Fast Slant Stack:
A notion of Radon Transform for Data in a Cartesian Grid
which is Rapidly Computible, Algebraically Exact, Geometrically Faithful and Invertible

A. Averbuch, R.R. Coifman, D.L. Donoho, M. Israeli, J. Waldén

Abstract. We define a notion of Radon Transform for data in an n by n grid. It is based on summation along lines of absolute slope less than 1 (as a function either of x or of y), with values at non-Cartesian locations defined using trigonometric interpolation on a zero-padded grid. The definition is geometrically faithful: the lines exhibit no 'wraparound effects'.

For a special set of lines equispaced in slope (rather than angle), we describe an exact algorithm which uses $O(N \log N)$ flops, where $N = n^2$ is the number of pixels. This relies on a discrete projection-slice theorem relating this Radon transform and what we call the *Pseudopolar Fourier transform.* The Pseudopolar FT evaluates the 2-D Fourier transform on a non-Cartesian pointset, which we call the pseudopolar grid. Fast Pseudopolar FT – the process of rapid exact evaluation of the 2-D Fourier transform at these non-Cartesian grid points – is possible using chirp-Z transforms.

This Radon transform is one-to-one and hence invertible on its range; it is rapidly invertible to any degree of desired accuracy using a preconditioned conjugate gradient solver. Empirically, the numerical conditioning is superb; the singular value spread of the preconditioned Radon transform turns out numerically to be less than 10%, and three iterations of the conjugate gradient solver typically suffice for 6 digit accuracy.

We also describe a 3-D version of the transform.

We review closely related algorithmic work of Mersereau and Oppenheim (1974), Pasciak (1980), Edholm and Herman (1988), Lawton (1988) and Bailey and Swartztrauber (1991) which are precursors of these ideas, but miss the mathematical framework, the geometric faithfulness, and the invertibility.

Key Words: Radon Transform, Projection-slice theorem, Sinc-Interpolation, Slant Stack, Linogram, Concentric-squares grid, Chirp-Z transform, Fractional Fourier Transform, Pseudopolar Fourier transform, Unequally-spaced FFT.

Dedication. In Memory of John W. Tukey 1915-2000.

Acknowledgements. This work was supported by AFOSR MURI95-F49620-96-1-0028, by DARPA BAA 98-04 and by NSF DMS 98–72890 (KDI). We would like to thank Frank Natterer and Leonid Yaroslavsky for helpful discussions. DLD would like to thank the Mortimer and Raymond Sackler Institute of Advanced Studies at Tel Aviv University for hospitality during preparation of this article

1 Introduction

The Radon Transform is a fundamental tool in a wide range of disciplines, including radar imaging, geophysical imaging, nondestructive testing, and medical imaging [10]. In these areas, the Radon transform is typically discussed in continuum terms, as a mapping from functions $f(x, y)$ with $(x, y) \in \mathbf{R}^2$ to a function $Rf(t, \theta)$ with $t \in \mathbf{R}$ and $\theta \in [0, \pi)$, defined by

$$Rf(t, \theta) = \int f(x, y)\delta(t - x\cos(\theta) + y\sin(\theta))dxdy, \tag{1}$$

where δ denotes Dirac δ-function. In words, Rf is the integral of f over the line $\mathcal{L}_{t,\theta}$ defined by $t = x\cos(\theta) + y\sin(\theta)$.

For most modern applications, it is important to have analogs of Rf for arrays of digital data $I = (I(u, v) : -n/2 \leq u, v < n/2)$. This has been the object of attention of many authors over the last twenty years; a very large literature has ensued [4, 6, 7, 20, 21, 18]. Despite many attempts at defining a "digital Radon transform", we believe that there is at present no definition which is both intellectually and practically satisfying.

1.1 Desiderata

To support our assertion, we propose the following desiderata for a notion of digital Radon transform.

[P1] Algebraic Exactness. The transform should be based on a clear definition, not for example merely on principles of analogy to (1), e.g. formulations such as we 'approximate the integral (1) by a sum' introduce vast amounts of arbitrariness and so are too imprecise to constitute a *definition* of a Radon transform for digital data; they should be avoided.

[P2] Geometric Fidelity. The transform should be based on true lines rather than, say, lines which wrap around or are otherwise non-geometric.

[P3] Speed. The transform should admit an algorithm which is order $O(N\log(N))$ flops, where $N = n^2$ is the number of data in the array I.

[P4] Invertibility. The transform should be one-to-one, and so be invertible on its range; there should be a fast reconstruction algorithm.

[P5] Parallels with Continuum Theory. The transform should obey relations which parallel those of the continuum theory, for example relations with the Fourier transform, and intertwining with differential operators.

The many existing contributions to the literature do not offer these properties simultaneously. A complete discussion would take space we do not have, so we content ourselves with three examples, which also help to illustrate the meaning of our desiderata above.

- Fourier Approaches. Some authors [14, 20] have attempted to exploit the projection-slice theorem, which says that the 1-dimensional constant-θ slice of the Radon transform $((Rf)(t,\theta) : -\infty < t < \infty)$ and the 1-dimensional radial slice of the Fourier transform $(\hat{f}(\lambda\cos(\theta), \lambda\sin(\theta) : -\infty < \lambda < \infty)$ make a 1-dimensional Fourier transform pair. In the continuum theory, this says that $Rf(\cdot,\theta)$ can be obtained by (a) performing a 2-d Fourier Transform, (b) obtaining a radial slice of the Fourier transform, and (c) applying a 1-d inverse Fourier transform to the obtained slice. This suggests an algorithm for discrete data, by replacing steps (a) and (c) by fast Fourier transforms for data on 2-d and 1-d Cartesian grids, respectively. However, strictly speaking, this continuum approach is problematic since step (b) is not naturally defined on digital data: the 2-d FFT outputs data in a Cartesian format, while the radial slices of the Fourier domain typically do not intersect the Cartesian grid. Therefore, some sort of interpolation is required, and so the transform is not algebraically exact. Also, even if the transform should turn out to be invertible (which may be very difficult to determine) the transform is typically not invertible by any straightforward algorithm.

- Multiscale Approaches. Other authors [6, 7, 8, 15] have attempted to exploit two-scale relations, which say that if one knows the Radon transform over four dyadic subsquares of a dyadic square these can be combined to obtain the Radon transform over the larger square. This suggests a recursive algorithm, in which the problem is broken up to the problem of computing Radon transforms over squares of smaller sizes which are then recombined. Strictly speaking, however, the driving identity is a fact about the continuum and does not directly apply to digital arrays, so that when this principle is operationalized, the results involve interpolation and other approximations, and end up being quite crude compared to what we have in mind here. Finally, the use of two-scale relations means that summation along lines is approximated by summation along line segments which are not exactly parallel and so the results can lack a certain degree of geometric fidelity.

- Algebraic Approaches. When n is a prime p, the data grid $G = \{(u,v) : 0 \le u,v < n\}$ may be considered as the group Z_p^2, which has very special properties [21]. The "lines" $\{(ka+b \bmod p, kc \bmod p + d) : 0 \le k < p\}$ for appropriate a, b, c, d have a very special structure: pairs of "lines" either do not intersect at all, or intersect in just one point. This property makes it possible to define an algebraically exact Radon transform for integration along "lines" which operates in $O(N\log(N))$ flops and is invertible. However, the "lines" have, for most parameters (a,b,c,d) very little connection with lines of $\mathbf{R}^2$; simple plots of such "lines" reveal that they are scattered point sets roughly equidistributed through the grid G. In effect, the "lines" wrap around (owing to the $mod\, p$ in their definition), which destroys the geometric fidelity of the transform.

In this note, we describe a notion of Radon transform for digital data which has all of our desired properties [P1]-[P5].

The notion we discuss belongs to a fourth stream of Radon research complementing the three streams of research just mentioned (Fourier-based methods, multiscale methods, and Algebraic Approaches) and represents in a certain sense the culmination of that stream. In effect, this fourth stream says that, to really make sense for digital data, the appropriate notions of continuum Radon transform, of discrete 2-D Fourier domain are subtly different than the usual ones.

1.2 Definition of Radon Transform

We think of Radon transform as an object that assigns a numerical value to each member of a family of lines. We parametrize our lines somewhat differently than in the traditional continuum discussion of Radon transforms, using slopes and intercepts rather than angles and offsets. We believe that this parametrization makes good sense in general, but is particularly natural for dealing with data on Cartesian grids. A *basically horizontal* line is a line of the form $y = sx + z$, where the slope $|s| \leq 1$; a *basically vertical* line has the form $x = sy + z$, where the slope $|s| \leq 1$. This separation of lines into two classes will require that throughout what follows, we maintain two separate but related data structures, based on interchange of roles of x and y.

Given an array $I(u, v)$, a slope s with $|s| \leq 1$, and an offset z, we initially define the Radon transform associated with the basically horizontal line $y = sx + z$ via

$$\mathrm{Radon}(\{y = sx + z\}, I) = \sum_u \tilde{I}^1(u, su + z).$$

Thus, we are summing at n values $(u, su + z)$ along the line $y = sx + z$. The values we are summing come not from the original image I, but instead an interpolant $\tilde{I}^1(u, y)$, which takes discrete arguments in the first argument, and continuous arguments in the second argument.

The interpolation "in y only", is performed as follows. Letting $m = 2n$, we define the Dirichlet kernel of order m by

$$D_m(t) = \frac{\sin(\pi t)}{m \sin(\pi t/m)}.$$

We then set

$$\tilde{I}^1(u, y) = \sum_{v=-n/2}^{n/2-1} I(u, v) D_m(y - v).$$

We note that this is an interpolating kernel, so that

$$I(u, v) = \tilde{I}^1(u, v), \qquad -n/2 \leq u, v < n/2.$$

In the case of basically vertical lines, we define the Radon transform similarly, interchanging roles of x and y:

$$\mathrm{Radon}(\{x = sy + z\}, I) = \sum_u \tilde{I}^2(sv + z, v),$$

with the interpolant defined analogously:

$$\tilde{I}^2(x,v) = \sum_{u=-n/2}^{n/2-1} I(u,v)D_m(x-u).$$

It is convenient to also have θ to represent the angle associated to the slope s. This gives a definition, for $\theta \in [-\pi/4, \pi, 4)$,

$$(RI)(t,\theta) = \text{Radon}(\{y = \tan(\theta)x + t\}, I),$$

and, for $\theta \in [\pi/4, 3\pi, 4)$,

$$(RI)(t,\theta) = \text{Radon}(\{x = \text{cotan}(\theta)y + t\}, I).$$

Below we consider only lines having an intercept $-n \le t < n$, and let T_n denote this set of intercept values.

Because the array $I(u,v)$ has only $N = n^2$ entries, we expect N pieces of Radon information to characterize I. We now fix our choice of angles $\theta_\ell^1 = \arctan(2\ell/n)$, $-n/2 \le \ell < n/2$ and $\theta_\ell^2 = \pi/4 + \arctan(2\ell/n)$. These are not equispaced in angle but instead in slope, having $s = 2\ell/n, -n/2 \le \ell < n/2$. For later use, let Θ_n denote this set of angles.

This set of angles has some very special properties. The first sign of this is that basically horizontal lines make an integer vertical displacement as they traverse from right to left of the image. Similarly, the corresponding basically vertical lines exhibit an integer horizontal displacement as they traverse the image from top to bottom. We could well call these angles "grid-friendly".

The object $RI = (RI(t,\theta) : t \in T_n, \theta \in \Theta_n)$, defined with the grid-friendly angles in Θ_n and intercepts in T_n, may be viewed as a result of mapping from the space of n-by-n arrays I to the space of $2n$ by $2n$ arrays RI. In this paper, we will call this mapping the Radon transform R. Evidently our Radon characterization is a factor 4 oversampled.

Because there are so many notions of Radon transform, it seems reasonable to give this one a name to distinguish it. While throughout this paper, we will call the notion just defined the Radon transform, outside the context of this paper, it is best called the *Slant Stack*, for reasons to be made clear in Section 2.1 below.

We also note that it is very important that we have chosen $m = 2n$ in this definition; $m = n$ would also be possible, but would lack geometric fidelity – see Section 7.1 below.

1.3 Simple Examples

We now give two quick examples of this transform in action. In Figure 1, we display the results of transforming an array containing a single nonzero entry in a sea of zeros. The Radon transform follows a broken line.

In Figure 2, we ask: for a given output coefficient (t_0, θ_0) of the Radon transform, which pixels contibuted, and how much? We can find this out by applying the so-called *Radon backprojection* to an array $Y(t,\theta)$ containing a single nonzero entry at $t = t_0$ and $\theta = \theta_0$. The result is an array $X(u,v)$ which shows us how individual pixels in I contribute to create the specific output coefficient. Figure 2 shows that the contributing pixels lie near a straight line. (Remark for the mathematically inclined: Radon Backprojection is the *adjoint* transformation adj R; with $\langle , \rangle$ the inner product on $n \times n$ arrays and $[,]$ the inner product on $m \times m$ arrays, the adjoint equation

$$\langle X, I \rangle = \langle (\text{adj } R)Y, I \rangle = [RI, Y] = (RI)(t_0, \theta_0)$$

justifies the interpretation of X as measuring the contributions by various pixels.)

These two figures illustrate important features of R. First, as Figure 2 shows, it is geometrically faithful; it is based on summation along true lines, as opposed to lines which exhibit wrap-around artifacts or other geometrical distortions. Second, as Figure 1 shows, the transform sends *points* in the spatial domain into *broken lines* in the new Radon domain.

This latter feature is true for a variant of the continuum Radon transform sometimes called the *Slant Stack* in seismics [10, 29]. Suppose we define, for $\theta \in [-\pi/4, \pi/4)$

$$S^1 f(t, \theta) = \int f(x, y)\delta(t - x - y\tan(\theta))dxdy, \tag{2}$$

and for $\theta \in [\pi/4, 3\pi/4)$

$$S^2 f(t, \theta) = \int f(x, y)\delta(t - \text{cotan}(\theta)x - y)dxdy \tag{3}$$

and encapsulate these in a single object Sf defined by

$$Sf(t, \theta) = \begin{cases} S^1 f(t, \theta) & \theta \in [-\pi/4, \pi/4) \\ S^2 f(t, \theta) & \theta \in [\pi/4, 3\pi/4) \end{cases} .$$

Then, if f is a small pointlike 'bump', a display of Sf will look like a broken line, with a break at the transition angle $\theta = \pi/4$.

Another field in which this continuum transform has been (independently) developed is medical tomography, where Sf is called the Linogram [12, 13], in reference to the fact that points map under S into broken lines, whereas in the usual Radon transform, points map into sinusoids; because of this in medical tomography, the usual Radon transform is sometimes called the sinogram. For this reason, our transform could also properly be called the *discrete Linogram*. We will discuss this topic further below.

We remark that, in the continuum model, the Slant stack and the Radon transform contain the same information: for $\theta \in [-\pi/4, \pi/4)$,

$$(Rf)(t \cdot \cos(\theta), \theta) = (Sf)(t, \theta);$$

a similar relationship holds for $\theta \in [\pi/4, 3\pi/4)$

1.4 Properties of Radon Transform

The following properties of our Radon transform R will be proved in later sections. The first is fundamental.

Theorem 1 *(Projection-Slice Theorem) Define the $2-d$ Fourier transform of the array I via:*

$$\hat{I}(\xi_1, \xi_2) = \sum_{u,v} I(u, v) \exp\{-i(u\xi_1 + v\xi_2)\}, \quad \xi_1, \xi_2 \in [-\pi, \pi).$$

For each fixed $\theta \in [-\pi/4, \pi/4)$, the $2n$ numbers

$$(RI)(t, \theta) \qquad -n \leq t < n,$$

are a 1-d discrete Fourier transform pair with the $2n$ numbers

$$\hat{I}(\pi \cdot \frac{k}{n} \cdot \tan(\theta), \pi \cdot \frac{k}{n}) \qquad -n \leq k < n.$$

For each fixed $\theta \in [\pi/4, 3\pi/4)$, the $2n$ numbers

$$(RI)(t, \theta) \qquad -n \leq t < n,$$

are a 1-d discrete Fourier transform pair with the $2n$ numbers

$$\hat{I}(\pi \cdot \frac{k}{n}, \pi \cdot \frac{k}{n} \cdot cotan(\theta)) \qquad -n \leq k < n.$$

It follows from the projection-slice theorem that one can obtain simultaneously a number of values of R starting from Fourier domain information. If one wants R at all values $t \in T_n, \theta \in \Theta_n$, one needs to know the Fourier transform $\hat{I}$ at all values in a certain non-Cartesian pointset, which we call the Pseudopolar grid. This grid is illustrated in Figure 5 for the case $n = 8$.

This is not the usual Cartesian grid for which the fast Fourier transform is so well-known. However, known but apparently not very well-known ideas provide an FFT for this grid, operating in $O(N \log(N))$ flops, which we will describe below in the section labelled 'Fast Pseudopolar FT'. As a result, we have

Theorem 2 *The $4n^2$ values*

$$\{RI(t, \theta) : t \in T_n, \theta \in \Theta_n\}$$

can be calculated in order $O(N \log(N))$ flops, where $N = n^2$ is the number of samples in the array I.

Moreover, using appropriate arguments in the Pseudopolar Fourier domain, the transform is invertible:

Theorem 3 *Let $\mathcal{I}$ denote the vector space of n-by-n arrays and $\mathcal{R}$ denote the vector space of $2n$-by-$2n$ arrays. The transform $R : \mathcal{I} \mapsto \mathcal{R}$ is one-to-one. There is a bounded operator $R^\dagger : \mathcal{R} \mapsto \mathcal{I}$ so that $R^\dagger R = Id$.*

Although we do not know a fast exact algorithm for the inverse transform, we do know a fast iterative approximation algorithm. This is based on two ingredients. First, a fast exact algorithm for the adjoint transform adj R.

Theorem 4 *The adjoint mapping adj $R : \mathcal{R} \mapsto \mathcal{I}$ taking $2n$ by $2n$ arrays into n-by-n arrays can be computed in order $O(N \log(N))$ flops, where $N = n^2$ is the number of samples in the array I.*

This fast algorithm relies on deployment of the fast Pseudopolar Fourier transform.

The second ingredient is a simple useful preconditioner, again Fourier-based.

Theorem 5 *Define a convolution operator Δ which acts one-dimensionally on each constant θ-slice of the array*

$$\{RI(t,\theta) : -n \leq t < n\}$$

giving

$$\widetilde{RI}(t,\theta) = \sum_u RI(u,\theta)\Delta_{t-u};$$

the operator is characterized by a frequency-domain representation

$$\hat{\Delta}_k = \begin{cases} \sqrt{|k|/2}/n & k \neq 0 \\ \sqrt{1/8}/\sqrt{n} & k = 0 \end{cases}$$

The resulting array

$$\{\widetilde{RI}(t,\theta) : -n \leq t < n, \theta \in \Theta_n\}$$

is a near-isometry with I; the mapping $\tilde{R} : \mathcal{I} \to \mathcal{R}$ has n^2 nonzero singular values with bounded ratios.

In this result, the behavior of the preconditioner weights at $k \neq 0$ can be motivated geometrically; the behavior at $k = 0$, while crucial, is not likely to have a geometric explanation.

Because of its Fourier domain representation, the convolution preconditioner can be computed rapidly for each fixed θ, using $O(n\log(n))$ flops; the preconditioner can therefore be applied for all $2n$ values of $\theta \in \Theta_n$ using $2n$ 1-D FFT's of length $2n$, for total work $O(N\log(N))$ flops. Because application of the preconditioned transform is so efficient, and because it has bounded condition number, it follows that traditional methods of iterative linear algebra (conjugate gradients) can efficiently yield approximate solutions of the equation

$$Y = RX$$

for X, given Y. Formally, we have

Corollary 1 *The generalized inverse $R^\dagger$ applied to an array $Y \in \mathcal{R}$ can be computed approximately within an ℓ^2 error ϵ in no more than $C_\epsilon N\log N$ flops, where $C_\epsilon = O(\log(\epsilon^{-1}))$.*

In practice, the behavior of the iterative algorithm is even more favorable than one might expect based on the above formal results. Empirically, the maximum ratio of the singular values of R is not greater than 1.1. Typically, three iterations of a CG solver are adequate for six-digit accurate reconstructions.

In short, this notion of Radon transform possesses properties [P1]-[P4] above: it is algebraically exact, geometrically faithful, rapidly calculable, and invertible on its range, with rapidly calculable approximate inverse. It also exhibits [P5] – parallels with the continuum theory – once we realize that the appropriate continuum theory is the Slant Stack Sf; our discrete projection-slice theorem and the convolution preconditioner mirror central facts in the continuum Radon theory.

1.5 Contents

The sections that follow prove the results just stated: Theorems 1-4 and the Corollary are proved in Sections 2-7, respectively. Section 8 discusses actual numerical behavior.

In Section 9, we discuss relations of our ideas with existing ideas in the tomography literature.

In Section 10, we discuss generalizations, for example to 3-D data.

1.6 Antecedents

We view the notion of discrete Radon transform developed here as the culmination of what we have earlier called a fourth stream of research on Radon transforms for discrete data – a stream which is comparatively unknown and scattered throughout the literature of engineering and applied mathematics, with roots in fields as diverse as the design of new medical scanners and the search for extraterrestrials. Although in writing this paper we were initially unaware of earlier work in this direction we became aware of work of Bailey and Swartztrauber [2, 3], Pasciak [25], Edholm and Herman [12, 13], and Lawton [19]. Each of these cited works makes a major algorithmic contribution, which contains within it key ingredients which could be used to establish Theorems 2 and 4 above.

We believe that prior to this paper, those algorithmic ideas were not used to define or to compute the Radon transform or its adjoint or inverse as we have defined it here; we explain our belief in detail in Section 9 below. We believe we have created a coherent mathematical framework that offers the most natural way to define a Radon transform for digital data, and which gives certain algorithmic ideas an intellectually clear role in supporting that framework.

2 Projection-Slice Theorem

2.1 Definition as Slant Stack

To prove Theorem 1, we make a few remarks about our interpolation method.

First, we have chosen $m = 2n$. This means that, in the case of I^1, our interpolation scheme is algebraically identical to embedding the image I in an array that is m-tall and n wide, with zero padding by $n/2$ rows of zeros both above and below the array, and using trigonometric interpolation of degree m within each column of the array.

Second, we remark that summing along a line $y = sx + z$ is equivalent to first shifting each column vertically by an amount $-su - z$ using trigonometric interpolation, which provides circulant shifts, and then summing along $y = 0$.

These observations furnish a different definition of R, for which Theorem 1 will be easier to prove.

Let E^1 be the operator of extension, padding an array n wide and n tall to be n wide and m tall by adding extra rows of zeros symmetrically above and below the input argument. Let E^2 be the operator of extension, padding an array n wide and n tall to be m wide and n tall by adding extra columns of zeros symmetrically left and right of the input array. Let $\tilde{I}^1 = E^1 I$ and $\tilde{I}^2 = E^2 I$. There is little risk of confusion of these arrays with the symbols introduced in Section 1, since they take the same values at the integer arguments, and since the values in between arguments of the earlier object can be obtained exactly by trigonometric interpolation from the values of the new object at the integer arguments.

Let T_τ denote the operator of τ translation, taking an m vector $\alpha = (\alpha_t : -n \leq t < n)$ into a vector $T_\tau \alpha$ of m elements indexed by $(-n \leq u < n)$ in which the position of elements is shifted by τ according to

$$(T_\tau \alpha)_u = \sum_{v=-n}^{n-1} \alpha_v D_m(v - u - \tau).$$

Here τ is not necessarily integer; the formula performs trigonometric interpolation when necessary.

For $-\pi/4 \leq \theta < \pi/4$, let $\tau(\theta, u; m) = \tan(\theta) \cdot u$; this is a shift which varies systematically with u following a line of slope $\tan(\theta)$. For $\pi/4 \leq \theta < 3\pi/4$, τ is defined analogously, with cotan playing the role of tan.

For $-\pi/4 \leq \theta < \pi/4$, let S^1_θ denote the operator of shearing the array I so that the line at slope $\tan(\theta)$ is moved to become a horizontal line. This takes an array of size n by n and produces an array of size n by m with

$$(S^1_\theta I)(u, \cdot) = T_{-\tau(\theta,u)} I(u, \cdot),$$

here the translation is applied in the v coordinate, with a different translation at each different value of u, i.e. in each column. For $\pi/4 \leq \theta < 3\pi/4$, S^2_θ is defined analogously with roles of v and u reversed, and with the names 'column' and 'row' reversed. See Figure 3. For more discussion of shearing using trigonometric interpolation, see Unser, Thevenaz, Yaroslavsky [28].

We have the following formal equivalence; see Figure 4 for a graphical illustration.

Lemma 1 *For each θ, summing along lines as in the Radon Transform produces the same result as shearing the array and summing the sheared array, either horizontally or vertically as the case may be:*

$$(RI)(t, \theta) = \begin{cases} \sum_u (S^1_\theta \tilde{I}^1)(u, t) & -\pi/4 \leq \theta < \pi/4, -n \leq t < n \\ \sum_v (S^2_\theta \tilde{I}^2)(t, v) & \pi/4 \leq \theta < 3\pi/4, -n \leq t < n \end{cases}$$

Indeed, the shearing has simply transported the values along certain specific basically horizontal lines to be exactly horizontal, (or else the values along basically vertical lines to be exactly vertical), and so simple summation across values $I(u, t)$ evaluates the same sum that earlier was across $I(u, su + t)$.

We find it instructive to think of this as performing what seismologists call a 'slant stack' [10, 29]. For seismologists, stacking is the operation of summing an array with two subscripts $A(u, v)$ to be an array of one subscript $\sum_u A(u, v)$, which is exactly the operation performed here. However, the operation is being performed on a slanted version of the original image; hence it is indeed a slant stack.

2.2 Proof of Theorem 1

$I \mapsto RI(\cdot, \theta)$ is a mapping from images in $\mathcal{I} \equiv \mathbf{C}^{n^2}$ to vectors in $V \equiv \mathbf{C}^m$ which we will denote R_θ . Let $[,]$ denote the standard inner product for $\mathcal{I} \equiv \mathbf{C}^{n^2}$ and $\langle , \rangle$ denote the standard inner product for $V \equiv \mathbf{C}^m$. We begin by computing the adjoint of R_θ.

Let $\psi = (\psi_t : -n \le t < n) \in V$ be a vector and let its backprojection be defined for $-\pi/4 \le \theta < \pi/4$ by

$$(B^1_\theta)(u, \cdot) = T_{\tau(u,\theta)}\psi,$$

and, similarly, for $\pi/4 \le \theta < 3\pi/4$ by

$$(B^2_\theta)(\cdot, v) = T_{\tau(v,\theta)}\psi.$$

Define also the truncation operator U^1 which takes an m-tall and n wide array into an n by n array by dropping rows outside the range $-n/2 \le v < n/2$ and similarly U^2 which takes an n-tall and m-wide array into an n by n array by dropping columns outside the range $-n/2 \le v < n/2$.

We now make the crucial observation about the translation operator:

$$\text{adj } T_\tau = T_{-\tau}.$$

This, the fact that U^1 is the adjoint of E^1, and similar observations about the duality of $\sum_u S^1_\theta$ and B^1_θ give

$$\text{adj } R(\cdot, \theta) = \begin{cases} U^1 B^1_\theta & -\pi/4 \le \theta < \pi/4, \\ U^2 B^2_\theta & \pi/4 \le \theta < 3\pi/4. \end{cases}$$

Now we have by definition of the adjoint that

$$[I, \text{adj } R_\theta \psi] = \langle R_\theta I, \psi \rangle, \qquad \forall I \in \mathcal{I}, \forall \psi \in V.$$

The remainder of the proof is to consider the implications of this relation in the special case where $\psi = \psi^{(k)}$, with

$$\psi^{(k)}_t = \exp\{i \frac{2\pi}{m} kt\}, \qquad -n \le k < n,$$

and to show that then

$$(\text{adj } R_\theta \psi)(u, v) = \exp\{i(u\xi_1 + v\xi_2)\}, \quad -n/2 \le u, v < n/2, \tag{4}$$

where

$$\xi_1 = \pi \cdot \frac{k}{m} \cdot \tan(\theta), \quad \xi_2 = \pi \cdot \frac{k}{m}. \tag{5}$$

This implies that

$$[I, \text{adj } R_\theta \psi] = \hat{I}(\pi \cdot \frac{k}{m} \cdot \tan(\theta), \pi \cdot \frac{k}{m})$$

giving a value of the 2-D Fourier transform of I, while

$$\langle R_\theta I, \psi \rangle = \widehat{R_\theta I}(k)$$

gives a value of the $1 - D$ Fourier transform of $R_\theta I$. This shows that the $1 - D$ Fourier transform of $R_\theta I$ gives the values in a radial slice of the $2 - D$ Fourier transform of I, and establishes Theorem 1.

It remains to establish (4) -(5). This says in words that *the Backprojection of a 1-D sinusoid at a Fourier Frequency for the m-long grid is a 2-D sinusoid for the m by m grid.*

The key observation is that *Translation by τ is algebraically exact for every sinusoid at a Fourier frequency of the m grid*, i.e. that if $\varphi(x) = \exp\{i\frac{2\pi}{m}kt\}$ is a function of a real variable x, and if $\phi = (\phi_t)$ is a vector of samples of φ, $\phi_t = \varphi(t)$ at the integers $-n \le t < n$, then

$$(T_\tau \phi)_t = (\varphi(t+\tau) : -n \le t < n).$$

Hence, for $-\pi/4 \le \theta < \pi/4$ and $-n/2 \le u, v < n/2$ we have

$$\begin{aligned}(\text{adj } R_\theta \psi^{(k)})(u,v) &= (B^1_\theta \psi^{(k)})(u,v) \\ &= \exp\{i(\frac{2\pi}{m}k(v+\tau(u,\theta)))\} \\ &= \exp\{i(\frac{2\pi}{m}k(v+\tan(\theta)u))\}.\end{aligned}$$

This establishes (4)-(5) and completes the proof.

3 Pseudopolar Fourier Transform

3.1 Pseudopolar Grid

We now consider Theorem 2. The key point here is the special nature of the angles $\theta^s_k \in \Theta_n$; this was alluded to in the Introduction, where the 'grid-friendliness' of this set of angles was mentioned.

Taking the Radon transform at these angles, we get, according to the Projection-Slice Theorem a one-one connection with the values of the 2-D Fourier transform at an associated set of frequencies Ξ_n. We label these frequencies $\xi^s_{\ell,k}$; they are given by

$$\begin{aligned}\xi^1_{\ell,k} &= (\frac{2\pi}{n}\ \ell k\ \frac{2}{n}\ , \frac{2\pi\ell}{n}) \qquad -n \le k < n, \quad -\frac{n}{2} \le \ell < \frac{n}{2}, \\ \xi^2_{\ell,k} &= (\frac{2\pi}{n}\ \ell, \frac{2\pi}{n}\ \ell k\ \frac{2}{n}) \qquad -n \le k < n, \quad -\frac{n}{2} \le \ell < \frac{n}{2}.\end{aligned}$$

This is a special non-Cartesian pointset in frequency space $[-\pi,\pi)^2$ which we earlier called the *Pseudopolar grid*, and illustrated in Figure 5. To make this set more concrete in the reader's mind, we visualize a geometric construction in Figures 6-7. First, we define a special collection of $2n$ lines in the continuum square, by markingout an equispaced sequence of n points on the upper boundary of the square and a comparable sequence of n points on the right-hand boundary. Connecting these points to the center of the square, we define the $2n$ lines; see Figure 6. We recall that in the continuous Fourier analysis, a radial line in the Fourier domain is associated, by duality, with a line in the spatial domain oriented at 90 degrees. The same thing is true for the Projection-slice theorem proved earlier. Hence the basically vertical lines in the frequency domain are actually *duals* of the basically horizontal lines in the ordinary spatial domain, so we associate them to $s = 1$; the basically horizontal lines we associate to $s = 2$. With these lines defined, the grid points arise as illustrated in Figure 7:

$\xi^1_{\ell,k}$ is the intersection of the ℓ-th line in $\mathcal{L}^1$ with the k-th horizontal line.

$\xi^2_{\ell,k}$ is the intersection of the ℓ-th line in $\mathcal{L}^2$ with the k-th vertical line.

In this pseudopolar representation, $|k|$ indexes (pseudo-) radius and ℓ indexes angle. The level sets of (pseudo-) radius are squares; so this grid may be called a *Concentric Squares Grid*; it was introduced by Mersereau and Oppenheim [22]. Obtaining the values of the Fourier transform at this pointset will be so important for us that we formalize the operation.

Definition. The *Pseudopolar Fourier Transform* P is the linear transformation from data $(I(u, v) : -n/2 \leq u, v < n/2$ to data $(\hat{I}(\xi^s_{\ell,k}) : s = 1, 2,\ -n \leq k < n, -\frac{n}{2} \leq \ell < \frac{n}{2})$, where $\hat{I}$ is the trigonometric sum

$$\hat{I}(\xi_1, \xi_2) = \sum I(u, v) \exp\{-i(\xi_1 u + \xi_2 v)\} \tag{6}$$

By the Projection-Slice Theorem, we have the decomposition

$$R = \mathcal{F}_1^{-1} \circ P, \tag{7}$$

where $\mathcal{F}_1^{-1}$ denotes the 1-D inverse discrete Fourier transform along the $t \leftrightarrow k$ transform pair.

3.2 Pseudopolar FFT

Identity (7) is interesting, in light of:

Theorem 6 *Pseudopolar FT can be computed in $O(n^2 \log(n))$ flops.*

This allows us to compute R rapidly. By (7), R is the composition of P with $\mathcal{F}_1^{-1}$. Now P can be computed in $O(n^2 \log(n))$ flops by Theorem 6, and $\mathcal{F}_1^{-1}$ requires a series of $2n$ 1-D FFT's, for $O(n^2 \log(n))$ flops. In terms of the $N = n^2$ values in the array, the whole procedure takes $O(N \log(N))$; Theorem 2 of the introduction follows.

The Pseudopolar FFT algorithm follows two stages. First, we calculate the usual 2-D FFT, which rapidly gives values of $\hat{I}$ on the Cartesian grid $(\frac{2\pi}{m}k_1, \frac{2\pi}{m}k_2)$ for $-n \leq k_1, k_2 < n$; Second, we perform *Cartesian-to-Pseudopolar conversion*, calculating values of $\hat{I}$ at the pseudopolar grid points. Both stages cost only $O(N \log(N))$ flops.

Figures 8 and 9 illustrate the structural features of Cartesian to Pseudopolar conversion. To obtain pseudopolar values in the Panel $s = 1$, we work one row at a time. The Cartesian values in a single row of the array are used to calculate the pseudopolar values in the same row. This is applied across all rows of the array, obtaining thereby all the pseudopolar values in panel $s = 1$. The approach for Panel $s = 2$ is similar, with the role of rows replaced by columns.

At the heart of the algorithm is the interpolation of $m = 2n$ equispaced points in the k-th row to produce n points with special spacing $\alpha = 2k/n$. This is rapidly computible according to:

Lemma 2 *Given m values of a trigonometric polynomial T of degree $m = 2n$ and period m*

$$T(\ell) = \sum_{u=-n}^{n-1} c_u \, e^{i\frac{2\pi}{m}\ell u}, \quad -n \le \ell < n,$$

it is possible, for any α, to find an equispaced set of m values

$$T(\ell\alpha) \qquad \ell = -n, \dots, n-1 \tag{8}$$

in order $O(n \log n)$ time.

Conceptually, this process allows us to define an operator $G_{n,k}$ which takes m values at the Cartesian grid points $-n \le k < n$, obtains the unique trigonometric polynomial generating those values, and delivers n values of that polynomial at more finely spaced points $-\alpha n/2 \le \ell\alpha < \alpha n/2$.

To show specifically how to rapidly compute $G_{n,k}$, write

$$G_{n,k} = \frac{1}{m} \cdot \mathcal{F}_\alpha \circ \mathcal{F}_1^{-1}.$$

where here $\mathcal{F}_1$ denotes the 1-D Fourier transform, and $\mathcal{F}_\alpha$ denotes the so-called *Fractional Fourier Transform* [2], defined by

$$(\mathcal{F}_\alpha X)_\ell = \sum_{k=-n}^{n-1} X_k \exp\{-i\alpha\frac{\pi}{n}\ell k\} \qquad -n \le \ell < n$$

As compared to the usual FFT, an extra factor α appears in the exponent; and $\mathcal{F}_1$ denotes the usual FFT, with $\mathcal{F}_{-1}$ denotes the usual inverse FFT (up to normalization).

Lemma 3 *Let X be a vector with $2n$ entries. The $2n$ entries of $\mathcal{F}_\alpha X$ can be evaluated in $O(n\log(n))$ flops.*

This lemma has been known since the 1960's [26, 5]; the Chirp-Z transform is the most well-known way to obtain it. An extensive discussion and a convenient algorithm for solving this problem using the fractional Fourier transform terminology can be found in [2], which, however, did not refer to much of the earlier literature.

The lemma allows us to obtain values of $\hat{I}$ on the pseudopolar grid from those on the Cartesian grid. Indeed the pseudopolar gridpoints $\xi^s_{\ell,k}$ for fixed s, k and variable ℓ, lie along a certain horizontal or vertical line in $[-\pi,\pi)^2$; along that same line there are $2n$ Cartesian grid points. Consider a line in panel $s = 1$. The restriction of $\hat{I}$ to this line is in fact a trigonometric polynomial T^1_k of degree m and period m which is determined by

$$T^1_k(\ell) = \hat{I}(\frac{2\pi}{m}\ell, \frac{2\pi}{m}k).$$

The desired pseudopolar values

$$\hat{I}(\xi^1_{\ell,k}) = T^1_k(\alpha\ell), \qquad -n/2 \le \ell < n/2$$

are provided by Lemma 2 setting $\alpha = \alpha_k = 2k/n$.

So Lemma 2 gives us an algorithm to evaluate these values $\hat{I}\big(\frac{2\pi k}{n}\ , \frac{2\pi}{n}\ \ell k \frac{n}{2}\big)$ for fixed k in order $n \log n$ time. Applying the algorithm at each possible combination of k and s, we get all values $PI(s,k,\ell)$ in order $2n\ O(n \log n)$ time; i.e., $O(n^2 \log n)$ flops. This completes the proof of Theorem 6.

4 Unique Reconstruction

We now consider Theorem 3, the assertion that R is of full rank. This is based on the following well-known observation.

Lemma 4 *A trigonometric polynomial* $T(\tau) = \sum_{u=-\frac{n}{2}}^{\frac{n}{2}-1} c_u\ e^{iu\tau}$ *is uniquely determined by its values* $T(\tau_j)$ *at any* n *distinct points* $(\tau_1, \dots, \tau_n)$.

One well-known proof of this lemma involves defining vectors $y = (T(\tau_j))$ and $x = (c_u)$, noting that these are related by $y = Vx$ where $V_{j,u} = z^{ju}$ is a Vandermonde matrix with $z = \exp\{i\delta\}$ for appropriate δ, and hence nonsingular.

Now as we have already pointed out, for every $k \neq 0$, the restriction of $\hat{I}(\xi)$ to the horizontal line $\xi_2 = \pi k/n$ is a trigonometric polynomial of degree at most n. There are n pseudopolar grid points belonging to that line. Hence simply the values of PI belonging to $s = 1$ alone determine the restriction of $\hat{I}$ to every horizontal line – except for the line $\xi_2 = 0$. Now among the Pseudopolar grid points for $s = 2$ there are $2n$ falling along the line $\xi_2 = 0$ (all those at index $\ell = 0$). Hence the values of PI associated to $s = 2$ determine the line $\xi_2 = 0$. In short, the values of $\hat{I}(\xi_1, \xi_2)$ are determined along each horizontal line. Hence the values at the grid points $\hat{I}(2\pi k_1/n, 2\pi k_2/n)$ are all determined, for $-n/2 \leq k_1, k_2 < n/2$; because these values generate I via inverse 2-D Fourier transformation, I itself is uniquely determined. Theorem 3 is proved.

5 Adjoint

We now turn to Theorem 4, rapid computation of adj R. Because of (7) we have

$$\text{adj } R = \text{adj } P \circ \mathcal{F}_1.$$

and so rapid computation of adj R reduces to rapid computation of adj P.

Conceptually, we have the decomposition

$$P = S \circ \mathcal{F}_2$$

where $\mathcal{F}_2$ denotes 2-D FFT, and S is the operator of resampling data on certain lines from the original Cartesian grid to the pseudopolar gridpoints. Now S is a block matrix, using only data on associated to a certain line to compute values at pseudopolar grid points on that line. The adjoint of a block matrix is itself a block matrix made of adjoints of block matrices. Hence, we focus on the adjoints of individual blocks.

Each nontrivial block operator B amounts to applying a sampling operator $G_{n,k}$ of the type discussed in Section 3, which conceptually takes $2n$ values at the Cartesian grid points $-n \leq k < n$, obtains the unique trigonometric polynomial generating those values, and delivers $2n$ values of that polynomial at more finely spaced points $-\alpha n \leq \ell\alpha < \alpha n$. Then the values outside the range $n/2 \leq \ell < n/2$ are discarded. Now such a block operator may be written as

$$B = U \circ G_{n,k} = \frac{1}{2n} \cdot U \circ \mathcal{F}_\alpha \circ \mathcal{F}_1^{-1}.$$

where U denotes the truncation operator which delivers just the values at $-n/2 \leq \ell < n/2$ from a sequence indexed by $-n \leq \ell < n$. Now by inspection

$$\text{adj } \mathcal{F}_\alpha = \mathcal{F}_{-\alpha}$$

and so

$$\text{adj } B = \frac{1}{2n} \cdot \mathcal{F}_{-1} \circ \mathcal{F}_{-\alpha} \circ E,$$

where $E = \text{adj } U$ is the extension operator which extends an array indexed by $-n/2 \leq \ell < n/2$ to be an array indexed $-n \leq \ell < n$, using zero-padding. In this display, the operator $\mathcal{F}_{-1}$ can be computed in $O(n \log(n))$ flops, and, again owing to Chirp-Z [26] or Fractional FT [2], so can $\mathcal{F}_\alpha$, for positive and negative alpha. Theorem 4 follows.

6 Conditioning

In this section we turn to Theorem 5, which asserts that a simple preconditioner yields a bounded ratio of singular values.

6.1 Ill-Conditioning

Like the continuous Radon transformation, the discrete Radon transformation is one-to-one, but the problem of recovering an image I from a noisy version of RI is ill-conditioned; there are objects I_1,I_2 of equal norm where RI_1 is small, but RI_2 has a large norm.

This is easily understood using the pseudopolar FT. By the projection-slice theorem, we have the isometry

$$\|RI\|_2 = \|PI\|_2.$$

Hence for equal norm objects I_1, I_2 to have very different norms under mapping by R, they must also have very different norms under mapping by P.

This issue is understandable in terms of the oversampling of the Fourier domain that is carried out by P. We note that the pseudopolar grid samples points at pseudo radius $k = n$ at a rate of exactly one pseudopolar sample per Cartesian sample. On the other hand, at pseudoradius zero, the grid samples points at a rate of n pseudopolar samples per Cartesian sample.

This is important, because we know that there is a Parseval relation for the 2-D FFT, which implies that the ℓ^2 norm of Cartesian samples of the Fourier transform is always identical to the normalized ℓ^2 norm of the original object I.

Combining these remarks, it is therefore clear that objects I which are concentrated in the Frequency domain at high pseudo-radius $k \approx n$ will have much smaller values of $\|PI\|_2$ than objects of equal norm concentrated at low pseudo-radius $k \approx 0$.

6.2 Preconditioning Operators

We now define a pre-conditioning operator for P; it normalizes pseudopolar samples by the sampling rate relative to the Cartesian samples; the appropriate weight is

$$D^s_{l,k} = \begin{cases} \sqrt{|k|/2}/n & k \neq 0 \\ \sqrt{1/8}/n & k = 0 \end{cases} .$$

The preconditioned pseudopolar FT is then defined by

$$\tilde{P} = D \circ P.$$

Here D is a purely diagonal operator in the pseudopolar domain, defined by $(D \circ X)^s_{l,k} = D^s_{l,k} \cdot X^s_{l,k}$. We note that the normalization at $k \neq 0$ is rather natural – based on the idea that the samples $(T(\ell))_\ell$ of a trigonometric polynomial of degree m and period m at unit sampling rate gives the same mean square as the normalized samples $(\sqrt{\alpha}T(\alpha\ell))_\ell$, while that at $k = 0$ is motivated by the $4n$-fold sampling of the point zero ($2n$ times in panel $s = 1$ and $2n$ times in panel $s = 2$), although an additional factor $1/\sqrt{2}$ has been inserted which cannot be explained in this way.

The corresponding preconditioned Radon transform is simply

$$\tilde{R} = \mathcal{F}_1 \circ \tilde{P}.$$

Because of the relation between 1-D convolution and 1-D Fourier transforms, we have along each constant-θ slice $\theta = \theta^s_k$ that

$$(\tilde{R}I(t,\theta) : -n \leq t < n) = \Delta \star (RI(t,\theta) : -n \leq t < n).$$

where the convolution operator Δ has frequency domain weights $\hat{\Delta}_k$ exactly as in Theorem 5.

To complete Theorem 5, we are interested in showing that for some constants C_0 and C_1, we have

$$C_0\|I\| \leq ||\tilde{P}I|| \leq C_1\|I\|.$$

which will imply that we have an effective preconditioner for P and consequently also for $\tilde{R}$.

We look at this as two inequalities, to be approached in two steps: an upper bound and a lower bound.

6.3 Analysis of Preconditioner, 1

To approach $||\tilde{P}I|| \leq C_1\|I\|$, we work in the frequency domain, and attempt to establish $||\tilde{P}I|| \leq C_1\|\hat{I}\|$.

Consider now each horizontal line in the Cartesian grid, and the corresponding samples in the horizontal line in the pseudopolar grid. After renormalization, we have a trigonometric polynomial T of period m, and two norms on T: first, the ℓ^2 norms of the samples in the Cartesian grid:

$$\|T\|_{Cart} = \left(\sum_{k=-n}^{n} |T(k)|^2 \right)^{1/2}$$

and second, the density-weighted norm of the samples in the pseudopolar grid.

$$\|T\|_{Frac} = \sqrt{\alpha} \cdot \left(\sum_{k=-n/2}^{n/2} |T(\alpha k)|^2 \right)^{1/2}$$

We have the following:

Lemma 5 *For each $\alpha \in [0,1]$,*

$$\|T\|_{Frac} \leq \sqrt{2} \cdot \|T\|_{Cart}.$$

It follows immediately from this that the density-weighted norm of the samples lying in panel $s = 1$ of the pseudopolar domain is at most $\sqrt{2}$ times as large as $\|\hat{I}\|$. Similarly for panel $s = 2$. Hence

$$||\tilde{P}I|| \leq \sqrt{8}\|\hat{I}\|, \qquad \forall I.$$

We now prove the lemma. We use a large sieve inequality of Donoho and Logan [11]. This says that if $S(\tau) = \sum_{k=0}^{m-1} c_k \exp\{i\frac{2\pi}{m}k\tau\}$ is a trigonometric polynomial of degree m and period 1, and if μ is nonnegative measure on $[0,1)$, then for each $\delta > 0$

$$\int_0^1 |S|^2(\tau)d\mu(\tau) \leq (n - 1 + \delta^{-1}) \cdot \sup_{\tau} \mu[\tau, \tau + \delta] \cdot \sum_k |c_k|^2.$$

Taking μ to be a sampling measure, this controls the size of a trigonometric polynomial on an irregular set as long as the set does not place too much mass in any interval of fixed length.

Adapting this to the present setting, we may take $S(\tau) = T(m(\tau - 1/2))$ and define the measure μ so that

$$\int_0^1 |S|^2(\tau)d\mu(\tau) = \alpha \cdot \sum_{k=-n/2}^{n/2} |T(\alpha k)|^2.$$

When we do so, we get for $\delta = 1/m$ and real-valued κ

$$\sup_{\tau\in[0,1)} \mu[\tau, \tau + \delta] = \sup_{-n\leq\kappa<n} \alpha \sum_{k=-n/2}^{n/2} 1_{[\kappa,\kappa+1]}(\alpha k);$$

in every interval of length 1 there are at most $\lceil 1/\alpha \rceil$ points of sampling, so

$$\sup_{\tau \in [0,1)} \mu[\tau, \tau + \delta] \leq \alpha \cdot \lceil 1/\alpha \rceil \leq 2.$$

This can be interpreted as saying that the norm is properly density-normalized. It follows that

$$\alpha \cdot \sum_{k=-n/2}^{n/2} |T(\alpha k)|^2 \leq (2n-1) \cdot 2 \cdot \sum_k |c_k|^2 \leq 2 \cdot \|T\|_{Cart}^2$$

where we used the Parseval relation $\|T\|_{Cart}^2 = m \cdot \sum_k |c_k|^2$. This completes the proof of the lemma.

6.4 Analysis of Preconditioner, 2

To get the lower bound $C_0\|I\| \leq ||\tilde{P}I||$, we again work in the frequency domain, and establish that $C_0\|\hat{I}\| \leq ||\tilde{P}I||$.

We are unable to establish this by mathematical analysis alone; our approach will be computer-assisted. That is, we will use analysis to reduce the question to the use of numerical linear algebra to establish that a certain matrix norm is strictly less than one. Then we will report the results of computations which establish this, and the availability of software that enable the reader to verify and replicate. We believe that the approach suggests a very informative viewpoint on the problem.

Now essentially the desired inequality $C_0\|\hat{I}\| \leq ||\tilde{P}I||$ states that, if we have a bivariate trigonometric polynomial of degree n the norm of the samples at the pseudopolar grid controls the norm of the samples on the Cartesian grid. To establish this, we will study an operator that takes samples on the pseudopolar grid and delivers samples on the Cartesian grid and show that this is bounded.

6.4.1 1-dimensional Resampling operator

Suppose we are working in dimension one, and we have a trigonometric polynomial T of degree n and period $2n$, and we are equipped with samples of T at two different sampling rates. For $\alpha = 2k/n$, we have density-normalized samples

$$\sqrt{\alpha} T(\alpha \ell) \qquad -n/2 \leq \ell \leq n/2$$

as well as

$$T(\ell) \qquad k \leq |\ell| \leq n.$$

Suppose we package these data into a vector W and consider the operator $H_{n,k}$ which, given such data, recovers the unique trigonometric polynomial T having such samples and then delivers the values

$$T(\ell) \qquad 0 \leq |\ell| < k$$

This is a linear operator, taking as argument vectors of $2n - 2k$ values and yielding as output vectors containing $2k - 1$ values. The problem is visualized in Figure 10

The operator describes a process of resampling from data which are oversampled at two different rates to data which are uniformly sampled at twice the Nyquist rate.

6.4.2 Pseudopolar-to-Cartesian Conversion by Onion-Peeling

Given the 1-dimensional operators $H_{n,k}$ we can perform a full 2-dimensional conversion from knowledge of pseudopolar to knowledge of Cartesian samples. Here is how. To begin with, if we know the pseudopolar samples, then we also know the Cartesian samples at the edges of the domain $[-\pi, \pi]^2$, along the main diagonal and skew diagonal, and along the axes. Now consider the problem of recovering all the Cartesian samples on the square associated with $|k| = n/2 - 1$. To get the Cartesian samples in the top row $s = 1$, $k = n/2 - 1$, we apply the operator $H_{n,n-1}$ to a vector consisting of the $n + 1$ pseudopolar samples in that row, together with the two Cartesian samples at the extremes of the array (which were known to begin with). We do something similar in the bottom row $s = 1$, $k = -n/2 + 1$ and in the rightmost column $s = 2$, $k = n/2 - 1$ and the leftmost column $s = 2$, $k = -n/2 + 1$. At this point, we have recovered all the Cartesian samples in the outermost two concentric squares. Continuing working in this way, we obtain in sequence the Cartesian samples in successively smaller concentric squares, until we reach $k = 1$, at which case the Cartesian samples are already present among the pseudo polar samples. One can liken this approach to peeling an onion. See Figure 11.

This should convince the reader that the operators $H_{n,k}$ are fundamental to the process of pseudopolar-to-Cartesian conversion.

6.4.3 Pseudopolar-to-Cartesian Conversion by Sectorial Imputation

To get quantitative control on the norm of pseudopolar to cartesian conversion, we turn from onion-peeling, which applied the $H_{n,k}$ noniteratively, recovering successively the various concentric squares, and develop instead an iterative scheme of approximately recovering one whole panel (or *sector*) at a time by applying the $H_{n,k}$ operators. This requires an appropriate set up. We will store the pseudopolar samples into two arrays corresponding to the basically vertical and basically horizontal directions. Hence, $X^{(1)}$ contains samples at pseudopolar frequencies which correspond to $|\theta| \leq \pi/4$ (both endpoints included) and also $\xi_1 = \pm\pi$; and $X^{(2)}$ contains samples at pseudopolar frequencies which correspond to $\theta \in [\pi/4, 3\pi/4]$ (both endpoints included) and also $\xi_2 = \pm\pi$. All these are samples which are known and fixed throughout the algorithm.

We will represent the Cartesian samples which are to be recovered in two arrays: $Y^{(1)}$ corresponding to samples in the cone covered by samples in $X^{(1)}$ and $Y^{(2)}$ corresponding to samples in the cone covered by samples in $X^{(2)}$. Now note that, *if we knew the Cartesian samples in $Y^{(1)}$, then, using the pseudopolar samples in $X^{(2)}$ and the operators $H_{n,k}$, we could recover exactly the Cartesian samples in $Y^{(2)}$.* Indeed, in each row the appropriate operator $H_{n,k}$ would want the pseudopolar samples in the inner region of that row, as well as the Cartesian samples in the exterior region of that row, and it would deliver the Cartesian samples in the inner region of that row. See Figure 12. Similarly, *if we knew the Cartesian samples in $Y^{(2)}$, then, using the pseudopolar samples in $X^{(1)}$ and the operators $H_{n,k}$, we could recover exactly the Cartesian samples in $Y^{(1)}$.* For future use, let $C(X, Y)$ denote the linear operator implicitly referred to by these italicized statements, so that we have

$$Y^{(2)} = C(X^{(2)}, Y^{(1)}), \qquad Y^{(1)} = C(X^{(1)}, Y^{(2)}).$$

Now consider applying the two italicized observations in a setting where we do not, initially, know the required samples $Y^{(i)}$. Instead, we make a sequence of guesses, as follows: starting with $Y_0^{(1)} = 0$, $Y_0^{(2)} = 0$, we calculate

$$Y_i^{(2)} = C(X^{(2)}, Y_{i-1}^{(1)}), \qquad Y_i^{(1)} = C(X^{(1)}, Y_{i-1}^{(2)}), \quad i = 1, 2, ... \tag{9}$$

In fact this process converges. To understand this, we can partition the operator in terms of two matrices C_1 and C_2:

$$C(X, Y) = C_1 X + C_2 Y.$$

Unrolling the iteration (9), we get, if i is even, that

$$\begin{aligned} Y_i^{(1)} &= C_1 X^{(1)} + C_2 Y_{i-1}^{(2)} \\ &= C_1 X^{(1)} + C_2 (C_1 X^{(2)} + C_2 Y_{i-2}^{(1)} \\ &= \sum_{j=0}^{i/2} C_2^{2j} C_1 X^{(1)} + \sum_{j=0}^{i/2-1} C_2^{2j+1} C_1 X^{(2)}. \end{aligned}$$

Hence, provided the operator norm $\|C_2\| < 1$, we have a series expansion for the Cartesian samples in terms of the pseudopolar samples:

$$Y^{(1)} = \sum_{j=0}^{\infty} C_2^{2j} C_1 X^{(1)} + C_2^{2j+1} C_1 X^{(2)}, \quad Y^{(2)} = \sum_{j=0}^{\infty} C_2^{2j} C_1 X^{(2)} + C_2^{2j+1} C_1 X^{(1)}.$$

To obtain a bound on the operator norm $\|C_2\|$, we note that, as remarked above, C is a block matrix, with $H_{n,k}$ operating in each block. The argument to $H_{n,k}$ is a vector which can be partitioned into components V and W, consisting of samples V from the interior zone, at high density, and samples W from the exterior zone, at low density; recall Figure 10. We can likewise partition the operator

$$H_{n,k} X = H_{n,k}^{int} V + H_{n,k}^{ext} W$$

We then observe that

$$\|C_2\| \leq \max_k \|H_{n,k}^{ext}\|.$$

Now for small-to-moderate n, the norm of $H_{n,k}^{ext}$ is computable using standard numerical linear algebra. The answers, given in Figure 13, show that this norm is well below 1.

Finding. *In the range of n for which it is possible to solve general n by n systems of equations by computer (say, $n < 1024$), and for all k in the range $1, \ldots, n/2 - 1$, the norm $\|H_{n,k}^{ext}\| < 1$.*

`Matlab` code for calculating Figure 13 is available at URL *http://www-stat.stanford.edu/~donoho/FastSlant/*. More extensive tabulations of $\|H_{n,k}^{ext}\|$ are given there as well.

Therefore, modulo the usual provisos in computer-assisted proofs (i.e. that the program has no bugs, and that `Matlab` software functions as advertised, etc.) we have proved the desired lower bound $C_0\|\hat{I}\| \leq ||\tilde{P}I||$ for a large range of n.

Although we do not give a proof here, we believe that the pattern established by computers at moderate n should persist for larger n. Our evidence in this direction is to consider an $n = \infty$ analog, using bandlimited functions in place of trigonometric polynomials and consider the analogous condition to $\|H^{ext}_{n,k}\| < 1$. Details will appear elsewhere.

7 Reconstruction Algorithm

We now turn to Corollary 1. Suppose we are given Radon data $Y = RI$. To reconstruct I, we pursue the following steps:

1. Define $F = \mathcal{F}_1 Y$, i.e. take the standard 1-dimensional Fourier transform of length m along each constant-θ slice.

2. Define $\tilde{F} = DF$, i.e. apply the preconditioning operator to the data F.

3. Define $\tilde{I} = (\text{adj } \tilde{P})\tilde{F}$, i.e. backproject to the image domain, using the adjoint of the preconditioned Radon transform.

4. Solve iteratively, by conjugate gradients, the system of equations

$$GI = \tilde{I}.$$

   where G is the Gram operator $G = (\text{adj } \tilde{P})\tilde{P}$. This requires a series of applications of G to various vectors, and a few vector operations per iteration.

This algorithm solves the problem given of the introduction, of iteratively inverting the Radon transform, and doing so rapdily. The first three steps are all accomplished exactly in exact arithmetic, and rapidly – in order $O(N \log N)$ flops or less. The only step which is not exact and not in closed form is the final step – solution of a Hermitian system by conjugate gradients.

Solving Hermitian systems by conjugate gradients is, of course, a central part of modern scientific computing and is extremely well understood [16, 17]. Defining the condition number κ of the Gram operator $G = (\text{adj } \tilde{P}) \circ \tilde{P}$ to be the ratio of largest to smallest eigenvalues of G, then the error of the k-th iteration's approximate solution I_k is bounded by

$$\|I_k - I\|_G \leq 2 \cdot \|I_0 - I\|_G \cdot \left[\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right]^k, \tag{10}$$

where $\|v\|_G^2 = v^H G v$. In practice this is an extremely pessimistic bound, but in fact it already establishes Corollary 1. Indeed, we know that κ is finite from the previous section, and therefore we can get an error tending geometrically to zero as a function of the number of iterations k. The cost per iteration is essentially the cost of P and adj P, each one being $O(N\log(N))$.

8 Numerical Evidence

We now present numerical evidence which shows that the efficiency of our inversion routine is far better than the mathematical analysis so far would suggest. Indeed, a large fraction of the eigenvalues of $G = (\mathrm{adj}\ \tilde{P}) \circ \tilde{P}$ are very close to 1. Moreover the condition number of the Gram operator is 1.2 or smaller. As a result, the conjugate gradients method converges after 3 steps – that is to say, error tolerances of size 10^{-7} are typically reached in 3 steps only.

8.1 Singular Values of $\tilde{P}$

Inequality (10) shows that the spread in the singular values of $\tilde{P}$ controls the convergence of CG. This makes it of interest to know the singular value distribution of $\tilde{P}$. For small n (say $n = 8$ or 16), it is possible to directly build the matrix of $\tilde{P}$ and perform a singular value analysis. For example, with $n = 8$, $\tilde{P}$ is a 256×64 matrix; with $n = 16$, $\tilde{P}$ is a 1024×256 matrix. Singular value decompositions for dense matrices of these sizes are well within the capacity of typical scientific computing equipment. We have the results in Table 8.1 below. Here $\sigma_{[f]}$ denotes the singular value which exceeds a fraction f of the nonzero singular values.

n	$\sigma_{\min}$	$\sigma_{[1/8]}$	$\sigma_{[1/4]}$	$\sigma_{[1/2]}$	$\sigma_{[3/4]}$	$\sigma_{[7/8]}$	$\sigma_{\max}$
8	0.9430	0.9912	0.9965	1.0000	1.0001	1.0015	1.0281
16	0.9586	0.9978	0.9999	1.0000	1.0000	1.0001	1.0008

Note the large fractions of singular values clustered very tightly around 1.

These tables show that the quantity $\sqrt{\kappa} \approx 1.090$ for $n = 8$ and $\sqrt{\kappa} \approx 1.074$ for $n = 16$.

8.2 Extreme Eigenvalues of G

For n in the range of much practical interest, the matrix of $\tilde{P}$, a $4n^2 \times n^2$ array, is simply too big for effective singular value decomposition. Even at $n = 64$, which is comparatively small in an image analysis context, we get a 16384×4096 array of complex numbers, requiring 1 gigabytes of storage in double precision, and, supposing the storage requirement were met, requiring comparably extreme cpu times.

Since we have fast algorithms for P and adj P, we can use the power method to obtain information about the extreme eigenvalues of $G = \mathrm{adj}\ \tilde{P} \circ \tilde{P}$. Letting x_0 be a random n^2 by 1 vector, we compute, for $t = 1, 2, \ldots$ the quantities $y_t = Gx_{t-1}$, $x_t = y_t/\|y_t\|_{\ell^2}$. The power method asserts that the quantity $\ell_t \equiv \|y_t\|_{\ell^2}$ converges to $\lambda_{\max}(G)$. The power method for $\lambda_{\min}$ works similarly, computing instead $y_t = (G)^{-1}x_{t-1}$, $x_t = y_t/\|y_t\|_{\ell^2}$; the quantity $\ell_t^{-1} \equiv \|y_t\|_{\ell^2}$ converges to $\lambda_{\max}((G)^{-1}) = \lambda_{\min}^{-1}(G)$. (Of course in fact we have only an approximation to $(G)^{-1}$, obtained by conjugate gradients; we ignore this for now.) We obtained these numerical results for $\hat{\kappa} = \hat{\lambda}_{\max}/\hat{\lambda}_{\min}$

n	$\hat{\kappa}$
32	1.2037
64	1.2124
128	1.1280
256	1.1317

Here $\hat{\lambda}_{\min}$ and $\hat{\lambda}_{\max}$ are decorated with hats to remind the reader of the fact that these are numerical estimates obtained by a fixed number of iterations of a conjugate gradients algorithm. These data are consistent with the finding that the condition number of the Gram system obeys $\sqrt{\kappa} \approx 1.1$ or less.

8.3 Convergence History

According to (10) and the numerical evidence just presented, we can expect rapid convergence of the reconstruction algorithm. In fact, we find that for practical work, 3 iterations of the CG solver suffice.

Typical behavior of the CG solver is illustrated by the following example. For $n = 32$, an object I of norm one was transformed to RI. During the reconstruction, the CG solver reported the error $\|I_k - I\|_G$, giving

```
Iteration 1: Residual error=0.000798
Iteration 2: Residual error=0.000097
Iteration 3: Residual error=0.000005
Iteration 4: Residual error=0.000000
```

So the reconstruction had already 3-digit accuracy after the first CG step, and rougly one additional digit was gained at each iteration. This particular object was simply the indicator of a single pixel; but similar results have been observed on random objects, and also on objects at much larger n.

9 Previous Work

We now clarify the relationship between earlier work and certain results in this paper, specifically Theorems 2 and 4 (concerning fast algorithms).

9.1 Bailey and Swartztrauber

Bailey and Swartztrauber wrote a lovely article on the Fractional Fourier transform for SIAM Review in 1992, which we encourage every computational scientist to review. (Though we regret the lack of any discussion of the Chirp-Z transform, dating from twenty years earlier [26], which is in fact more general than the fractional Fourier transform as introduced in [2]).

In the examples section at the end of their article, Bailey and Swartztrauber consider the problem of summing along a family of lines in an array, and suggest the idea – like in our introduction – of trigonometric interpolation. They then show that, using the fractional Fourier transform, one can calculate rapidly the array of sums over all lines with slopes $\beta \cdot k/n$ and offsets γt, for constants β and γ. See also [3]. Obviously this comes close to containing Theorem 2.

However, Bailey and Swartztrauber do not explicitly define or defend a notion "Radon transform"; they seem merely interested in getting sums over a rich family of lines. Moreover, they do not proceed as above, i.e. establishing a projection-slice theorem which relates sums along lines to Fourier coefficients on a pseudopolar grid. Another difference is that their definition of summing along lines can be shown equivalent to ours using interpolating kernel D_m for $m = n$ rather than $2n$. For us, this is a crucial difference, because of the *wrap-around artifacts* that it causes.

Figure 14 gives an example of an array sheared using $m = n$ rather than $m = 2n$. Comparing this to Figure 3, the reader can see the image wraparound caused by $m = n$. Figure 15 exhibits an analog of our Lemma 1, this time for $m = n$, namely that summation over slanted lines in the original image is equivalent to summation over horizontal lines in the sheared image. As the sheared image in the $m = n$ case exhibits wraparound, the implication is that the 'lines' summed over by Bailey and Swartztrauber are not true geometric lines. To check this, Figure 16 computes the analog of applying adj R for the $m = n$ case to an array with a single nonzero element; this is analogous to Figure 2 for the $m = 2n$ case. The display shows that coefficients in the $m = n$ case are indeed formed by summing along 'wrapped lines'.

Bailey and Swatrztrauber deserve a great deal of credit for an important series of algorithmic insights. Our contributions are: (1) to insist on a specific definition of a full Radon transform, using a specific set of offsets and angles, and not merely a convenient algorithm for general families of sums; (2) to insist on $m = 2n$ and so on geometric fidelity of the corresponding family of lines; and (3) to formalize a pseudopolar FFT and recognize a Projection-Slice Theorem relating the pseudopolar FFT to the Radon; (4) to establish properties for the Radon transform, including injectivity and inversion algorithms; and (5) to put the various results in scholarly context.

9.2 Mersereau and Oppenheim

Mersereau and Oppenheim, in an influential survey paper [22], introduced a non-Cartesian grid in the 2-D Fourier plane. This a pseudopolar grid of the type we have described here in Section 3.1, except for the degree of radial sampling. See Figure 17. Mersereau and Oppenheim called this the *concentric squares grid.*

Mersereau and Oppenheim worked in this article from the viewpoint of computed tomography. They assumed that data on a continuum object were gathered in unequally spaced projections chosen so that the 1-d Fourier transform corresponded to the concentric squares grid.

They considered the problem of reconstructing a discrete array of n^2 pixels from such Fourier domain data, and developed an algorithm based on interpolating from the data given in the concentric squares grid to the Cartesian grid. Mersereau and Oppenheim used simple 1-dimensional interpolation based on linear interpolation in rows/columns.

In short, a key organizational tool – a trapezoidal grid for Fourier space – has been known since 1974, under the name ‘concentric squares grid’. In fact, this grid has since been rediscovered numerous times. Mersereau and Oppenheim seem to be the real inventors of this concept and deserve clear credit.

In comparison to our work: (1) Comparing Figures 17 and 5, we see that Mersereau and Oppenheim’s definition samples half as frequently in the radial direction. This can be shown to be exactly the grid which would arise if we had developed our original Radon definition for the $m = n$ case. Hence, the original concentric squares grid involves wrap-around of the underlying lines; (2) Mersereau and Oppenheim’s methodology is about reconstruction from data given about a continuum object; they do not attempt to define a Radon transform on digital data, or establish the invertibility and conditioning of such a transform; and (3) their methodology is approximate – they do not obtain an exact conversion between concentric-squares and Cartesian grids.

9.3 Pasciak, Lawton, Herman and Edholm

We now consider an important set of papers in the literature of computed tomography – both medical tomography [25, 12, 13] and synthetic aperture radar imaging [19]. Like Mersereau and Oppenheim, these authors are concerned with image reconstruction; effectively they assume that one is given data in the Fourier domain on a concentric squares grid.

Pasciak’s unpublished work [25] – which is known among tomography experts through a citation in Natterer’s book – showed in 1980 that, given data on a pseudopolar grid in Fourier space, one could calculate a collection of n^2 sums which, using the notation of this paper we can write as

$$\sum c^s_{k,\ell} \exp\{i(u,v)'\zeta^s_{k,\ell}\}, \qquad -n/2 \leq u, v < n/2, \tag{11}$$

where the $\zeta^s_{k,\ell}$ are points in the concentric squares grid. (Pasciak makes no reference to Mersereau and Oppenheim.) Pasciak studied this calculation, which is essentially what we would call the calculation of adj P for a variant of P based on $m = n$ rather than $m = 2n$, and shows it may be done in order $n^2 \log(n)$ time. His key insight was to use the chirp-Z transform to calculate Fourier-like sums with exponents different from the usual $2\pi/nkt$ by a factor α.

Edholm and Herman [12, 13] develop the linogram, with a very similar point of view. They assume that data on a continuum object have been gathered by what we have called the continuous Slant Stack, at a set of projections which are equispaced in $\tan(\theta)$ rather than θ. By digitally sampling each constant θ projection and making a 1-D discrete Fourier transform of the resulting samples, they argue that they are essentially given data on a concentric squares grid in Fourier space, (making no reference to Mersereau and Oppenheim or Pasciak.) They are concerned with reconstruction and consider the sum (11) and derive a fast algorithm – the same as Pasciak’s, using again the chirp-Z transform.

Contemporaneously with Edholm and Herman, Lawton [19] develops a so called Polar Fourier transform for Synthetic Aperture Radar (SAR) imagery. He introduces a concentric squares grid, assumes that SAR data are essentially given on such a concentric squares grid in Fourier space, and considers the problem of rapidly reconstructing an image from such data. He considers the sum (11) and derives a fast algorithm using again the chirp-Z transform. He refers to Mersereau and Oppenheim.

These authors deserve major credit for identifying an important algorithmic idea – use of Chirp-Z techniques to resample data from Cartesian to Concentric-squares grids – which obviously is the same idea we use in our fast algorithms.

In comparison to our work: (1) this methodology is about reconstruction only, assuming that data are gathered about a continuum object by a physical device, and (2) the algorithmic problem they consider is equivalent to rapidly computing (11).

From the viewpoint of this paper, they developed ideas which are the same ideas we use to yield a fast algorithm adj R. However, they do not identify what we call R nor therefore adj R; they conceptualize the sum (11) as a Riemann sum approximation to an inversion formula for data gathered on the continuum in Slant-Stack form. Thus for example, the fact that adj R does not provide exact reconstuction of I from data RI would not even be in their conceptual vocabulary. Finally, their trapezoidal grids $\zeta^s_{k,l}$ coincide with the Mersereau-Oppenheim Concentric Squares grid of Figure 17, rather than our Pseudopolar grid of Figure 5. This difference can be seen from the perspective of this paper to be equivalent to defining the Radon transform using $m = n$ rather than $m = 2n$; hence these approaches can be said to lack geometric fidelity. If these authors had been interested in defining R and adj R, they might also have found out, as we have, that the variant based on $m = n$ is much more poorly conditioned than the variant based on $m = 2n$, so that iterative methods of linear algebra work considerably more poorly in the $m = n$ situation.

Our contribution is to consider both analysis and reconstruction, to define a geometrically faithful notion of Radon transform, show that it has an inverse and a natural preconditioner – and to identify the earlier algorithmic ideas as contributing a fast algorithm for the adjoint of our Radon transformation.

10 Generalization to Three Dimensions

10.1 Three-Dimensional Radon Transform

The 3-D Radon transform can be defined as a sum of $I(u, v, w)$ over planes, using interpolation. We will call a z-plane a plane where $z = s_1x + s_2y + t$ with slopes s_i less than one in magnitude. For z-planes, we define

$$\text{Radon}(\{z = s_1x + s_2y + t\}, I) = \sum_u \tilde{I}^z(u, v, s_1u + s_2v + t).$$

where we employ zero-extended Trigonometric interpolation with $m = 2n$:

$$\tilde{I}^z(u, v, z) = \sum_{u,v=-n/2}^{n/2-1} I(u, v, w)D_m(z - w).$$

Similar definitions can be made for x-planes and y-planes. If we now define offsets $t \in T_n$ and consider slope-pairs s_1, s_2 obeying

$$s_i = 2\ell_i/n, \quad -n/2 \leq \ell_i \leq n/2,$$

then setting

$$(RI)(1, t, \ell_1, \ell_2) = \text{Radon}(\{z = s_1 x + s_2 y + t\}, I),$$

with similar definitions for $s = 2$,3, gives a transformation from voxel images $I(u, v, w)$ to their Radon transforms $RI(s, t, \ell_1, \ell_2)$, taking n^3 arrays into four way arrays with $6n^3$ elements.

10.2 Three-Dimensional Pseudopolar FFT

For three dimensions, we can define a Pseudopolar grid in a fashion analogous to the two dimensional case. First, we define lattices of n^2 equally spaced points on each of the faces $x = pi$, $y = pi$ and $z = pi$, respectively. Next, we connect each lattice point to the origin, creating $3n^2$ x-like lines, y-like lines, and z-like lines, respectively. Finally, we consider the intersection of these lines with equispaced families of $2m$ planes of the form $x =$ constant, (respectively $y =$ constant, $z =$ constant). This creates a collection of $6n^3$ points in the frequency domain, lying in three cones, as in Figure 18. The Pseudopolar FT is simply the problem of calculating all sums

$$I(\xi^s_{k,l_1,l_2}) = \sum_{u,v,w} I(u, v, w) \exp\{-i(u, v, w) \cdot \xi^s_{k,l_1,l_2}\}$$

for $-n \le k < n$, $-n/2 \le \ell_1, \ell_2 < n/2$ and $s \in \{1, 2, 3\}$. An algorithm for calculating a 3-D Pseudopolar FFT can be developed just as in the 2-D problem. One takes a standard 3-D FFT to get m^3 Cartesian points, and then, to get the pseudopolar points in panel $s = 1$, one resamples along lines embedded in $x =$ constant planes, those lines being parallel either to the z or y axis. The resmpling operator is exactly the $G_{n,k}$ operator discussed before. The total operations count is again $O(N \log N)$, this time with $N = n^3$.

10.3 Main Results for 3-D

With the principal objects defined, we note that, just as in 2-D, there is a Projection-Slice Theorem relating the 3-D Radon transform with the Pseudopolar FFT via a 1-D FFT. One this is established, 3-D analogs of Theorems 2-5 follow immediately as before.

For example, both the 3-D Radon transform and its adjoint can be computed in $O(N \log(N))$ operations, where $N = n^3$ is the number of voxels. A simple preconditioner can be applied, which is diagonal in the 3-D Pseudopolar domain. After preconditioning the condition number is tightly controlled, as a consequence of properties of the one-dimensional $G_{n,k}$ and $H_{n,k}$ resampling operators which were already discussed. Hence very simple iterative techniques allow to rapidly invert the 3-D Radon transform.

The proof of the key identity, the Projection-Slice theorem, is exactly analogous to the proof given for 2-D in Section 2 above. Generalizations to still higher dimensions are equally straightforward.

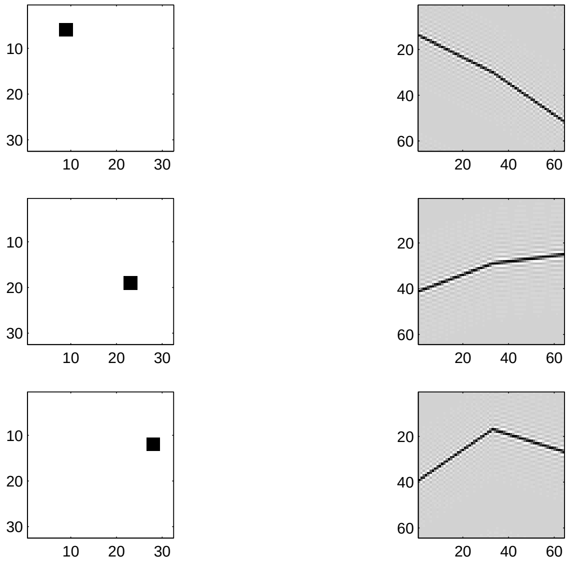


Figure 1: The Slant Stack of a Point is a Broken Line. Left Column: Images with single nonzero entry. Right Column: Corresponding Radon Transform, with break in slope at n, corresponding to $\theta = \pi/4$

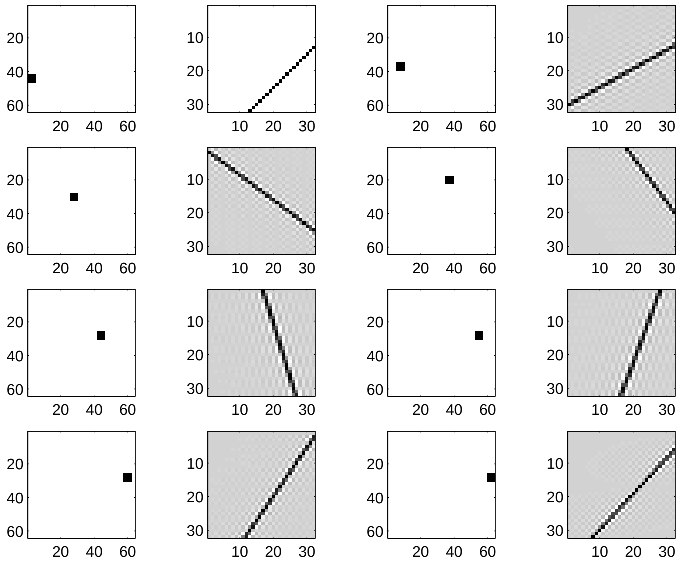


Figure 2: The Backprojection of a Point is a Line. Columns 1,3 (from left): Radon domain objects with nonzero entry at a single point. Columns 2-4. Corresponding Radon Backprojection.

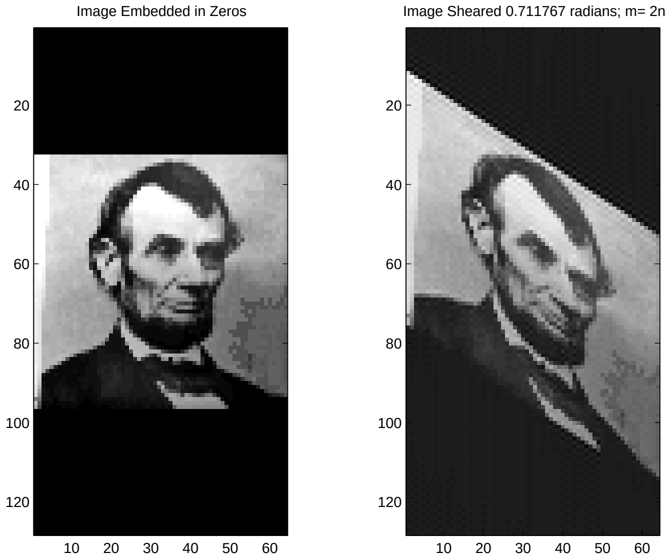


Figure 3: Shearing of an image, $m = 2n$.

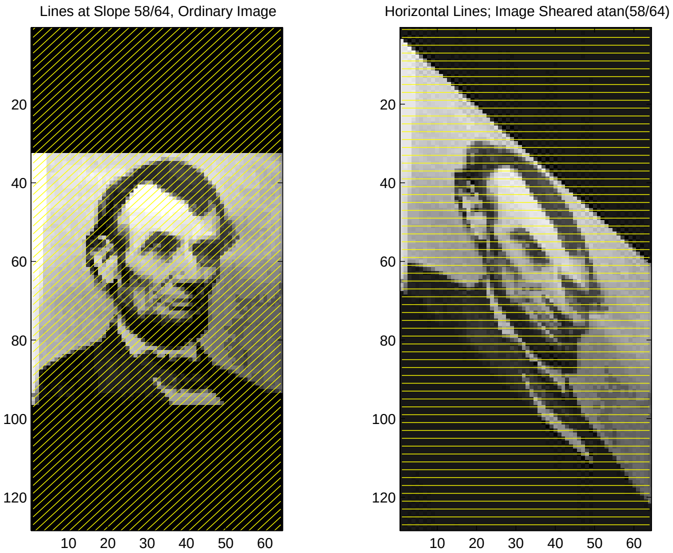


Figure 4: Summing unsheared image along slanted lines is the same as summing a sheared image along horizontal lines.

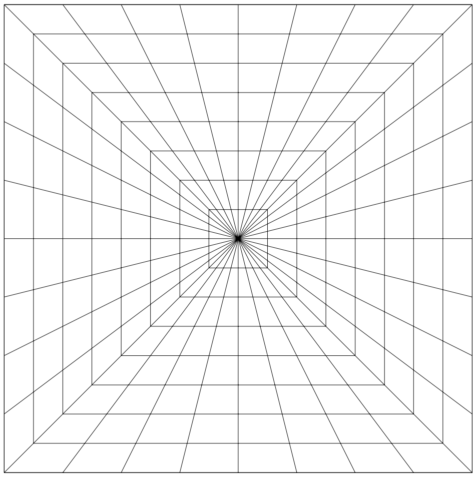


Figure 5: The Pseudopolar Grid for $n = 8$

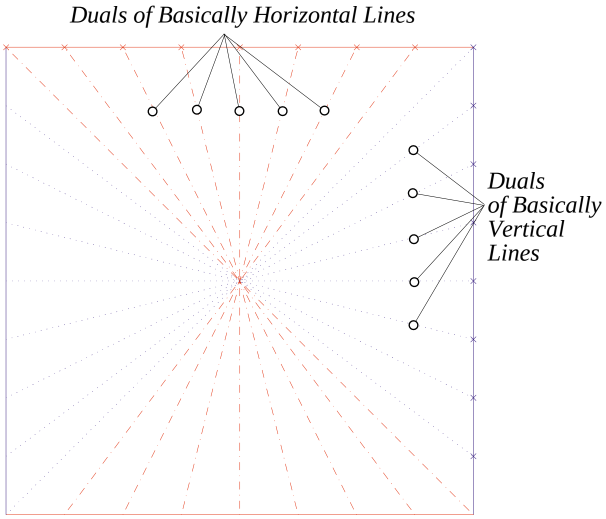

Duals of Basically Horizontal Lines
Duals of Basically Vertical Lines

Figure 6: Lines in frequency space corresponding to pseudopolar angles

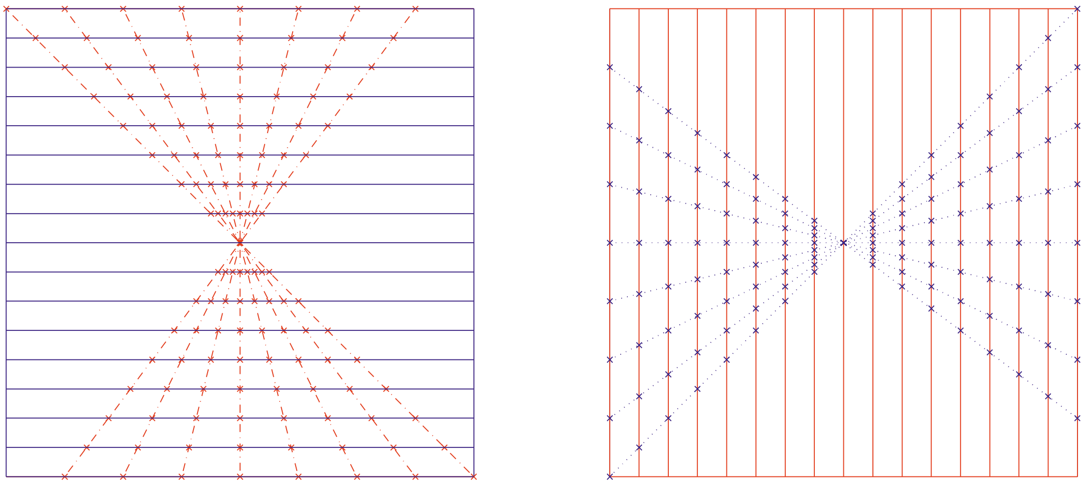


Figure 7: Definition of pseudopolar grid points. Left: Panel $s = 1$, using duals of basically horizontal lines; Right: Panel $s = 2$, using duals of basically vertical lines.

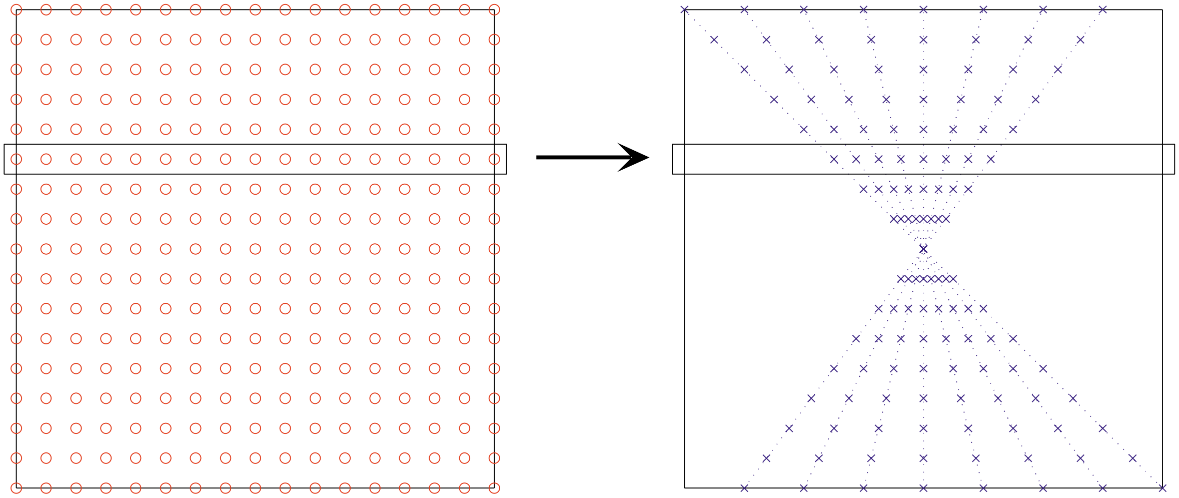


Figure 8: Converting from Cartesian Grid to Panel $s = 1$ of Pseudopolar Grid.

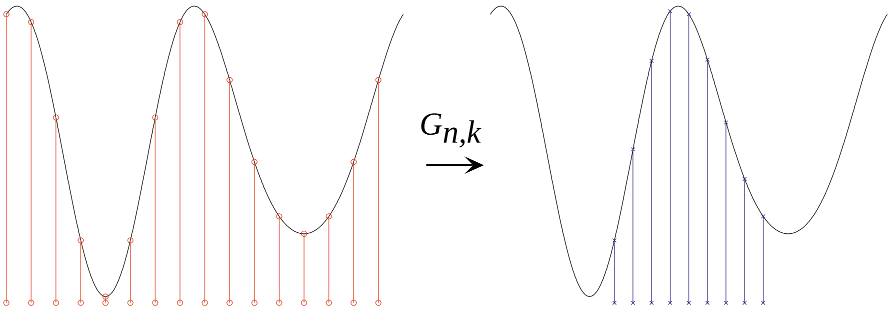


Figure 9: Operator $G_{n,k}$. Cartesian to Pseudopolar Resampling within a Single Row

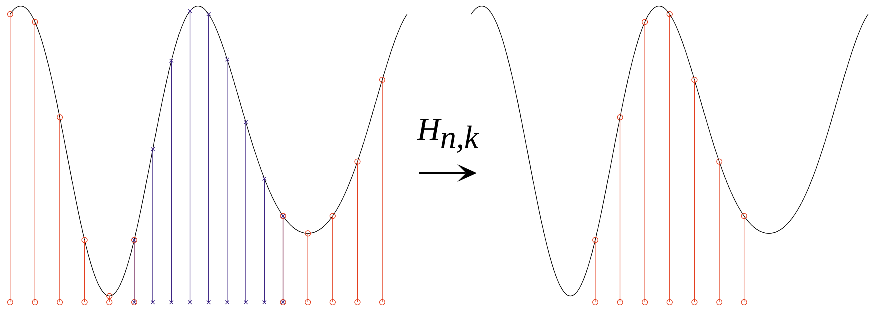


Figure 10: Operator $H_{n,k}$. Pseudopolar to Cartesian Resampling within a Single Row

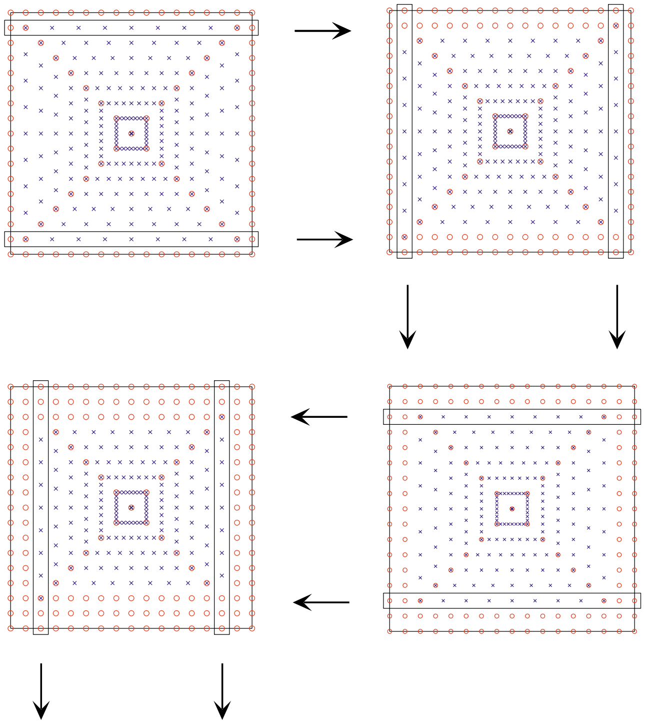


Figure 11: Recovering Cartesian points from Pseudopolar points. Starting from the outside, where the Cartesian samples are known, proceed one 'layer' at a time by sequential application of $H_{n,k}$ operators.

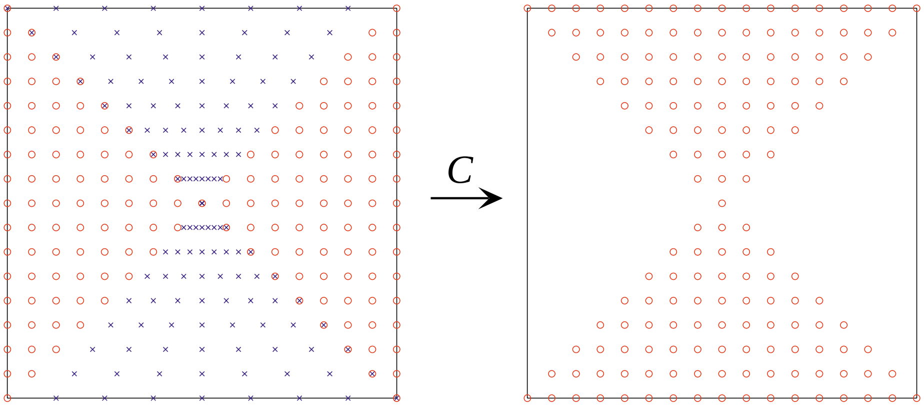


Figure 12: Resampling Effect of Operator C. Data available at ordinary Cartesian sampling points in Sector 2 and at pseudopolar sampling points in Sector 1 are used to obtain samples at Cartesian points in Sector 1.

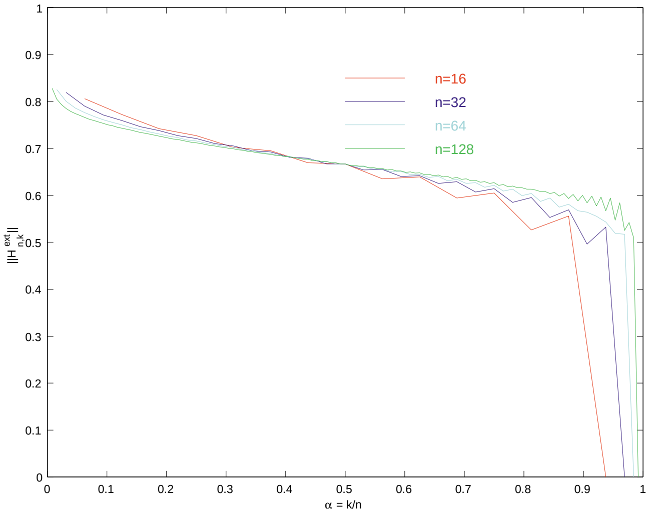


Figure 13: Norm $\|H^{ext}_{n,k}\|$ for a range of n and k

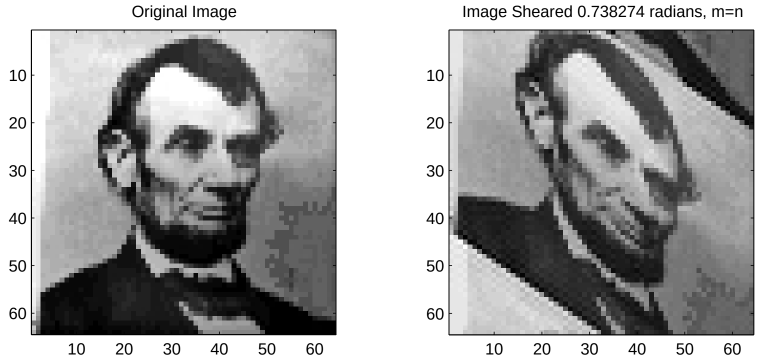


Figure 14: Image shearing with $m = n$. Notice how, in comparison to Figure 3, the columns wrap around.

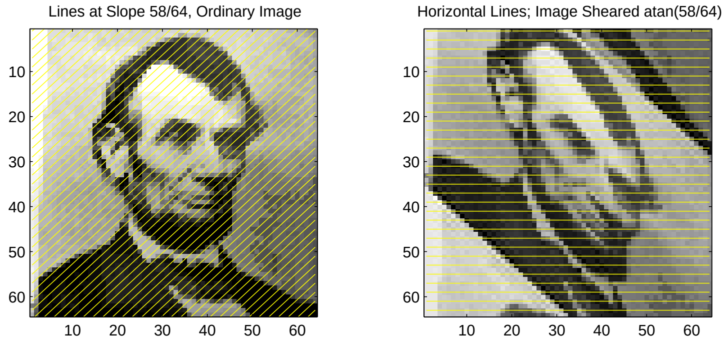
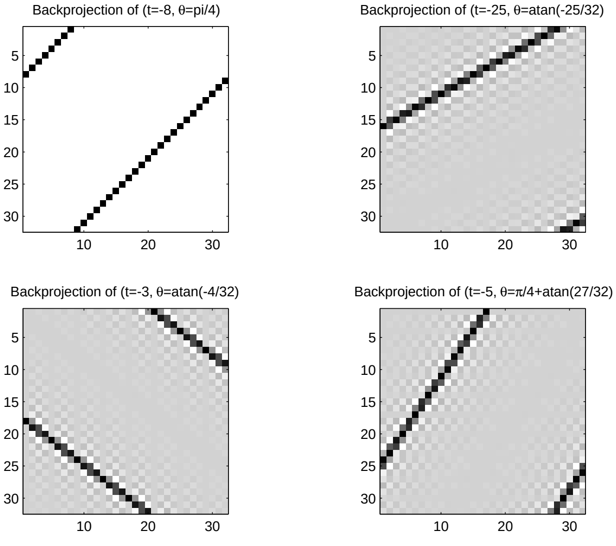


Figure 15: Summing along slanted lines with $m = n$ is the same as summing a wrapped image along horizontal lines


Backprojection of (t=-8, θ=pi/4)

Backprojection of (t=-25, θ=atan(-25/32)

Backprojection of (t=-3, θ=atan(-4/32)

Backprojection of (t=-5, θ=π/4+atan(27/32)

Figure 16: With $m = n$, backprojection of a Radon-domain array with single nonzero entry yields "lines" which exhibit wrap-around artifacts.

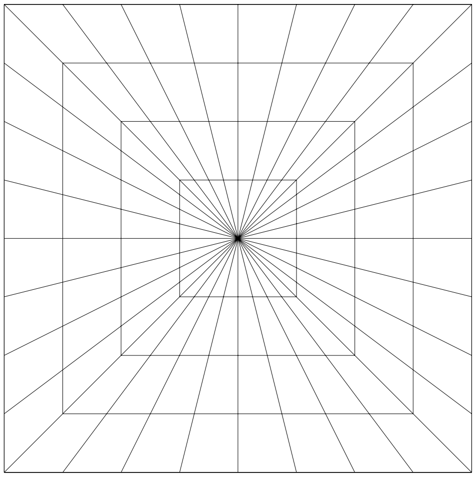


Figure 17: The Concentric Squares Grid of Mersereau and Oppenheim for $n = 8$

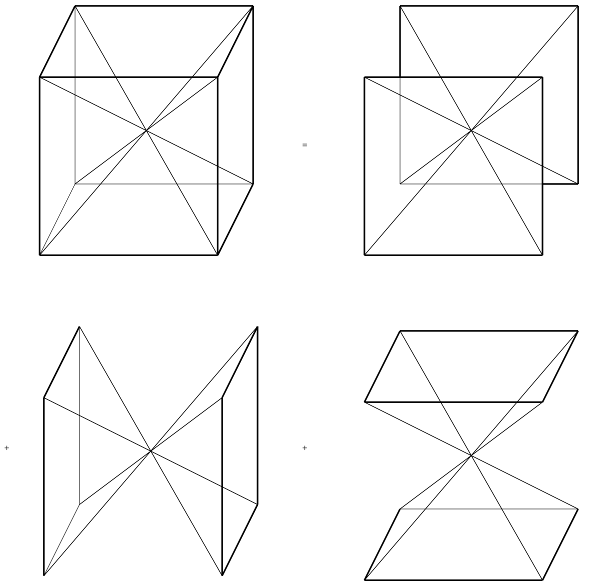


Figure 18: Decomposition of the Three-D Fourier Domain into 3 Panels

References

[1] L. Axel, G.T. Herman, D.A. Roberts, and L. Dougherty. Linogram reconstruction for magnetic resonance imaging. *IEEE Trans. Medical Imaging*, 9(4):447–449, 1990.

[2] D.H. Bailey and P. Swarztrauber. The fractional Fourier transform and applications. *SIAM Review*, 33(3):389–404, 1991.

[3] P. Swarztrauber and D.H. Bailey. Efficient detection of a continuous-wave signal with a linear frequency drift. SIAM Journal on Scientific Computing, 16(1995), pp. 1233-1239

[4] G. Beylkin. Discrete Radon Transform. *IEEE Trans. Acoustics, Speech, Signal Processing* **35** 162-

[5] L.I. Bluestein. A linear filtering approach to the computation of the discrete Fourier transform. *IEEE Trans. Audio Electroacout.*, AE-18: 451-455, 1970.

[6] M.L. Brady. A fast discrete approximation algorithm for the Radon transform. *SIAM J. Comput.*, 27(1):107–119, 1998.

[7] A. Brandt and J. Dym. Fast Calculation of Multiple Line Integrals. *SIAM J. Sci. Comput.* **20** 1417-1429.

[8] A. Brandt, J. Mann, M. Brodski, M. Galun. A Fast and Accurate Multilevel Inversion of the Radon Transform. *SIAM J. Appl. Math.* 60 (2000), no. 2, 437–462

[9] H. Choi and D.C. Munson. Direct-Fourier reconstruction in tomography and synthetic apperture radar. *Int. J. of Imaging System and Technology*, 9(1):1–13, 1998.

[10] S.R. Deans. *The Radon Transform and Some of its Applications.* Krieger Publishing, Malabar Fl. 1993.

[11] D.L. Donoho and B.F. Logan. Signal recovery and the Large Sieve. *SIAM J. Math. Anal.* **52** 577-591, 1992.

[12] P. Edholm and G. T. Herman. Linograms in image reconstruction from projections. *IEEE Trans. Medical Imaging*, MI-6(4):301–307, 1987.

[13] P. Edholm, G. T. Herman, and D. A. Roberts. Image reconstruction from linograms: Implementation and evaluation. *IEEE Trans. Medical Imaging*, MI-7(3):239–246, 1988.

[14] K. Fourmont. Schnelle Fourier-tranformation bei nichtäquidistanten Gittern und tomographische anwendungen. Ph.D Thesis, Universität Münster, 1999.

[15] Götze, W.A. and Druckmüller, H.J. (1995) A fast digital Radon transform – an efficient means for evaluating the Hough Transform. *Pattern Recognition.* **28**, 12, 1985-1992.

[16] Golub, G. and van Loan, C. (1983) *Matrix Computations.* Baltimore: Johns Hopkins University Press.

[17] Greenbaum, Anne. *Iterative Methods for Solving Linear Systems.* SIAM: Philadelphia, PA, 1997.

[18] T.C. Hsung, D.P.K. Lun, W.C. Siu The Discrete Periodic Radon Transform. *IEEE Trans. Signal Processing* **44** 2651-2657, 1996.

[19] W. Lawton. A new polar Fourier transform for computer-aided tomography and spotlight synthetic aperture radar. *IEEE Trans. Acoustics Speech Signal Process.*, 36(6):931-933, 1988.

[20] B.T. Kelley and V.K. Madisetti. The Discrete Radon Transform: Part I – Theory. *IEEE Transactions on Image Processing* **2** 382-400, 1993.

[21] F. Matus and J Flusser. Image representations via a Finite Radon Transform. *IEEE Trans. Pattern Ana. Machine Intell.* **15** 996-1006. 1993.

[22] R.M. Mersereau and A.V. Oppenheim. Digital reconstruction of multidimensional signals from their projections. *Proc. IEEE*, 62(10):1319-1338, 1974.

[23] F. Natterer. Fourier reconstruction in tomography. *Numer. Math.*, 47:343-353, 1985.

[24] J.D. O'Sullivan. A fast sinc gridding algorithm for Fourier inversion in computer tomography. *IEEE Trans. Med. Imaging*, MI-4(4):200-207, 1985.

[25] J.E. Pasciak. A note on the Fourier algorithm for image reconstruction. Preprint AMD 896 Applied Mathematics Department, Brookhaven National Laboratory, Upton, New York 11973, 1981.

[26] L.R. Rabiner, R.W. Schafer and C.M. Rader. The chirp-z transform algorithm and its applications. *Bell System Tech. J.*, 48:1249-1292, 1969.

[27] H. Schomberg and J. Timmer. The gridding method for image reconstruction by Fourier tranformation. *IEEE Trans. Med. Imag.*, MI-14(3):596-607, 1995.

[28] M. Unser, P. Thévenaz and L. Yaroslavsky. Convolution-based interpolation for fast, high-quality rotation of images. *IEEE Trans. on Image Proc.*, 4(10):1371–1381, 1995.

[29] Oz Yilmaz, *Seismic Data Processing* (SEG Investigations in Geophysics N 2.) 2000.