Discrete Radon transform has an exact, fast inverse and generalizes to operations other than sums along lines

William H. Press[†]

Los Alamos National Laboratory, Los Alamos, NM 87545

Contributed by William H. Press, October 18, 2006 (sent for review September 13, 2006)

Götz, Druckmüller, and, independently, Brady have defined a discrete Radon transform (DRT) that sums an image's pixel values along a set of aptly chosen discrete lines, complete in slope and intercept. The transform is fast, $O(N^2 \log N)$ for an $N \times N$ image; it uses only addition, not multiplication or interpolation, and it admits a fast, exact algorithm for the adjoint operation, namely backprojection. This paper shows that the transform additionally has a fast, exact (although iterative) inverse. The inverse reproduces to machine accuracy the pixel-by-pixel values of the original image from its DRT, without artifacts or a finite point-spread function. Fourier or fast Fourier transform methods are not used. The inverse can also be calculated from sampled sinograms and is well conditioned in the presence of noise. Also introduced are generalizations of the DRT that combine pixel values along lines by operations other than addition. For example, there is a fast transform that calculates median values along all discrete lines and is able to detect linear features at low signal-to-noise ratios in the presence of pointlike clutter features of arbitrarily large amplitude.

backprojection | computerized tomography | inverse methods

The Radon transform (RT) of a two-dimensional function $f(x, y)$ with compact support that includes the origin is familiar as the set of projections along angles θ, $0 \le \theta < \pi$,

$$\mathcal{R}f \equiv p(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$

$$= \int_{-\infty}^{\infty} f(\rho\cos\theta - \ell\sin\theta,\ \rho\sin\theta + \ell\cos\theta)d\ell \qquad [1]$$

where the δ-function converts the two-dimensional integral to a line integral $d\ell$ along the line $x\cos\theta + y\sin\theta = \rho$. The transformed function $p(\rho, \theta)$ is often referred to as the sinogram of $f(x, y)$, because a point δ-function in f transforms to a sinusoidal line δ-function in p.

Although occasionally useful and certainly interesting in its own right, the RT pales in practical importance by comparison with its inverse, $\mathcal{R}^{-1}p$, which recovers $f(x, y)$ from its projections. The inverse RT and its approximations enable computerized tomography and related medical and other imaging technologies. That the inverse exists (for suitably well behaved functions f) follows immediately from the Fourier Slice Theorem, which says that the 1D Fourier transform of a projection at angle θ has values identical to a radial slice through the origin of the 2D Fourier transform of the original image. The proof is straightforward (e.g., ref. 1). In the continuous case, then, the 2D Fourier transform of f is recovered in polar coordinates from the slices, and an inverse 2D Fourier transform recovers f.

Practical applications inevitably deal with images $f(x, y)$ and sinograms $p(\rho, \theta)$ that are represented discretely, usually as 2D arrays of values. There is a large, if scattered, literature concerning approximations of the continuous RT and its inverse in such cases. Standard treatments include refs. 1–3. Some algorithms have the characteristic of being not only approximations of the continuous case, but also interesting discrete transforms in their own right, for example, refs. 4–9. Collectively these transforms are known as discrete RTs (DRTs). However, no single algorithm has successfully laid claim to being "the" DRT.

A key issue in DRT algorithmics is whether an algorithm and/or its inverse is fast in the sense of achievable for an $N \times N$ image in $O[N^2 (\log N)^q]$ operations, for some small integer q (ideally 1). Fast DRT algorithms are almost always based on a discretization of the Fourier slice properties of the continuous case, because the fast Fourier transform (FFT) approximates the 1D continuous transform in $O(N \log N)$ operations and the 2D transform in $O(N^2 \log N)$. Or, as in ref. 4, fast algorithms may derive from using the FFT for the fast solution of special sets of linear equations, such as block-circulant forms. There are also "slow" DRTs and inverses, with $O(N^3)$ and larger operations count, but they are of little practical interest.

Another key issue concerning DRTs is their accuracy. This issue can be framed in various ways. A somewhat argumentative framing is this: Given your choice for a discrete representation of $f(x, y)$ as $O(N^2)$ values and your choice of a DRT algorithm approximating a continuous RT, is there a fast inverse DRT that exactly reproduces your discrete representation of f from the values of its DRT? This formulation does not even begin to address the issue of how accurately f is captured by its discrete representation in the first place or how accurately the DRT approximates a continuous RT. Nevertheless, there seems to be no published DRT that can answer unequivocally "yes" to the question. Those that come closest make special assumptions about $f(x, y)$, for example, periodicity along one axis or severely bandwidth-limited. Fast inverse algorithms that are sometimes termed "accurate" are generally so in the sense that a round-trip (sampled image to DRT to reconstructed image) has a point-spread function that is highly concentrated in a few pixels and with tolerably small tails outside those few.

The situation is not materially different even if we allow iterative inverse algorithms, where fast now can mean $O[N^2(\log N)^q \log\varepsilon]$, for small integer q, where ε is the desired rms accuracy, which can be made arbitrarily small. (This is called linear convergence by numerical analysts and exponential convergence colloquially.)

The principal result of the present study is to show that a particular DRT, proposed independently by Götz and Druckmüller (6) and by Brady (7) (and related to ref. 10) has an exact inverse

Author contributions: W.H.P. designed research, performed research, and wrote the paper.

The author declares no conflict of interest.

Freely available online through the PNAS open access option.

Abbreviations: RT, Radon transform; d-line, digital line; DRT, discrete RT; DMRT, discrete median RT; FFT, fast Fourier transform.

[†]E-mail: wpress@lanl.gov.

This article contains supporting information online at www.pnas.org/cgi/content/full/0609228103/DC1.

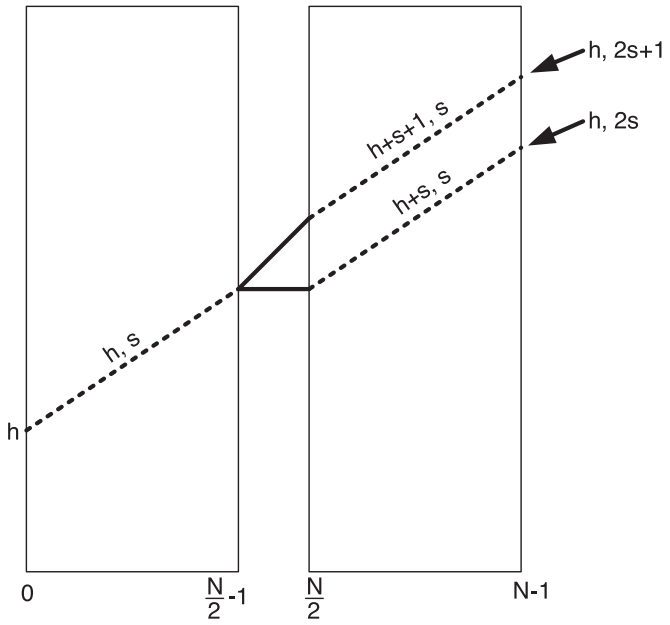

Fig. 1. d-lines of width N are defined recursively in terms of d-lines of width $N/2$. The integers h and s parameterize the intercept and rise of each d-line.

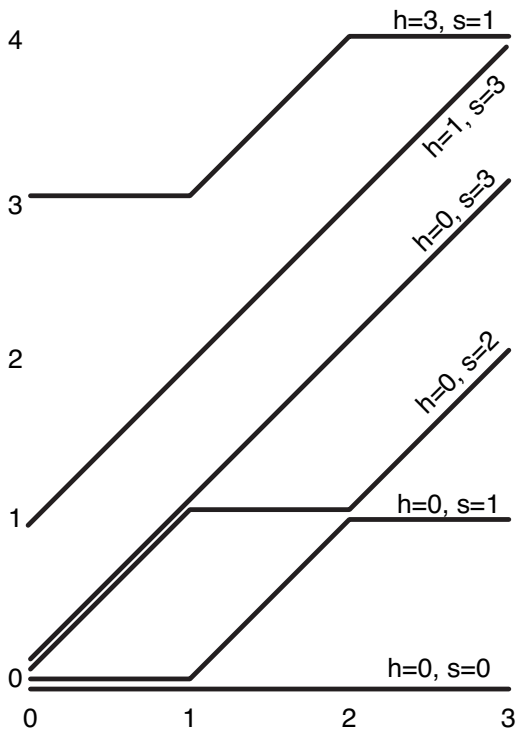

Fig. 2. Examples of d-lines for the case $n = 4$. The small vertical offsets are for clarity only; all d-lines pass exactly through integer lattice points, one in each column.

achievable by a (iteratively) fast algorithm. By calling the inverse exact, we mean that the inverse is capable of recovering with arbitrary precision the pixel-by-pixel values of the input image, with no spreading or other artifacts. We give a specific algorithm exhibiting $q = 3$ and suggest that $q = 2$ is likely achievable. Our inverse does not use Fourier methods or FFTs. Other interesting properties of the Götz–Druckmüller–Brady DRT, which may lead to new applications, are obtained in the present study and are described below.

Having a pixel-exact, fast-transform pair, DRT and its inverse, does not necessarily advance the practical art of computerized tomography. However, it does allow for a clearer distinction between the accuracy of the DRT inversion itself and the various approximations that may be made in mapping physical data to or from the DRT.

The Götz–Druckmüller–Brady DRT

Discrete Approximations of Lines. For definiteness, consider an image represented as an $N \times N$ array of intensity values f_{ji}, with $0 \le i, j < N$, and N an integer power of 2. Following ref. 6, we define a set of digital lines (d-lines) that transect the image, passing exactly through one array point in each column of the array and parameterized by integers h and s. The d-line $D_N(h, s)$ connects the array point $(0, h)$ (meaning, by convention, $i = 0$ and $j =$ h) and the array point $(N - 1, h + s)$. We refer to h as the intercept and s as the rise of the d-line. We consider for now only the case $0 \le s \le N - 1$, corresponding to slopes from 0° to 45°, inclusive. d-lines are defined recursively in terms of d-lines on images half as wide:

$$D_N(h, 2s) = D_{N/2}(h, s) \cup D_{N/2}(h + s, s)$$
$$D_N(h, 2s + 1) = D_{N/2}(h, s) \cup D_{N/2}(h + s + 1, s) \quad [2]$$

where $\cup$ indicates joining left and right halves. Fig. 1 illustrates the recursion, and Fig. 2 shows some d-lines for the case $N = 4$.

The d-line $D_N(h, s)$ approximates a continuous line with intercept h and slope $s/(N - 1)$. As suggested in ref. 7, its maximum vertical deviation from the continuous line is $\le (1/6)\log_2 N$.

DRT in One Quadrant. We define the DRT as simply the sum of image intensities over array points on a d-line:

$$R^a(h, s) = \sum_{(i,j)\in D_N(h,s)} f_{ji} \quad [3]$$

with the convention that $f_{ji} = 0$ if i or j is outside of the range $[0, N - 1]$. The superscript a is used to indicate the "quadrant" from 0° to 45°. (We define b, c, and d quadrant transforms below.) One sees immediately that the recursive definition of the d-lines induces a recursive calculation of the DRT components simply by associating each partial d-line with its corresponding partial sum of function values. This is illustrated in Fig. 3, which shows how two half-images of partial sums are converted to one full image in a single upward sweep. As shown, one row of scratch space is used. Actually, if it mattered, the sweep could be done completely in place by the use of a bit-reversal technique similar to that of the FFT (6). The transform $R^a(h, s)$ of an image is calculated by performing $\log_2 N$ sweeps, first combining adjacent pairs of columns, then pairs of pairs, and so forth. An exact representation of the algorithm is in *SI Text*.

Notice that h, the intercept, takes on some negative values so as to include all d-lines that intersect the partial images or the final full image. When sweeping to produce a partial image of width n, the most negative value of h is $-n + 1$, because a 45° d-line with that intercept will intersect just the single pixel in the lower right of the partial image. Specifically, sums are performed for i and j on each partial image satisfying all of

$$0 \le i \le n - 1$$
$$-n + 1 \le j \le N - 1 \quad [4]$$
$$0 \le i + j$$

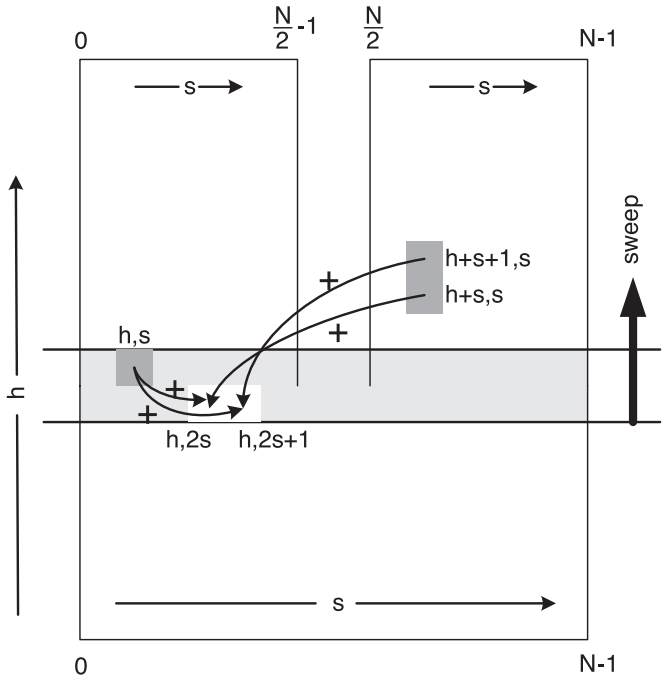

Fig. 3. Partial sums on two half-images are replaced by partial sums on a full image in a single upward sweep. The DRT is computed by performing this process first for pairs of columns in an image, then for pairs of pairs, and so forth in $\log_2 N$ sweeps.

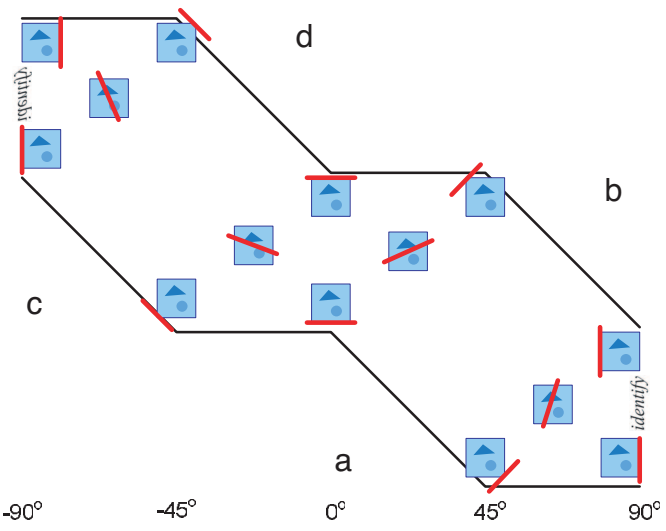

Fig. 4. DRTs calculated in four quadrants by Eq. **5** stitch together as shown to give the global DRT. In each quadrant, the parameter h varies along the vertical direction and the parameter s varies along the horizontal. Red lines indicate schematically the sums through the original image (blue) corresponding to locations in the DRT. Left and right edges are identified with a twist: a Möbius strip.

for a total of $nN + (1/2)n(n-1)$ in each partial image. Summing the partial images in each sweep and the $\log_2 N$ sweeps gives a total of $(N^2 - N/2)\log_2 n + N(N-1) = O(N^2 \log_2 N)$ additional operations. Note that there are no multiplications or other floating operations, a point to which we return below.

Remaining Quadrants and Global Topology. Again, following refs. 6 and 7, we define the DRT for other angle ranges by appropriately flipping the original image, then repeating exactly the same algorithm as for quadrant a:

$$\begin{aligned} R^b\{f_{ji}\} &= R^a\{f_{ij}\} && (45° \text{ to } 90°) \\ R^c\{f_{ji}\} &= R^a\{f_{i,N-1-j}\} && (-90° \text{ to } -45°) \\ R^d\{f_{ji}\} &= R^a\{f_{N-1-j,i}\} && (-45° \text{ to } 0°) \end{aligned} \qquad [5]$$

Fig. 4 shows how the four quadrants stitch together to form the global DRT. In each quadrant, the intercept h varies in the vertical direction, whereas the rise s varies horizontally. In quadrants a and c, both parameters increase from the lower left; in b and d, both parameters increase from the upper right. Each quadrant contains $N^2 + (1/2)N(N-1)$ values, a triangle below (or above) a square. The whole DRT therefore contains $6N^2 - 2N$ values. Each point in the DRT corresponds to a line through the original image, shown schematically as red lines through a fixed blue image. The top and bottom boundaries of the DRT correspond to bounding cases where the line only just intersects the image. The left and right edges are identified with a twist, as indicated.

Supporting information (SI) Fig. 7 shows two sample images, and SI Figs. 8 and 9 show their global (i.e., four quadrant) DRTs plotted graphically. The images are $256 \times 256 \times 8$ bits and are, respectively, a standard photographic test image and a section of Zubal's standard head phantom (14).

Relationship Between DRT and Sinogram. Although the DRT is defined from a discrete image, as above, one may also identify it with values sampled from the continuous sinogram of a continuous image. Details are given in *SI Text*. SI Fig. 10 shows a $256 \times 256 \times 8$ bit sinogram of the standard Shepp–Logan head phantom, captured from the web as a compressed GIF image, and its nearly perfect reconstruction by the methods of this paper.

Inverse DRT

Backprojection is Fast and Exact. For the continuous RT, backprojection is the adjoint operator to the transform,

$$\mathcal{R}^* p \equiv b(x,y) = \int_{-\infty}^{\infty} \int_0^{\pi} p(\rho, \theta)\delta(x\cos\theta + y\sin\theta - \rho) d\rho\, d\theta$$

$$= \int_0^{\pi} p(x\cos\theta + y\sin\theta, \theta) d\theta \qquad [6]$$

One sees that it is the integral over all directions of all of the projections that pass through the point (x, y). Although the adjoint operator is not the same as the inverse, we will see that its discrete analog is a useful building block.

Just as the recursive definition of d-lines (Eq. **2** and Fig. 1) induced a fast DRT, it induces a fast backprojection algorithm (7, 11). The process essentially reverses the sweep shown in Fig. 3: Every element of left- and right-half partial images is assigned the sum of two full-image elements, in correspondence with the d-line recurrence. If $B_n(h, s)$ denotes a partial backtransformation on an image of width, n, then we have,

$$B^L_{n/2}(h,s) = B_n(h,2s) + B_n(h,2s+1) \qquad [7]$$

$$B^R_{n/2}(h+s,s) = B_n(h,2s) + B_n(h-1,2s+1)$$

where superscript L and R refer to the left and right half-width images. The backprojection for each quadrant is obtained in $\log_2 N$ sweeps terminating with $n = 1$. Each sweep can be done in place, or with a small amount of scratch memory. The full backprojection is an appropriate sum over quadrants,

$$B_{ji} = \frac{1}{4(N-1)} \left(B^a_{ji} + B^b_{ij} + B^c_{i,\,N-1-j} + B^d_{N-1-j,i}\right) \qquad [8]$$

where the normalization constant shown will be convenient later. As should be clear, this calculation is exact. That is, the result is exactly the sum of the discrete projections along d-lines that pass through an image pixel. The *SI Text* has an exact statement of the algorithm represented by Eqs. **7** and **8**.

"Natural" Approximate Fourier Inversion. There is a natural approximate inverse of the DRT as we have defined it that is obtainable by Fourier methods. By natural, we mean that no interpolation is required in getting into and out of Fourier space, nor necessarily any explicit application of a ramp or other (e.g., Shepp–Logan) filter.

Because the main point of this paper is to give an exact inverse that does not use Fourier methods, we relegate discussion of the approximate Fourier inverse to *SI Text*. SI Fig. 11 shows the approximate Fourier inverses of the sample images shown in SI Fig. 7 without any additional filtering. With additional filtering, the visual quality of these images could be substantially improved. However, this would lose information and merely be moving us toward existing, approximate, techniques (1, 2, 3).

Also discussed in *SI Text* is an exact, but ill conditioned, formal inverse. SI Fig. 12 shows how the growth of instabilities render such an inverse useless. Needed is not just an inverse, but a well conditioned inverse.

Key Ideas for a Fast, Exact Inverse. We get a fast, exact inverse by combining three ideas, each one well known in other contexts. The first idea is to use the fast (forward) DRT for the iterative improvement of an approximate inverse. Suppose, in general, we want to solve the linear system $\mathbf{Ax} = \mathbf{b}$ and that $\mathbf{B}_0$ is an approximate inverse to $\mathbf{A}$ in that the matrix

$$\mathbf{R} \equiv 1 - \mathbf{B}_0\mathbf{A} \qquad [9]$$

is small (in a sense we discuss below). Then one easily shows (section 2.5 in ref. 12) that

$$\mathbf{B}_n \equiv (1 + \mathbf{R} + \mathbf{R}^2 + \cdots + \mathbf{R}^n)\mathbf{B}_0 \qquad [10]$$

becomes $\mathbf{A}^{-1}$ as $n \to \infty$ if the series converges. It follows from this that the recurrence,

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{B}_0(\mathbf{b} - \mathbf{A}\mathbf{x}_n) \qquad [11]$$

converges to the solution $\mathbf{x}$ from all starting values. Note that Eq. **11** requires only the application of the (exact) forward operator $\mathbf{A}$ and the approximate inverse operator $\mathbf{B}_0$. At each step, we calculate a correction by approximately inverting the residual of the previous step. Eq. **10** already shows the sense in which $\mathbf{R}$ must be small: All of its eigenvalues must have magnitudes ≤ 1, or else the series will diverge. This is by no means easy to achieve for complicated operators in large-dimensional spaces. For example, the natural, approximate Fourier DRT inverse mentioned above does not satisfy this condition and cannot be iterated.

The second idea is to high-pass filter the (exact, fast) backprojection by a local convolution with the goal of reproducing only the highest frequency information in the image to tolerable accuracy. By highest frequency, we mean frequencies between 1/2 and 1 times the Nyquist frequency. We will see below what tolerable accuracy means.

For example, the centered nine-point filter with coefficients,

$$\mathcal{H} = \begin{pmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{3}{4} & -\frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{pmatrix} \qquad [12]$$

(and the obvious reflections at edges and corners), has unity response both to a binary checkerboard (Nyquist frequency in both directions) and binary stripes (Nyquist frequency in one direction) and zero response to a constant. The intent is to use backprojection, filtered in this way, not on the actual image but only on a residual (compare with Eq. **11**) that has missing or inaccurate high-frequency information. The filtering requires $O(N^2)$ operations, as compared with $O(N^2 \log_2 N)$ for the backprojection.

The third idea is the full multigrid method, usually called FMG (13, 12), and the related concepts of restriction and prolongation operators. A restriction operator is, here, a linear map from a DRT of size N, denoted $R^{a..d}_N(h, s)$ to one of size $N/2$, with the idea that the smaller DRT should approximate the larger one. A prolongation operator is a map on images (not DRTs) from size $N/2$ to size N. There are many such operators. With foresight, we use these specific ones:

$$R^{a...d}_{N/2}(h, s) = \frac{1}{4}\left[R^{a...d}_N(2h, 2s) + R^{a...d}_N(2h+1, 2s)\right] \text{ (restriction } \mathcal{S})$$

$$f^N_{ji} = f^N_{j+1,\,i} = f^N_{j,i+1} = f^N_{j+1,i+1} = f^{N/2}_{\lfloor j/2 \rfloor, \lfloor i/2 \rfloor}, \text{ (prolongation } \mathcal{P})$$

[13]

where $a..d$ denotes any of the values a, b, c, or d. Notice that $\mathcal{S}$ averages adjacent values of h, but samples only even values of s. This choice works, whereas other seemingly equally valid choices don't work, as we shall discuss. Also to be noted is that the output of $\mathcal{S}$ is not necessarily the exact DRT of any image; it will prove useful nevertheless.

Details of the Fast, Exact Inverse. Define an algorithm $\mathcal{B}$ that constructs an approximate inverse f^N_{fi} of a DRT $R^{a..d}_N$ as follows:

1. Apply $\mathcal{S}$ to $R^{a..d}_N$, giving $R^{a..d}_{N/2}$.
2. Recursively call $\mathcal{B}(R^{a..d}_{N/2})$ (this algorithm), giving $f^{N/2}$.
3. Apply $\mathcal{P}$ to $f^{N/2}$, giving $f^{N\prime}$, a low-frequency approximation to f^N.
4. Calculate the forward DRT of $f^{N\prime}$ and subtract it from the original $R^{a..d}_N$, giving a residual DRT.
5. Backproject and locally high-pass filter the residual DRT, giving an image correction.
6. Add the image correction to $f^{N\prime}$, producing f^N.

SI Fig. 13 shows the result of applying the approximate inverse operator $\mathcal{B}$ to the DRTs shown in SI Figs. 8 and 9.

Note that, recursion and all, $\mathcal{B}(R^{a..d}_N)$ is a strictly linear operator on $R^{a..d}_N$. In fact, it can be used as $\mathbf{B}_0$ in Eq. **11**, but only if the series Eq. **10** converges. To prove convergence, we must show that the residual matrix $\mathbf{R}$ has eigenvalues that are <1 in magnitude. Because of $\mathcal{B}$'s complicated, recursive algorithmic description, it seems hopeless to do this analytically, so we must resort to numerical experiment.

Convergence Tests. We now show that the specific choices given for the high-pass filter (Eq. **12**) and prolongation and restriction operators (Eq. **13**) do produce a convergent series. There is art in those specific choices. As remarked above, other choices that might seem equally valid *a priori* yield divergent results. On the

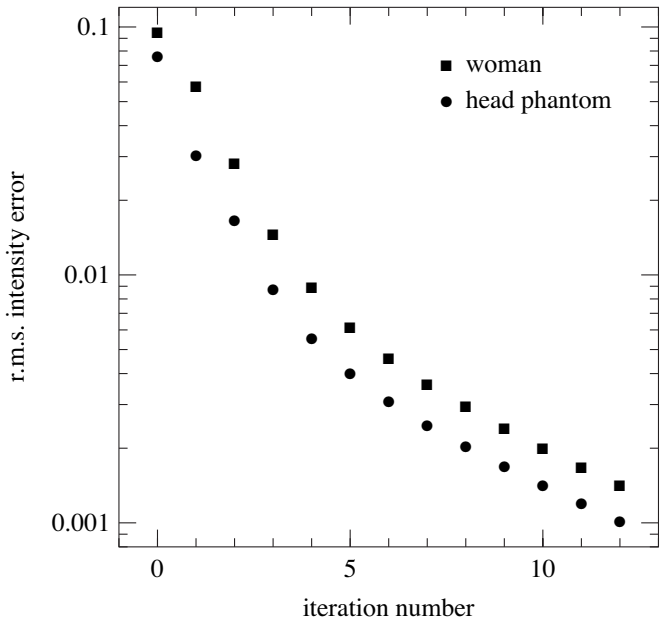

Fig. 5. Error versus iteration number for the two sample images in SI Fig. 7. Iteration 0 is the error of the approximate inverse $\mathcal{B}$; subsequent iterations are improved by reapplying $\mathcal{B}$ to the residual DFTs. Errors are normalized to a grayscale range of (0, 1).

other hand, the specific choices shown are known to be not optimal (see *Discussion*) and can be improved.

Statistically, an image whose pixels are identically and independently distributed normal deviates will have a significant projection into all eigenmodes. We take the DRT of such images, then iterate Eq. **11** and verify convergence (in L_2 norm) to the original image. Initial convergence is faster than a single exponential, because it is dominated by large numbers of modes with a wide range of small eigenvalues. As iteration number n increases, the residual error is found to approach a single exponential, characteristic of the largest eigenvalue of **R**. Multiple trials over a range of image sizes from $n = 16$ to $n = 1024$ are accurately summarized, asymptotically, by the empirical relation

$$\Delta \ln \text{error (per iteration)} \approx -\frac{13.8}{(\log_2 N)^2} \qquad [14]$$

The workload to converge to accuracy ε thus scales as $O[N^2(\log N)^3 \log \varepsilon]$.

In practical work, asymptotic convergence rates may not matter that much. For all sizes up to $n = 2{,}048$, realistic images are found to converge to useful accuracies (e.g., indistinguishable by eye from the exact answer) in a modest number of iterations. Fig. 5 shows how the rms errors decrease with iteration number for the two sample images shown in SI Fig. 7. The leftmost values (plotted as iteration 0) are the errors relative to a grayscale range (0, 1), after the initial application of the approximate inverse operator $\mathcal{B}$. Iterations 1, 2, . . . show the result of iterative improvement. The asymptotic approach to a single dominant exponential is clearly seen.

SI Fig. 14 shows the inverses obtained from the DRTs in SI Figs. 8 and 9 after only four iterations. After a few more iterations, the inverses become perceptually indistinguishable from the original images in SI Fig. 7. Further iterations converge to arbitrarily small rms errors.

The DRT Need Not Use Ordinary Addition

We already noted that the DRT defined above uses only addition, and no other arithmetic operations, in combining data values along a d-line. A consequence is that any associative and commutative operation can serve instead of addition, yielding a number of interesting generalizations of the DRT. The DRT algorithm "presents" data elements (and their partial "sums") to us in an arbitrary order; we can combine these by using any rule with the semantics of addition that we want.

One example is a discrete median RT (DMRT), for which the transform output is the median of all image values along a d-line. Here the combination rule is to update an approximate representation of the cumulative distribution function (see, e.g., ref. 15) and finally output the median quantile. The combination rule can be viewed as an associative and commutative operator on the data values. In fact, we can use exactly the same code as for the standard DRT, simply overloading the addition operator with machinery that combines distribution functions.

Because the median along a line is a robust estimator of that line's central value but is insensitive to large fluctuations in the tails, the DMRT is good at finding low signal-to-noise straight lines in the presence of much larger pointlike clutter. Such lines appear as clusters of unusually large values in the transform space. Detection can be accomplished by identifying transform values above some threshold or above some threshold of statistical significance (adjusting for the variable number of image pixels intersected by a d-line). Alternatively, one can invert the thresholded DMRT as if it were an ordinary DRT. The goal is

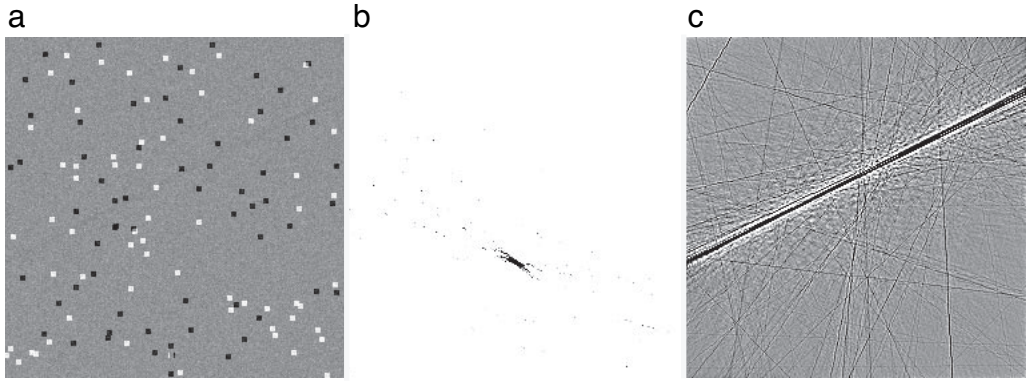

Fig. 6. Example of a DMRT. (*a*) Image with an almost invisible line, 4 pixels wide and 0.5 σ above the noise background. (*b*) Portion of its DMRT, whose values are medians, not sums, along lines. Only the largest 0.1% of transform values are plotted. The line produces a significant feature. (*c*) Inverse of *b*, computed as if it were a DRT, not a DMRT. The location of line feature is now apparent.

not to reconstruct an accurate image but rather to reconstruct in the image space a mask that shows the detected lines.

Fig. 6 shows an example. A synthetic image consists of Gaussian noise pixels, small positive and negative squares of arbitrarily large amplitude, and an almost invisible line that is 4 pixels wide and has values 0.5 standard deviations larger than the background. A portion of the DMRT is shown after it has been thresholded at the 99.9% quantile level. The DRT inverse of this DMRT strongly highlights the line. The small squares appear only as faint background linear features when there are chance alignments among them.

Because the ordinary DRT itself increases the significance of line features (which project to a point) over point features (which smear to a sinusoid), the power of the DMRT may not be intuitive. For the image in Fig. 6, however, ordinary projection is not nearly good enough. SI Fig. 15 shows the same region of transform space as Fig. 6*b* but for the ordinary DRT. One sees that the weak feature of interest is completely lost in a confusion of much stronger, overlapping sinusoids.

SI Fig. 16 shows the result of taking the DRT inverse of the DMRT of a conventional test image of a face without thresholding. Surprisingly, a good deal of detail from the original image is reconstructed from the median values along the lines alone.

Not only the median but also any other property of the distribution function along the lines can be computed in the same way. For example, one could compute several moments of the distribution and thus recognize linear features that have means identical to the background but with different variances or skews.

Discussion

Inverse or Pseudoinverse? The alert reader will already have caught our consistent, but slightly inaccurate, use of the term "inverse." The DRT, as we have defined it, is a linear map from a space of dimension N^2 into one of dimension $6N^2 - 2N$. Because our inverse also is a linear map, it must map (at most) an N^2 dimensional linear subspace of the DRT back into the image space. There must therefore be a another subspace of (at least) dimension $5N^2 - 2N$ that is mapped to zero by the inverse. The fact that numerical experiments on random matrices yield exponential convergence to known original images provides strong evidence that the respective subspace dimensions are as indicated and, moreover, that the maps are well conditioned. We can shed some additional light on the condition number by a different experiment: We put random identically and independently distributed normal deviates $N(0, 1)$ into all $6N^2 - 2N$ components of the DRT, then take the inverse and ask what the resulting rms value of an image pixel is. Summarizing results for a range of N, an empirical relation is

$$\text{rms image pixel} \approx 0.50N^{-0.44} \quad \textbf{[15]}$$

which is never large and decreases as N increases. Finally, we can ask what fraction of the initial variance of the DRT is removed if we subtract the DRT of the inverse image. The result is very close to 5/6, as we expect from counting dimensions and again suggesting well conditioned maps.

These checks are relevant to our use of a restriction operator like $\mathcal{S}$ in Eq. **13**. The smaller DRT obtained by applying $\mathcal{S}$ to a DRT does not exactly correspond to any image. However, we can view it as being the sum of the DRT of a nearby image plus noise. If the approximate inverse were ill conditioned, this imagined decomposition might cause iterative improvement to fail; however, such failure is not observed.

Conjectured Improvements. The appearance of two powers of the logarithm in Eq. **14** is not surprising. It is indicative of errors diffusing in logarithmic scale by a gradient-driven relaxation process. In other such situations (e.g., section 19.5 of ref. 12), the use of techniques such as overrelaxation can change the diffusion rate by one power of the problem size. We therefore conjecture that variants of the methods described here can achieve $O[N^2(\log N)^2 \log\varepsilon]$ workload.

If one examines the residual images at a late stage, one sees that residual errors are dominated by frequencies $\approx 2/3$ of the Nyquist frequency, just where the nine-point filter might be expected to be starting to fail. Even for nine-point filters, a numerical search easily turns up filters different from Eq. **12** that have slightly better convergence rates than Eq. **14** (although none seem to be universal for all N). It is therefore a reasonable conjecture that there should exist 25-point filters (i.e., 5×5) that considerably improve the constant in Eq. **14**. (We would skip 4×4 filters because they can't be centered.)

1. Kak A, Slaney M (1988) *Principles of Computerized Tomographic Imaging* (Inst Electric Electronics Eng, New York).
2. Natterer F, O'Malley R (2001) *The Mathematics of Computerized Tomography* (Cambridge Univ Press, Cambridge, UK).
3. Natterer F, Wübbeling F (2001) *Mathematical Methods in Image Reconstruction* (Soc Ind Appl Math, Philadelphia).
4. Beylkin G (1987) *IEEE Trans ASSP* 35:162–172.
5. Kelley BT, Madisetti VK (1993) *IEEE Trans Image Proc* 2:382–400.
6. Götz WA, Druckmüller HJ (1996) *Pattern Recognition* 29:711–718.
7. Brady ML (1998) *SIAM J Comput* 27:107–119.
8. Boag A, Bresler Y, Michielssen E (2000) *IEEE Trans Image Proc* 9:1573–1582.
9. Brandt A, Mann J, Brodski M, Galun M (1999) *SIAM J Appl Math* 60:437–462.
10. Donoho DL, Huo X (2000) *Proc SPIE* 4119:434–444.
11. Nilsson S (1997) PhD thesis (Linköping University, Linköping, Sweden).
12. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2002) *Numerical Recipes in C++* (Cambridge Univ Press, New York).
13. Hackbusch W (1985) *Multi-Grid Methods and Applications* (Springer, New York).
14. Zubal IG, Harrell CR, Smith EO, Rattner Z, Gindi G, Hoffer PB (1994) *Med Phys* 21:299–302.
15. Chambers JM, James DA, Lambert D, Vander Wiel S (2006) *Stat Sci* 21:433–450.