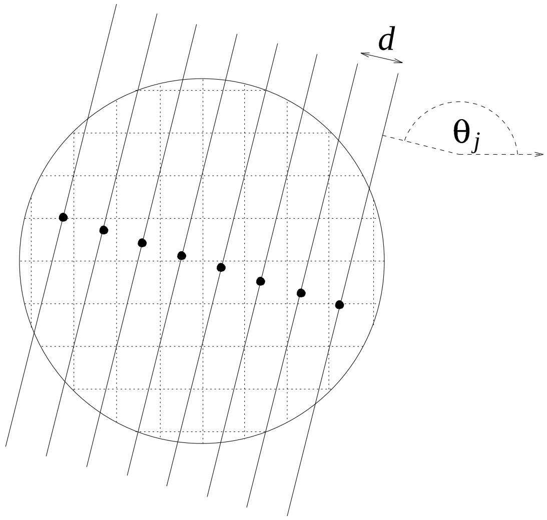

Figure 2: Construction of the zeroth-level grid g_j^0, which contains the results of a single backprojection, that of $\tilde{P}_{\theta_j} f$. Although g_j^0 is meant to contain the samples of the function f_j^0 defined on a disk in the plane, and the disk contains $O(N^2)$ pixels, shown in the figure as the intersections of dotted lines, it is sufficient to compute and store in g_j^0 samples of f_j^0 taken at only $O(N)$ points, shown in the figure as black dots, since f_j^0 only varies in the direction $(\cos\theta_j, \sin\theta_j)$.

of one, the total number of point values stored is $O(NQ)$. The distance between adjacent sample points of f_j^0 for any j is d, the distance between adjacent sample points of $\tilde{P}_{\theta_j} f$.

Eventually, the various initial functions f_j^0 must be added together to form the final image. As addition is commutative and associative, they may be added together in any order, and our method chooses an order that reduces the number of necessary computations. Since Q is even by assumption, our first level of grid merges will consist of adding together pairs of functions f_j^0, f_k^0 whose corresponding values of θ are close to each other, and thus the direction in which f_j^0 is constant is close to the direction in which f_k^0 is constant. The sum of these two functions, f_l^1, will vary slowly in a certain direction, and thus it will only be necessary to store samples of f_l^1 in g_l^1 at a handful of widely-spaced points along each line parallel to the "slow" direction in the x-y plane.

Specifically, we may assume without loss of generality that $0 \le \theta_j < \pi$ for all j and that the θ_j are ordered in such a way that $\theta_j < \theta_{j+1}$ for all $j = 1, \ldots, Q-1$. For all $j = 1, \ldots, Q$, let $\theta_j^0 = \theta_j$. For all k from 1 to $Q/2$, we will add together the functions f_{2k-1}^0 and f_{2k}^0 to obtain f_k^1. Since for any j, f_j^0 does not vary in the direction $(-\sin\theta_j, \cos\theta_j)$, it follows that f_{2k-1}^0 and f_{2k}^0 vary slowly in the direction $(-\sin\theta_k^1, \cos\theta_k^1)$, where θ_k^1 is the average of θ_{2k-1} and θ_{2k}. Therefore f_k^1 varies slowly in the $(-\sin\theta_k^1, \cos\theta_k^1)$ direction, and when forming g_k^1, it is only necessary to store samples of f_k^1 at a few widely-spaced points along each line in that direction. The spacing between sample points in the fast direction, i.e., $(\cos\theta_k^1, \sin\theta_k^1)$, will again be d. The samples are computed from g_{2k-1}^0 and g_{2k}^0 by interpolation.

By similar reasoning, for $l = 1, \ldots, Q/4$, we may add f_{2l-1}^1 and f_{2l}^1 together to obtain f_l^2 in such a way that f_l^2 varies slowly in the direction $(-\sin\theta_l^2, \cos\theta_l^2)$, where θ_l^2 is the average of θ_{2l-1}^1 and θ_{2l}^1, and consequently, when computing and storing samples of f_l^2 to form g_l^2, it is only necessary to store values of f_l^2 at a few widely-spaced points in the direction $(-\sin\theta_l^2, \cos\theta_l^2)$. The spacing between sample points in the fast direction will again be d. f_l^2 will not vary as slowly in its "slow" direction as f_{2l}^1 varies in *its* "slow" direction, and therefore g_l^2 will require more sample points in its slow direction than g_{2l}^1 requires in its slow direction. Nevertheless, there are half as many second-level grids g_l^2 as there are first-level grids g_k^1, and as we will show, the total number of point values that must be computed and stored for all the grids at any one level of merges is $O(N^2)$. See Figure 3. Samples of f_l^2 to be stored in g_l^2 are computed from g_{2l-1}^2 and g_{2l}^2 by interpolation.

Continuing in this way, we may construct a sequence of levels of grids and functions. The functions at the i-th level are constructed from pairs of functions at the $(i-1)$-th level in such a way that each i-th level functions varies slowly in a certain direction and only a few samples along lines in that direction need to be stored in the function's grid. At the $\log_2 Q$-th level, there is only one grid, and this grid represents the sum of all the original grids g^0, that is, the sum of all the backprojections of the $\tilde{P}_{\theta_j} f$. This grid is therefore the resulting reconstruction. Since $O(N^2)$ operations are needed to build the grids at each level, the overall cost of the algorithm is $O(N^2 \log Q)$.

4.2 Computing sample point spacing in the "slow" direction

As explained in subsection 4.1, the low computational complexity of the algorithm depends on the judicious choice of sample point spacing, and hence, the number of sample points, in a grid's slow direction. This spacing is chosen based on the following reasoning.

Since at every merge level i, two consecutive functions f_{2k-1}^{i-1} and f_{2k}^{i-1} from the previous level are merged to form f_k^i, it follows that 2^i consecutive original functions f_j^0 were merged and remerged to obtain f_k^i. We will refer to these 2^i original functions as the merged original functions. We note that the slow directions of any two consecutive original functions f_j^0 and f_{j+1}^0 (actually, since these are original functions, these are not just slow directions but constand directions) differ by π/Q radians, and it follows from this that the slow directions of the first and last of the merged original functions differ by $(2^i-1)\pi/Q$ radians. It is trivial to show that the slow direction of f_k^i is halfway between the slow directions of the first and last merged original functions, and thus it differs by no more than $(2^i-1)\pi/(2Q)$ from the slow direction of any of the merged original functions.

For any f_k^i, the fast direction is perpendicular to the slow direction and the spacing in the fast direction is d. We wish to choose spacing in the slow direction of the function f_k^i in such a way that interpolating a value of f_k^i between two sample points A and B adjacent to each other in the slow direction would incur an error no greater than the error incurred by interpolating a value of any of the merged original functions between A and B. The spacing in the slow direction should therefore be such that if one selects a coordinate axis in the fast direction of any of the merged original functions, then the distance along this coordinate axis between A and B will be no more than d. It can be seen from Figure 4 that the desired distance between adjacent points in the slow direction of f_k^i is

$$\frac{d}{\sin((2^i-1)\pi/(2Q))}. \tag{5}$$

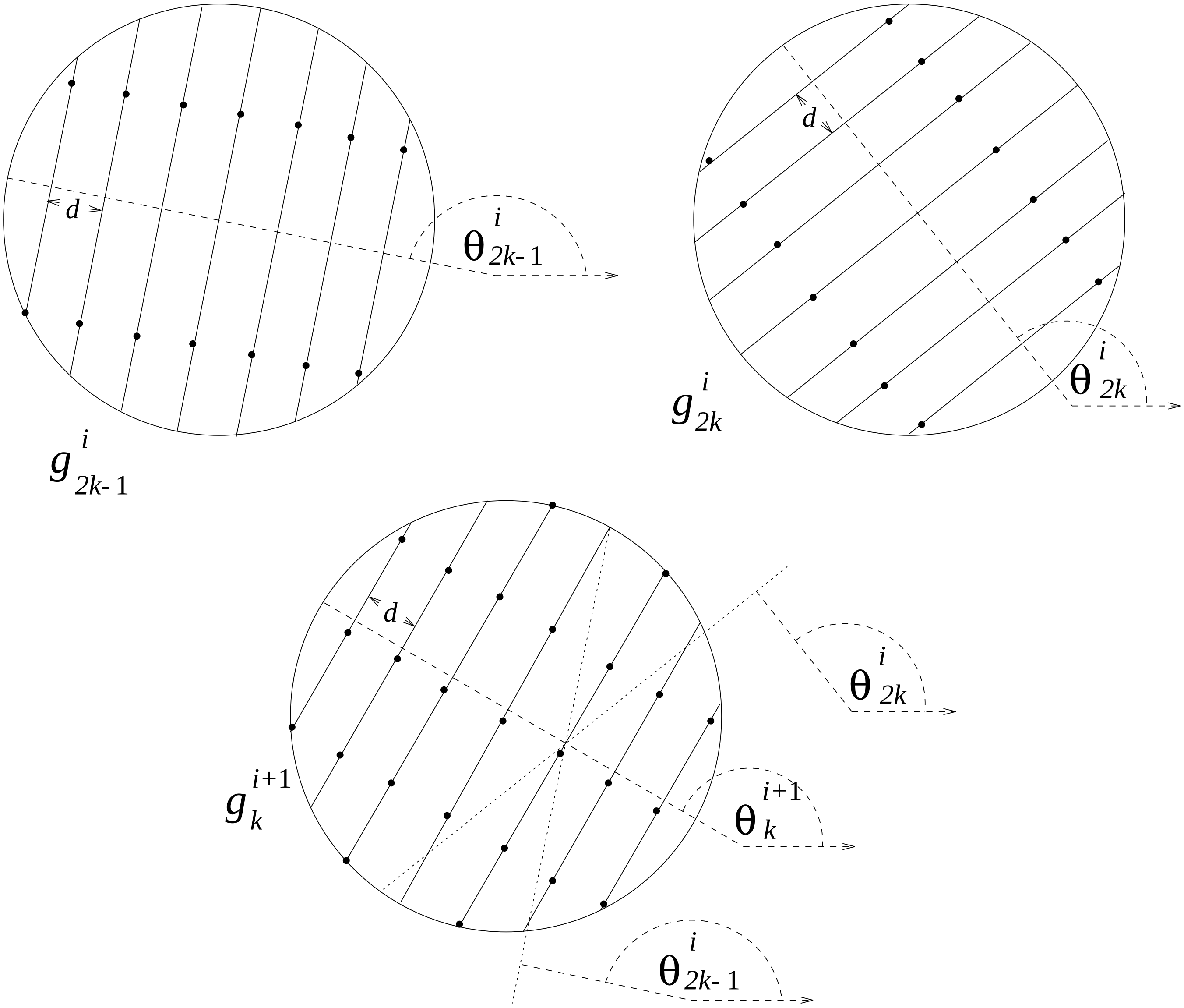

Figure 3: Merge of g^i_{2k-1} and g^i_{2k} to form g^{i+1}_k. The slow directions of f^i_{2k-1}, f^i_{2k}, and f^{i+1}_k are $(-\sin\theta^i_{2k-1}, \cos\theta^i_{2k-1}), (-\sin\theta^i_{2k}, \cos\theta^i_{2k})$, and $(-\sin\theta^{i+1}_k, \cos\theta^{i+1}_k)$ respectively, where θ^{i+1}_k is the average of θ^i_{2k-1} and θ^i_{2k}. Sample points are represented in the diagrams by black dots. g^{i+1}_k contains more samples in its slow direction than g^i_{2k-1} and g^i_{2k} do in theirs. The distance between samples in the fast direction, indicated in the diagrams by the distance between adjacent solid lines, is always d.

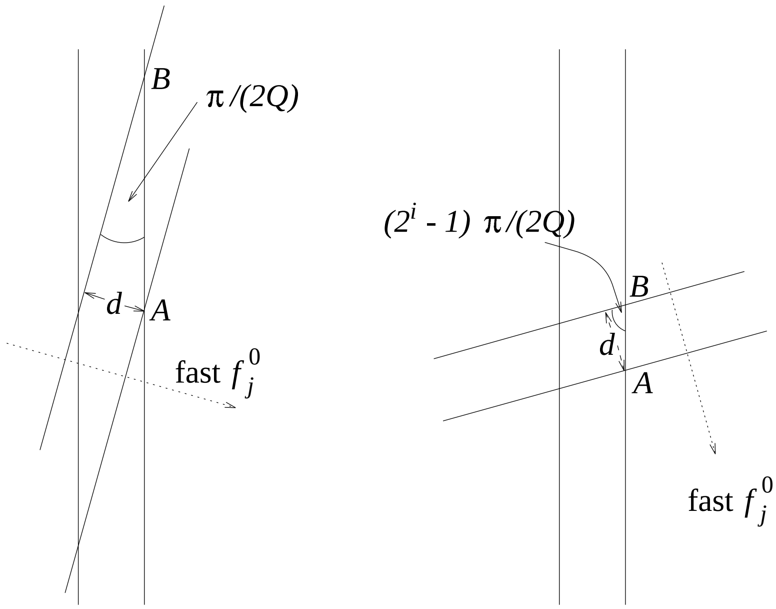

Figure 4: Choice of spacing in slow direction. The difference between the slow direction (vertical lines) of the function f_k^i, currently being formed, and the slow direction (solid diagonal lines) of any of the merged original functions is never greater than $(2^i - 1)\pi/(2Q)$. The fast direction of the merged original function is perpendicular to its slow one and its spacing in that direction is d. The distance in that direction between adjacent sample points A and B of f_k^i should not exceed d. Left, first merge level. Right, arbitrary merge level i.

4.3 Computational complexity

Since the diameter of the support of the reconstruction is Nd, the number of points necessary along a single line of f_k^i in its slow direction is Nd divided by the sampling interval, or

$$N\sin((2^i-1)\pi/(2Q)).$$

We will assume that if this number is not an integer then the lowest integer greater than this number will be used. Since the sample spacing in the fast direction is always d, it follows that for any i and j, there are N lines parallel to the slow direction along which f_j^i will be sampled, so the total number of points in g_j^i is

$$N\lceil N\sin((2^i-1)\pi/(2Q))\rceil.$$

Also, there are $Q/2^i$ different grids g_j^i to be computed (i.e., at the ith level of merges, j takes the values $1,\ldots,Q/2^i$). It follows from this that the total number of samples to be computed and stored at the ith level of merges is

$$\left\lceil N\sin\left((2^i-1)\frac{\pi}{2Q}\right)\right\rceil\left(\frac{NQ}{2^i}\right) < \left(N\sin\left(\frac{2^i\pi}{2Q}\right)+1\right)\left(\frac{NQ}{2^i}\right)$$

$$< \left(N\frac{2^i\pi}{2Q}+1\right)\left(\frac{NQ}{2^i}\right) = \frac{\pi}{2}N^2+\frac{NQ}{2^i}.$$

The sum of this for all levels of merges, i.e., for $i=1,\ldots,\log_2 Q$, is less than

$$\frac{\pi}{2}N^2\log_2 Q+NQ.$$

If, as is normally assumed, $Q\approx N$, then the order of the backprojection is $N^2\log N$.

4.4 Practical considerations

For $i=1,\ldots,\log_2 Q$, samples of f_l^i are computed from g_{2l-1}^{i-1} and g_{2l}^{i-1} by interpolation, but there will generally be some points where f_l^i is to be sampled that are not surrounded by sampling points of f_{2l-1}^{i-1} or of f_{2l}^{i-1}, and extrapolation must be performed there instead of interpolation. Computing a few samples of f_l^i just outside the disk in which the image is to be reconstructed will reduce the number of points at which extrapolation is necessary at the $(i+1)$-th level, but will in itself require extrapolation unless a sufficient number of points outside the disk were computed at the $(i-1)$-th level. A scheme with the same computational complexity is conceivable in which at each level of grid merging, sample points outside the disk are chosen in such a way that extrapolation is never required, but we did not use such a scheme in our implementation.

In our implementation, we used one more than the

$$\lceil N\sin((2^i-1)\pi/2Q))\rceil$$

points per line in the slow direction prescribed in subsection 4.3 so that the first and last sampling points on each line would be at or beyond the boundary of the disk supporting the reconstruction. This does not change the computational complexity.

We have found empirically that image quality can be improved by doubling the number of samples per Radon projection by interpolation, or by doubling the number of projections (i.e., the number of angles at which projections are computed) by interpolation. A cheaper way to improve image quality is to sample f_i^j for all i and j at no fewer than 5 points in the slow direction, although Formula 5 may imply that fewer points are necessary in the slow directions in the first few levels of merges. In any case, neither of these adjustments are necessary when the postprocessing correction method described below is applied.

5 Postprocessing

Images resulting from the multiscale backprojection algortihm as described up to this point are somewhat blurred due to the many interpolations necessary. The point spread function of the algorithm is wider than that of the classical backprojection. In this section, we describe an $O(N^2 \log N)$ Fourier domain correction of the image which greatly reduces the width of the point spread function of the multiscale backprojection and enhances the resulting images. A similar correction can be performed after the classical backprojection, but with less of an improvement, and images produced by the multiscale method with Fourier domain correction are as good qualitatively as those produced using the classical backprojection with Fourier domain correction.

5.1 Basic concept

If the point spread function of a given backprojection method were shift invariant, then the obvious correction would be to divide the Fourier transform of the reconstruction by that of the point spread function and to take the inverse Fourier transform of the result. However, the point spread functions of both the classical and multiscale backprojection methods vary slightly over the image. Our postprocessing correction consists of dividing the Fourier transform of the image by a Gaussian which approximates the Fourier transforms of the point spread functions obtained at various points in the image. The width of the Gaussian is chosen in such a way as to optimize this approximation. As we will show, this technique is more effective for the multiscale backprojection than for the classical one, as the point spread functions obtained for the multiscale method can be more closely approximated by a Gaussian.

5.2 Determination of Gaussian width σ_0

For each of the two methods under consideration (the classical and multiscale backprojection methods), we wished to find the 2-D Gaussian that in some sense fit a selection of point responses better than any other 2-D Gaussian. The 2-D Gaussians are of the form

$$\epsilon^{-(x^2+y^2)/\sigma^2},$$

where the width $\sigma > 0$ of the Gaussian may be chosen arbitrarily. In our context, x and y may be regarded as pixel coordinates (i.e., the x coordinates of horizontally adjacent pixels differ by 1, and the y coordinates of vertically adjacent pixels differ by 1), and the goal was to choose the value σ_0 of σ for which the resulting Gaussian best fit a selection of point responses in the following sense.

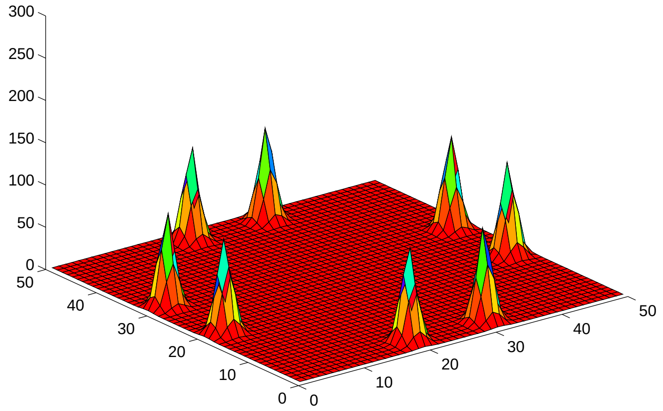

Figure 5: Eight-fold symmetry of the operations.
A point response of the Radon transform-CB sequence is identical, up to rotation and reflection, to as many as seven other point responses.

The point responses were all represented by real-valued matrices. From each such matrix, the 7×7 submatrix whith the peak of the point response at its center was extracted. Let A be the normalized sum of these submatrices. Let the indices i and j of A run from -3 to 3. Let

$$m_{ij}(\sigma) = \epsilon^{-(i^2+j^2)/\sigma^2}.$$

For each method, σ_0 is defined as the value of σ which minimizes

$$\sum_{i,j=-3}^{3} (A_{ij} - m_{ij}(\sigma))^2.$$

5.3 Selection of point responses for computation of σ_0

We assume that backprojection is performed on a square grid of pixels, or on a circular central region of this square. The computation of the Radon projections of the image to be represented by this grid, and the convolution and backprojection of the Radon projections onto the grid, can be perfomed in such a way that the entire sequence of operations is symmetric with regard to the horizontal and vertical axes of the grid and its two diagonals. In such a case, a point response anywhere in the grid is identical, up to rotation and reflection, to as many as seven other point responses, whose positions are obtained from the position of the original point response by reflection across one or more of the axes of symmetry of the operations. See Figure 5. It follows that when selecting positions on the grid at which to compute point responses for the purpose of computing σ_0, one may restrict the positions to a single octant of the grid and compute the point responses at those positions, and then the point responses at the corresponding positions in the other seven octants can be obtained by rotating and reflecting the computed point responses.

In our research, point responses were computed at 15 randomly selected points in a single octant, and the point responses at the corresponding points in the remaining octants were obtained by rotation and reflection of the original 15 point responses, as explained above.

Point responses and the resulting value of σ_0 depend on the process by which the Radon projections are obtained. In our research, Radon projection data were computed in software, but an apparatus implementing the method described here might obtain the projection data from physical measurements. In any case, for any such apparatus, σ_0 should be computed from point responses generated by Radon data obtained from the source from which the apparatus will obtain the Radon data in practice. The eight-fold symmetry exploited for the purpose of our research may not hold for all systems.

5.4 Results

The figures on Page A indicate the optimal value σ_0 of σ, and the degree to which the resulting Gaussian fits the point responses, for the classical backprojections and several variants of the multiscale backprojection. Over each point response grayscale image is displayed a pair of numbers. The first is the optimal value σ_0 of σ for the method, and the second is the error ε given by

$$\varepsilon = \max_{i,j=-1,\ldots,1} |A_{ij} - m_{ij}(\sigma_0)|.$$

It can be seen that while σ_0 is greater for the multiscale backprojection than for the classical backprojection, implying that multiscale backprojection without postprocessing has a wider point response and therefore produces blurrier images, the multiscale method point responses can be approximated better by a Gaussian, and therefore the multiscale backprojection with postprocessing can produce sharper images than the classical backprojection with postprocessing.

On Page B are shown surface plots of point spread functions for the classical and multiscale backprojection methods and surface plots of their Fourier tranforms. It can be seen from these figures that although the uncorrected multiscale backprojection produces a wider point spread than the classical method, the Fourier transform of the multiscale point spread function more closely resembles a Gaussian than the Fourier transform of the classical point spread function. It follows that the multiscale method lends itself to postprocessing correction described here than does the classical method.

Page C shows an image from which Radon data were computed for the purpose of testing different variations of CB. Page D shows the results of CB with classical backprojection. Page E shows the results of CB with multiscale backprojection without postprocessing. Page F shows the results of CB with multiscale backprojection, without postprocessing but with doubling of the data by means of interpolation. Page G shows the results of CB with multiscale backprojection, without doubling of the data but with postprocessing.

References

[1] S. Alliney, S. Matej, I. Bajla, "On the possibility of direct Fourier reconstruction from divergent-beam projections," *IEEE Transactions on Medical Imaging*, vol. 12, no. 2, June 1993, pp. 173–181.

[2] A. Brandt, J. Dym, “Fast Calculation of Multiple Line Integrals,” in preparation.

[3] S.R. Deans, *The Radon Transform and Some of its Applications.* New York: John Wiley and Sons, 1983.

[4] M.D. Desai, W.K. Jenkins, “Convolution backprojection image reconstruction for spotlight mode synthetic aperture radar,” *IEEE Transactions on Image Processing*, vol. 1, no. 4, Oct. 1992, pp. 505–517.

[5] H. Fan, J. Sanz, “Comments on ‘Direct Fourier reconstruction in computer tomography,’ ”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, pp. 446–449, April 1985.

[6] G. Herman, *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography.* New York: Academic Press, 1980.

[7] S. Matej, I. Bajla, “A high-speed reconstruction from projections using direct Fourier method with optimized parameters — an experimental analysis,” *IEEE Transactions on Medical Imaging*, vol. 9, no. 4, Dec. 1990, pp. 421–429.

[8] R. Mersereau and A. Oppenheim, “Digital reconstruction of multidimensional signals from their projections,” *Proceedings of the IEEE*, vol. 62, no. 10, Oct. 1974, pp. 1319–1338.

[9] D.C. Munson, Jr., J.D. O'Brien, W.K. Jenkins, “A tomographic formulation of spotlight-mode synthetic aperture radar,” *Proceedings of the IEEE*, vol. 71, no. 8, Aug. 1983, pp. 917–925.

[10] A.V. Oppenheim, R.W. Schafer, *Digital Signal Processing.* Englewood Cliffs: Prentice-Hall, 1975.

[11] S.X. Pan, A. Kak, “A computational study of reconstruction algorithms for diffraction tomography: interpolation versus filtered backpropagation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, no. 5., Oct. 1983, pp. 1262–1275.

[12] H. Peng, H. Stark, “Direct Fourier reconstruction in fan-beam tomography,” *IEEE Transactions on Medical Imaging*, vol MI-6, no. 3, Sept. 1987, pp. 209–219.

[13] H. Stark, M. Wengrovitz, “Comments and corrections on the use of polar sampling theorems in CT,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, pp. 1329–1331, Oct. 1983.

[14] H. Stark, J.W. Woods, “Authors' reply to ‘Comments on “Direct fourier reconstruction in computer tomography” ’,” *IEEE IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 2, April 1986, pp. 379–380.

[15] H. Stark, J. Woods, I. Paul, R. Hingorani, “Direct Fourier reconstruction in computer tomography,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, pp. 237–245, Apr. 1981.