2016

Chebyshev Inversion of the Radon Transform

Jared Cameron Szi
University of South Carolina

Chebyshev Inversion of the Radon Transform

by

Jared Cameron Szi

Bachelor of Arts
University of California, Davis 2014

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Arts in

Mathematics

College of Arts and Sciences

University of South Carolina

2016

Accepted by:

Pencho Petrushev, Director of Thesis

Joshua Cooper, Reader

Lacy Ford, Senior Vice Provost and Dean of Graduate Studies

Abstract

In its two-dimensional form, the Radon transform of an image (function) is a collection of projections of the image which are parameterized by a set of angles (from the positive x-axis) and distances from the origin. Computational methods of the Radon transform are important in many image processing and computer vision problems, such as pattern recognition and the reconstruction of medical images. However, computability requires the construction of a discrete analog to the Radon transform, along with discrete alternatives for its inversion. In this paper, we present discrete analogs using classical methods of Chebyshev polynomial reconstruction, along with a new computational method which makes use of sub-exponentially localized frames comprised of Chebyshev polynomials. This new method leads directly to a potential new algorithm for image reconstruction using Radon inversion.

Table of Contents

List of Figures

Chapter 1

Introduction

An integral transform is any transform T of the form

$$(Tf)(u) = \int_{t_1}^{t_2} K(t,u)\, f(t)dt.$$

It is a transform in the sense that the input of the transform is a function f, and the output is yet another function Tf. While there are many integral transforms that have useful applications, one that has remained of primary interest (particularly in the field of tomography) is the Radon transform.

Let $f \in C_0^1(\mathbb{R}^2)$, the space of compactly supported, continuously differentiable functions. The Radon transform of f is defined to be a function on the space of straight lines in $\mathbb{R}^2$,

$$Rf(L) = \int_L f(\mathbf{x})d\mathbf{x}. \tag{1.1}$$

Any straight line L in $\mathbb{R}^2$ can be parameterized in the form

$$L = (s\cos\phi + t\sin\phi, -s\sin\phi + t\cos\phi), \quad s \in \mathbb{R},$$

where t is the distance of L from the origin and ϕ is angle between L and the positive x-axis. Using the following notation,

$$\theta = (\cos\phi, \sin\phi), \quad \theta^{\perp} = (-\sin\phi, \cos\phi), \tag{1.2}$$

it follows that (1.1) can be expressed as

$$R_\theta f(s) = R_f(\theta, s) = \int_{\mathbb{R}} f(s\theta + t\theta^{\perp})dt. \tag{1.3}$$

As we've alluded to, the Radon transform is used in a variety of applications in the field of tomography, which is concerned with the reconstruction of an image from cross-sectional projection data of an object. In this manner, the function f can be thought of as an unknown density function (i.e. density of pixels in a region). The Radon transform then represents projection data obtained from a tomographic scan, such as an X-ray (see Figure 1.2). Therefore, the inversion of the Radon transform may be used to recreate the original density function from the projection data.

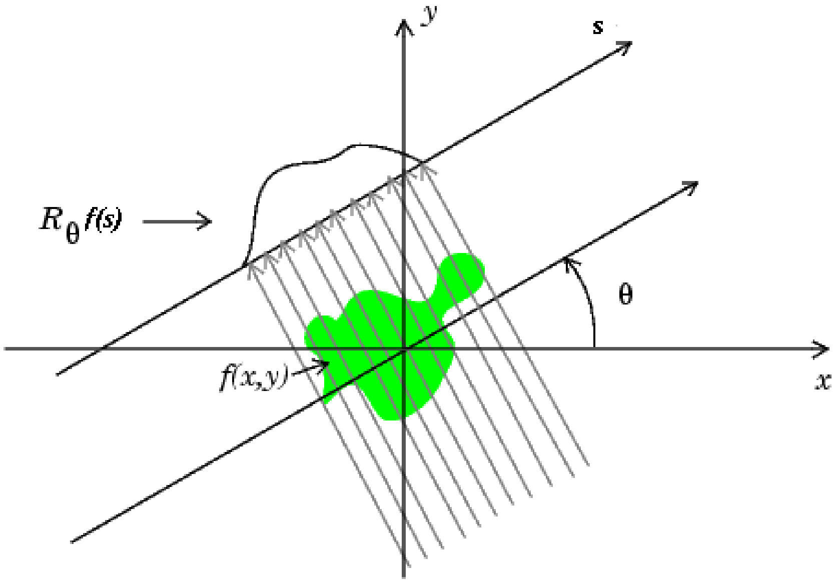


Figure 1.1: A visualization of the Radon transform.

The focus of this paper will remain on inversion formulas for the Radon transform. Chapter 2 focuses on two classical inversion formulas for the Radon transform in the continuous case. These results make use of Chebyshev polynomials and the Fourier transform, and a more concise version of their constructions can be found in [10].

Chapter 3 is comprised of two sections. The first of which will focus on developing a discretization of the Chebyshev inversion formula presented in Chapter 2, a result which can also be found in [10] (albeit in briefer form). The second section of Chapter 3 provides a semi-rigorous explanation behind algorithms for capturing the projection data given by the Radon transform, an absolutely necessary consideration for the development of any reconstruction algorithm. Many such algorithms exist (see [3], [6], and [12]), but the explanations provided are more loosely based on algorithms found in [7] and [9].

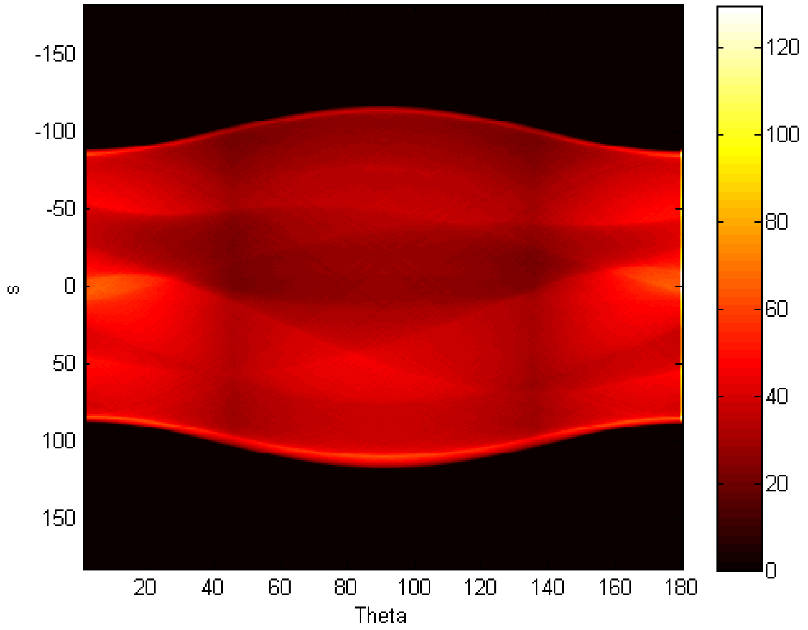


Figure 1.2: A visualization of projection data, also called a sinogram.

In Chapter 4, we present in rigorous detail some localization properties of specialized types of kernels and frames which are used in the construction of a new Radon inversion formula to be presented in Chapter 5. These properties are explored in a more general sense in [8] and [11], though our goals only require us to consider a specific case. Finally, in Chapter 5, we present in the first section the aforementioned new Radon inversion formula based on the results of Chapter 4. The second section of Chapter 5 includes the associated discretization, which thereby provides the computational framework necessary to lead to a new potential reconstruction algorithm.

Overall, this paper highlights the importance of the development of inversion formulas for the Radon transform. Numerical methods rely on discrete versions of approximate results, but the approximation of inversions rely on the theoretical existence of analytic forms. Hence, to put it simply, theoretical underpinnings for inversions must be put in place before numerical methods can be developed for industrial applications.

Chapter 2

The Continuous Radon Transform

2.1 Classical Chebyshev Inversion Formula

In this section, we will present an inversion formula for the Radon transform based on the so-called Ridge Chebyshev Polynomials of the Second Kind. For the sake of simplicity, it will be useful to use the following notation in this section, as opposed to the notation given by (1.2):

$$\mathbf{e}_\alpha = (\cos\alpha, \sin\alpha), \quad \alpha \in \mathbb{R}. \tag{2.1}$$

Definition 2.1. *Let $f, g \in L_2(\mathcal{D})$. Then, we define the inner product to be*

$$\langle f, g\rangle := \frac{1}{\pi}\int\int_{\mathcal{D}} f(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x}.$$

Definition 2.2. *Let m be a positive integer. Then, the Chebyshev Polynomials of the First Kind are defined by*

$$T_m(t) := \cos(m \arccos t). \tag{2.2}$$

Definition 2.3. *Similarly, we define the Chebyshev Polynomials of the Second Kind by*

$$U_m(t) := \frac{\sin(m \arccos t)}{\sin(\arccos t)}. \tag{2.3}$$

Lemma 2.4. *Let m be a positive integer. Then,*

$$U_m \in \Pi_{m-1}, \tag{2.4}$$

where Π_{m-1} is the space of polynomials of degree less than or equal to $m-1$.

Proof. Note that $(T_m)' = U_m$. Moreover, it is known that $T_m \in \Pi_m$. Therefore, $U_m \in \Pi_{m-1}$. □

Definition 2.5. *For any positive integer m, define*

$$\Omega_m := \left\{ \frac{k\pi}{m} : k = 1, \ldots, m \right\}.$$

The collection $\{U_m(\mathbf{x} \cdot \mathbf{e}_{\alpha_k})\}_{k=1}^{m}$, where $\alpha_k = \frac{\pi k}{m} \in \Omega_m$ for $k = 1, 2, \ldots, m$, is called a collection of Ridge Chebyshev Polynomials. To prove orthonormality, it will be beneficial to make use of rotations of the Ridge Chebyshev Polynomials.

Lemma 2.6. *Let m, n be positive integers and let $\alpha_1, \alpha_2 \in \mathbb{R}$. Then, for all $\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2$,*

$$\langle U_m \left(\boldsymbol{x} \cdot \boldsymbol{e}_{\alpha_1} \right), U_n \left(\boldsymbol{x} \cdot \boldsymbol{e}_{\alpha_2} \right) \rangle = \langle U_m \left(\boldsymbol{x} \cdot \boldsymbol{e}_{\phi} \right), U_n \left(\boldsymbol{x} \cdot \boldsymbol{e}_0 \right) \rangle,$$

where $\phi = \alpha_1 - \alpha_2$ and $\boldsymbol{u} = (u_1, u_2) \in \mathbb{R}^2$.

Proof. By definition,

$$\langle U_m(\mathbf{x} \cdot \mathbf{e}_{\alpha_1}), U_n(\mathbf{x} \cdot \mathbf{e}_{\alpha_2}) \rangle = \frac{1}{\pi} \int \int_{\mathcal{D}} U_m(\mathbf{x} \cdot \mathbf{e}_{\alpha_1}) U_n(\mathbf{x} \cdot \mathbf{e}_{\alpha_2}) d\mathbf{x}.$$

Explicitly writing the dot products, the foregoing becomes

$$\frac{1}{\pi} \int \int_{\mathcal{D}} U_m(x_1 \cos\alpha_1 + x_2 \sin\alpha_1) U_n(x_1 \cos\alpha_2 + x_2 \sin\alpha_2) dx_2 dx_1. \tag{2.5}$$

We will use a change of variables to rotate the coordinate system by an angle of α_2.

Let $x_1 = u_1 \cos\alpha_2 - u_2 \sin\alpha_2$ and $x_2 = u_1 \sin\alpha_2 + u_2 \cos\alpha_2$. Then, the Jacobian corresponding to this change of variables, which is given by

$$J = \begin{bmatrix} \cos\alpha_2 & -\sin\alpha_2 \\ \sin\alpha_2 & \cos\alpha_2 \end{bmatrix},$$

has a determinant of 1. Now, performing our substitution, we see that

$$x_1 \cos\alpha_1 + x_2 \sin\alpha_1 = u_1 \cos\phi + u_2 \sin\phi, \quad x_1 \cos\alpha_2 + x_2 \sin\alpha_2 = u_1. \tag{2.6}$$

Since the Jacobian matrix corresponding to the change of variables has determinant 1, the integration factor for this change of variables is 1. This fact coupled with the observations given by (2.6), we observe that (2.5) becomes

$$\frac{1}{\pi}\int_{-1}^{1}\int_{-\sqrt{1-u_1^2}}^{\sqrt{1-u_1^2}} U_m(u_1\cos(\phi)+u_2\sin(\phi))U_n(u_1)du_2du_1.$$

Therefore, by writing $\mathbf{u}=(u_1,u_2)$, the foregoing becomes

$$\frac{1}{\pi}\int\int_{\mathcal{D}} U_m(\mathbf{u}\cdot\mathbf{e}_\phi)U_n(\mathbf{u}\cdot\mathbf{e}_0)du_2du_1,$$

and so we achieve our desired result. That is,

$$\langle U_m\left(\mathbf{x}\cdot\mathbf{e}_{\alpha_1}\right), U_n\left(\mathbf{x}\cdot\mathbf{e}_{\alpha_2}\right)\rangle = \langle U_m\left(\mathbf{u}\cdot\mathbf{e}_{\phi}\right), U_n\left(\mathbf{u}\cdot\mathbf{e}_{0}\right)\rangle,$$

where $\phi=\alpha_1-\alpha_2$, $\mathbf{u}=(u_1,u_2)\in\mathbb{R}^2$. □

We will now move on to the proof of orthonormality of the Ridge Chebvyshev Polynomials of the Second Kind.

Theorem 2.7. *The Ridge Chebyshev Polynomials of the Second Kind are orthonormal in $L_2(\mathcal{D})$.*

Proof. Let m,n be positive integers and $\alpha_i,\alpha_j\in\Omega_m$, where $1\le i,j\le m$. Assume, without loss of generality, that $i>j$. Let $\phi=\alpha_i-\alpha_j$. By Lemma 2.6, it follows that

$$\langle U_m(\mathbf{x}\cdot\mathbf{e}_{\alpha_i}), U_n(\mathbf{x}\cdot\mathbf{e}_{\alpha_j})\rangle = \langle U_m(\mathbf{x}\cdot\mathbf{e}_{\phi}), U_n(\mathbf{x}\cdot\mathbf{e}_{0})\rangle.$$

Now, observe that

$$\langle U_m(\mathbf{x}\cdot\mathbf{e}_{\phi}), U_n(\mathbf{x}\cdot\mathbf{e}_{0})\rangle = \frac{1}{\pi}\int_{-1}^{1}\left(\int_{-\sqrt{1-x_1^2}}^{\sqrt{1-x_1^2}} U_m(x_1\cos\phi+x_2\sin\phi)dx_2\right)U_n(x_1)dx_1.$$

Performing the substituion $x_1=\cos\tau$, it follows that $dx_1=-\sin\tau d\tau$ and so the foregoing becomes

$$\frac{1}{\pi}\int_{0}^{\pi}\left(\int_{-\sin\tau}^{\sin\tau} U_m(\cos\tau\cos\phi+x_2\sin\phi)dx_2\right)U_n(\cos\tau)\sin\tau d\tau.$$

Subsequently setting $u = \cos\tau\cos\phi + x_2\sin\phi$, it follows that $dx_2 = \frac{du}{\sin\phi}$ and so we obtain

$$\frac{1}{\pi\sin\phi}\int_0^\pi \left(\int_{\cos(\phi+\tau)}^{\cos(\phi-\tau)} U_m(u)du\right) U_n(\cos\tau)\sin\tau d\tau.$$

Making yet another substitution, we let $u = \cos t$. Then, $du = -\sin t dt$ and we achieve

$$\frac{1}{\pi\sin\phi}\int_0^\pi \left(\int_{\phi-\tau}^{\phi+\tau} U_m(\cos t)\sin t dt\right) U_n(\cos\tau)\sin\tau d\tau.$$

Applying (2.3) to the previous expression yields

$$\frac{1}{\pi\sin\phi}\int_0^\pi \left(\int_{\phi-\tau}^{\phi+\tau} \sin(mt)dt\right)\sin(n\tau)d\tau$$

and so we have that

$$\langle U_m(\mathbf{x}\cdot\mathbf{e}_\phi), U_n(\mathbf{x}\cdot\mathbf{e}_0)\rangle = \frac{2\sin(m\phi)}{m\pi\sin\phi}\int_0^\pi \sin(m\tau)\sin(n\tau)d\tau.$$

We now consider two cases.

- Case 1: Suppose $m \neq n$. Then,

$$\begin{aligned}\langle U_m(\mathbf{x}\cdot\mathbf{e}_\phi), U_n(\mathbf{x}\cdot\mathbf{e}_0)\rangle &= \frac{2\sin(m\phi)}{m\pi\sin\phi}\int_0^\pi \sin(m\tau)\sin(n\tau)d\tau \\ &= \frac{2\sin(m\phi)}{m\pi\sin\phi}\left[\int_0^\pi \cos\left((m-n)\tau\right) - \cos\left((m+n)\tau\right)d\tau\right] \\ &= 0.\end{aligned}$$

  Hence, from Lemma 2.6, it follows that

$$\langle U_m(\mathbf{x}\cdot\mathbf{e}_{\alpha_i}), U_n(\mathbf{x}\cdot\mathbf{e}_{\alpha_j})\rangle = 0,$$

  whenever $m \neq n$.

- Case 2: Suppose $m = n$. Then,

$$\begin{aligned}\langle U_m(\mathbf{x}\cdot\mathbf{e}_\phi), U_n(\mathbf{x}\cdot\mathbf{e}_0)\rangle &= \frac{2\sin(m\phi)}{m\pi\sin\phi}\int_0^\pi \sin^2(n\tau)d\tau \\ &= \frac{\sin(m\phi)}{m\sin\phi}. \qquad (2.7)\end{aligned}$$

- Subcase 1: Suppose $\phi = 0$, then (2.7) is undefined. Taking a limit tending toward $\phi = 0$ yields an indeterminate form. However, by using L'Hopital's rule, we find that (2.7) becomes

$$\langle U_m(\mathbf{x} \cdot \mathbf{e}_\phi), U_n(\mathbf{x} \cdot \mathbf{e}_0)\rangle = 1.$$

  Therefore, it follows from Lemma 2.6 that

$$\langle Um(\mathbf{x} \cdot \mathbf{e}_{\alpha_i}), U_n(\mathbf{x} \cdot \mathbf{e}_{\alpha_j})\rangle = 1,$$

  whenever $m = n$ and $i = j$.

- Subcase 2: Now, suppose $\phi \neq 0$. Then, since $\phi = \alpha_i - \alpha_j$, it follows that

$$\langle U_m(\mathbf{x} \cdot \mathbf{e}_\phi), U_n(\mathbf{x} \cdot \mathbf{e}_0)\rangle = \frac{\sin((i-j)\pi)}{m \sin \phi} = 0.$$

  Hence, from Lemma 2.6, we have that

$$\langle U_m(\mathbf{x} \cdot \mathbf{e}_{\alpha_i}), U_n(\mathbf{x} \cdot \mathbf{e}_{\alpha_j})\rangle = 0,$$

  whenever $m = n$ and $i \neq j$.

To summarize, suppose we have a collection $\{U_m(\alpha_i)\}_{i=1}^m$ and $\alpha_i \in \Omega_m$ for all $i = 1, 2, \ldots, m$. Then, for all $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$,

$$\langle U_m(\mathbf{x} \cdot \mathbf{e}_{\alpha_i}), U_n(\mathbf{x} \cdot \mathbf{e}_{\alpha_j})\rangle = \begin{cases} 0 & \text{if } m \neq n \\ 0 & \text{if } m = n \text{ and } i \neq j, \\ 1 & \text{if } m = n \text{ and } i = j. \end{cases} \tag{2.8}$$

□

Using a dimensionality argument, we will now show that the Ridge Chebyshev Polynomials of degree up to $m-1$ span the space of polynomials Π_{m-1} . From here, we will see that the Ridge Chebyshev Polynomials are dense in $L_2(\mathcal{D})$. Coupling this with their orthonormal properties will then provide the conclusion that the Ridge Chebyshev Polynomials form an orthonormal basis for the space $L_2(\mathcal{D})$.

Theorem 2.8. *Let $N \geq 1$. Then, the Ridge Chebyshev Polynomials of degree up to $N-1$ span the space of polynomials Π_{N-1}.*

Proof. For each $N \geq 1$, define

$$\mathcal{U}_N := \left\{ f \in L_2(\mathcal{D}) : f = \sum_{m=1}^{N} \sum_{\theta \in \Omega_m} a_m(\theta) U_m(\bullet \cdot \mathbf{e}_\theta) \right\},$$

where

$$a_m(\theta) := \langle f, U_m(\bullet \cdot \mathbf{e}_\theta) \rangle. \tag{2.9}$$

Now, it follows that

$$\begin{aligned} \dim \mathcal{U}_N &= \sum_{m=1}^{N} \#\Omega_m \\ &= \frac{N(N+1)}{2}, \end{aligned}$$

where $\#\Omega_m$ denotes the cardinality. Therefore, the dimension of the spaced spanned by the Ridge Chebyshev Polynomials (of degree up to $N-1$) is $N(N+1)/2$. A simple counting argument shows that

$$\dim \Pi_{N-1} = \frac{N(N+1)}{2}.$$

From (2.4), it follows that the space spanned by the Ridge Chebyshev Polynomials of degree up to $N-1$ is a subspace of the space spanned by Π_{N-1}. However, the dimensions of each space are the same and so it follows that the Ridge Chebyshev Polynomials span Π_{N-1}. □

Theorem 2.9. *Let $f \in L_2(\mathcal{D})$. Then,*

$$f = \sum_{m=1}^{\infty} \sum_{\theta \in \Omega_m} \left(\int_{\mathbb{R}} R_\theta f(s) U_m(s) ds \right) U_m(\bullet \cdot \boldsymbol{e}_\theta). \tag{2.10}$$

Proof. From Theorem 2.7, it follows that the Ridge Chebyshev Polynomials form an orthonormal system on $L_2(\mathcal{D})$. Theorem 2.8 tells us that for each $N \geq 1$, the space $\mathcal{U}_N$ spans the space Π_{N-1}, and being that $\mathcal{U}_N \subseteq \Pi_{N-1}$, it follows then that

$\mathcal{U}_N = \Pi_{N-1}$. The well-known Weierstrass Approximation theorem asserts that the space

$$\bigcup_{N=1}^{\infty} \Pi_{N-1}$$

is dense in $L_2(\mathcal{D})$. Hence, as an immediate corollary, the Ridge Chebyshev polynomials are dense in $L_2(\mathcal{D})$. Combining these results, we see that the Ridge Chebyshev polynomials form an orthonormal basis for $L_2(\mathcal{D})$.

Thus, following a similar construction scheme as seen in the famous proof of the Weierstrass Approximation Theorem using Bernstein Polynomials, it follows that any $f \in L_2(\mathcal{D})$ can be expressed as

$$f(\mathbf{x}) = \sum_{m=1}^{\infty} \sum_{\theta \in \Omega_m} a_m(\theta) U_m(\mathbf{x} \cdot \mathbf{e}_\theta), \tag{2.11}$$

where $a_m(\theta)$ is defined as in (2.9). Therefore, we have that

$$a_m(\theta) = \int\int_{\mathbb{R}^2} f(x_1, x_2) U_m(x_1 \cos\alpha + x_2 \sin\alpha) dx_1 dx_2. \tag{2.12}$$

Using the change of variables $x_1 = s\cos\alpha - t\sin\alpha$ and $x_2 = s\sin\alpha + t\cos\alpha$, it follows that (2.12) becomes

$$a_m(\theta) = \int_{\mathbb{R}} \left(\int_{\mathbb{R}} f(s\mathbf{e}_\theta + t\mathbf{e}_\theta^\perp) dt \right) U_m(s) ds.$$

Therefore, from (1.3), we obtain that

$$a_m(\theta) = \int_{\mathbb{R}} R_\theta f(s) U_m(s) ds$$

and by subsequently applying the foregoing result to (2.11), it follows that

$$f = \sum_{m=1}^{\infty} \sum_{\theta \in \Omega_m} \left(\int_{\mathbb{R}} R_\theta f(s) U_m(s) ds \right) U_m(\bullet \cdot \mathbf{e}_\theta).$$

□

This concludes the proof of the first of our two inversion formulas. This particular inversion formula is most relevant to the discussions of this paper, as we will construct an additional inversion formula in Chapter 5 which is also based on algebraic reconstruction using the Chebyshev Polynomials of the Second Kind. However, to give the reader a different flavor of the sort of inversion formulas which exist in classical literature, we present in the next section an inversion formula for the Radon transform based on the Fourier and Hilbert transforms. While the proof is given in the two-dimensional case, these particular inversion formulas generalize rather nicely to the n-dimensional case (and thus, this highlights a fundamental advantage over the previous inversion formula).

2.2 Classical Fourier Inversion Formula

In this section, we will introduce an inversion formula for the Radon transform based on the Fourier and Hilbert transforms.

Definition 2.10. *Let $h : \mathbb{R} \to \mathbb{R}$. Then, the Hilbert transform of h is defined to be*

$$\begin{aligned}(Hh)(x) &:= p.v. \int_{-\infty}^{\infty} \frac{h(y)}{x-y} dy \qquad (2.13)\\ &= \lim_{\varepsilon \to 0} \int\limits_{|y-x| \geq \varepsilon} \frac{h(y)}{x-y} dy, \quad x \in \mathbb{R}, \qquad (2.14)\end{aligned}$$

where p.v. stands for the principal value.

Definition 2.11. *Let $f \in C_0^\infty(\mathbb{R}^n)$. Then, the Fourier transform of f is defined to be*

$$Ff(\xi) := \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} f(\boldsymbol{x}) e^{-i\boldsymbol{x}\cdot\xi} d\boldsymbol{x}, \quad \xi \in \mathbb{R}^n. \qquad (2.15)$$

We sometimes use the equivalent notation

$$Ff(\xi) = \hat{f}(\xi). \qquad (2.16)$$

Definition 2.12. *Let $f \in C_0^\infty(\mathbb{R}^n)$. Then, we define the Inverse Fourier transform of f to be*

$$F^{-1}f(\boldsymbol{x}) := \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} f(\boldsymbol{x}) e^{i\boldsymbol{x}\cdot\xi} d\xi, \quad \boldsymbol{x} \in \mathbb{R}^n. \qquad (2.17)$$

Again, we sometimes use the equivalent notation

$$F^{-1}f(\boldsymbol{x}) = \check{f}(\boldsymbol{x}). \tag{2.18}$$

One particular property of the Fourier and Inverse Fourier transforms that will become useful is that for particularly well-behaved functions, the Inverse Fourier transform of the Fourier transform of a function is the function itself. We elect to omit the proof, but state the result here for future reference.

Lemma 2.13. *Let $f \in C_0^\infty(\mathbb{R}^n)$. Then,*

$$\left(\hat{f}\right)^{\check{}} = \left(\check{f}\right)^{\hat{}} = f. \tag{2.19}$$

In order to prove our inversion formula, we will first need a few important results. The first of which is often referred to as the Fourier Slice Theorem.

Theorem 2.14. *Let $f \in C_0^\infty(\mathbb{R}^2)$ and let $\theta, \theta^\perp$ be defined as in (1.2). Then,*

$$\widehat{R_\theta f(\sigma)} = \sqrt{2\pi}\hat{f}(\sigma\theta), \quad \sigma \in \mathbb{R}.$$

Proof. Consider the Fourier transform (with respect to the variable s) of the Radon transform (in the direction of θ) of f,

$$F_s\left(R_\theta f\right)(\sigma) = \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\left(R_\theta f\right)(s)e^{-is\sigma}ds.$$

Applying (1.3), from the preceding expression we obtain

$$\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\int_{\mathbb{R}} f(s\theta + t\theta^\perp)dte^{-is\sigma}ds.$$

Letting $\mathbf{x} = s\theta + t\theta^\perp$, we obtain

$$\frac{1}{\sqrt{2\pi}}\int\int_{\mathbb{R}^2} f(\mathbf{x})e^{-i\theta\cdot\mathbf{x}\sigma}d\mathbf{x}.$$

A simple rearrangement yields

$$\frac{1}{\sqrt{2\pi}}\int\int_{\mathbb{R}^2} f(\mathbf{x})e^{-\mathbf{x}\cdot\sigma\theta}.$$

Thus, we achieve our desired result:

$$\widehat{R_\theta f(\sigma)} = \sqrt{2\pi}\hat{f}(\sigma\theta), \quad \sigma \in \mathbb{R}.$$

□

We will also need this important relationship between the Fourier transform and the Hilbert transform of a function, which states that the Fourier transform of the Hilbert transform of a function is related to the Fourier transform of the function. As this is a well-known result, the proof is omitted but the result is stated.

Lemma 2.15. *Let $h \in C_0^\infty(\mathbb{R}^2)$. Then,*

$$\widehat{(Hh)}(\xi) = (-i\mathrm{sign}\,\xi)\hat{h}(\xi). \tag{2.20}$$

We now begin with the primary result of this section.

Theorem 2.16. *Let $f \in C_0^\infty(\mathbb{R}^2)$ and $\theta, \theta^\perp$ be defined as in (1.2). Then,*

$$f(\boldsymbol{x}) = \frac{1}{2(2\pi)^{3/2}}\int_0^{2\pi}\left(H\left(R_\theta f\right)'\right)(\theta\cdot\boldsymbol{x})d\phi \tag{2.21}$$

Proof. From (2.19), we obtain

$$f(\mathbf{x}) = \left(\hat{f}\right)^{\check{}} = \frac{1}{2\pi}\int_{\mathbb{R}^2}\hat{f}(\xi)e^{i\mathbf{x}\cdot\xi}d\xi. \tag{2.22}$$

Expressing ξ in terms of polar coordinates,

$$\xi = (\rho\cos\phi, \rho\sin\phi),$$

it follows that (2.22) becomes

$$\frac{1}{2\pi}\int_0^{2\pi}\int_0^{\infty}\hat{f}\left(\rho\cos\alpha,\rho\sin\alpha\right)\rho e^{i\rho\theta\cdot\mathbf{x}}d\rho d\phi.$$

In the notation of (1.2), we obtain

$$f(\mathbf{x})=\frac{1}{2\pi}\int_0^{2\pi}\int_0^{\infty}\hat{f}\left(\rho\theta\right)\rho e^{i\rho\theta\cdot\mathbf{x}}d\rho d\phi. \tag{2.23}$$

Now, from (2.14), we know that

$$\hat{f}(\rho\theta)=\frac{1}{\sqrt{2\pi}}\widehat{R_\theta f(\rho)}. \tag{2.24}$$

Therefore, (2.23) can be expressed as

$$f(\mathbf{x})=\frac{1}{(2\pi)^{3/2}}\int_0^{2\pi}\int_0^{\infty}\widehat{R_\theta}f(\rho)\rho e^{i\rho\theta\cdot\mathbf{x}}d\rho d\phi. \tag{2.25}$$

Note that

$$R_\theta f(\rho)=R_{-\theta}f(-\rho), \tag{2.26}$$

by observing that the line integral over a line defined by a fixed distance ρ from the origin in the direction of $\theta=(\cos\phi,\sin\phi)$ is the same as the line integral over a line defined by the fixed distance $-\rho$ from the origin in the direction $-\theta=(-\cos\phi,-\sin\phi)$. In this sense, the Radon transform is an even function. Moreover, by a similar argument, the Radon transform is 2π periodic. That is,

$$\widehat{R_\theta}f(\rho)=\widehat{R_{\theta+2\pi}}f(\rho). \tag{2.27}$$

By applying (2.26) to (2.24), we obtain

$$\hat{f}(\rho\theta)=\frac{1}{\sqrt{2\pi}}\widehat{R_{-\theta}}f(-\rho)$$

Through the application of this result to (2.25), it follows that

$$f(\mathbf{x})=\frac{1}{(2\pi)^{3/2}}\int_0^{2\pi}\int_0^{\infty}\widehat{R_{-\theta}}f(-\rho)\rho e^{i\rho\theta\cdot\mathbf{x}}drd\phi. \tag{2.28}$$

Letting $r = -\rho$ and $\omega = -\theta$, this change of variables leads us to

$$\frac{1}{(2\pi)^{3/2}} \int_0^{2\pi} \int_{-\infty}^{0} \widehat{R_\omega f}(r)(-r)e^{-ir\theta\cdot\mathbf{x}} dr d\phi.$$

However, since the Radon function is an even function, the foregoing results in

$$\frac{1}{(2\pi)^{3/2}} \int_0^{2\pi} \int_{-\infty}^{0} \widehat{R_{-\omega} f}(-r)(-r)e^{-ir\theta\cdot\mathbf{x}} dr d\phi.$$

Therefore, an equivalent expression for f is

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{3/2}} \int_0^{2\pi} \int_{-\infty}^{0} \widehat{R_\theta f}(\rho)\rho e^{-i\rho\theta\cdot\mathbf{x}} d\rho d\phi.$$

By adding (2.25) to the previous result, then subsequently dividing both sides by two, we achieve

$$f(\mathbf{x}) = \frac{1}{2(2\pi)^{3/2}} \int_0^{2\pi} \int_{\mathbb{R}} \widehat{R_\theta f}(\rho)|\rho| e^{i\rho\theta\cdot\mathbf{x}} d\rho d\phi. \tag{2.29}$$

Now, $\rho = |\rho| \operatorname{sign} \rho$ and so the right hand side of the previous equation becomes

$$\frac{1}{2(2\pi)^{3/2}} \int_0^{2\pi} \int_{\mathbb{R}} \widehat{R_\theta f}(\rho)\rho(\operatorname{sign} \rho) e^{i\rho\theta\cdot\mathbf{x}} d\rho d\phi.$$

Since $-i^2 = 1$, it follows that this expression is equivalent to

$$f(\mathbf{x}) = \frac{1}{2(2\pi)^{3/2}} \int_0^{2\pi} \int_{\mathbb{R}} (-i \operatorname{sign} \rho) \left[(i\rho)\widehat{R_\theta f}(\rho)\right] e^{i\rho\theta\cdot\mathbf{x}} d\rho d\phi. \tag{2.30}$$

Observe that for any function $h \in C_0^\infty(\mathbb{R})$,

$$\widehat{h'}(\tau) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} h'(x) e^{-ix\tau} dx, \quad \tau \in \mathbb{R}.$$

Then, by moving $h'(x)$ into the differential and subsequently performing integration by parts, it follows that

$$\begin{aligned} \widehat{h'}(\tau) &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{-ix\tau} dh(x) \\ &= -\frac{1}{\sqrt{2\pi}}(-i\tau) \int_{\mathbb{R}} h(x) e^{-x\tau} dx \\ &= (i\tau)\hat{h}(\tau). \end{aligned}$$

Therefore, for any function $h \in C_0^\infty(\mathbb{R})$,

$$\widehat{h'}(\tau) = (i\tau)\hat{h}(\tau).$$

Applying this result to (2.30) yields

$$f(\mathbf{x}) = \frac{1}{2(2\pi)^{3/2}} \int_0^{2\pi} \int_{\mathbb{R}} (-i2\rho) F_\rho \left((R_\theta f)' \right) e^{i\rho\theta\cdot\mathbf{x}} d\rho d\phi,$$

where $(R_\theta f)'$ denotes the derivative with respect to ρ. Now, from (2.20), the foregoing expression becomes

$$\frac{1}{2(2\pi)^{3/2}} \int_0^{2\pi} \left(\int_{\mathbb{R}} F_\rho \left(H(R_\theta f)' \right) e^{i\rho\theta\cdot\mathbf{x}} d\rho \right) d\phi$$

Now, the inner most integral is just an inverse Fourier transform, and so by (2.19), we achieve

$$f(\mathbf{x}) = \frac{1}{2(2\pi)^{3/2}} \int_0^{2\pi} \left(H(R_\theta f)' \right) (\theta \cdot \mathbf{x}) d\phi.$$

□

Chapter 3

The Discrete Radon Transform

3.1 Discrete Analog of Classical Chebyshev Inversion Formula

Recall the classical Chebyshev Radon inversion formula given in the previous section:

$$f = \sum_{m=1}^{\infty} \sum_{\theta \in \Omega_m} \langle f, U_m\left(x \cdot e_\theta\right)\rangle U_m(\bullet \cdot \mathbf{e}_\theta), \quad f \in L_2,$$

which has the approximate form

$$f \approx \sum_{m=1}^{M} \sum_{\theta \in \Omega_m} \langle f, U_m\left(x \cdot e_\theta\right)\rangle U_m(\bullet \cdot \mathbf{e}_\theta), \quad f \in L_2.$$

Narrowing our focus to functions $f \in L_2\left(\mathcal{D}\right)$, we will achieve a discretization of the aforementioned approximate inversion formula from discretizing the integral appearing in the inner product. Using the notation $x = (x_1, x_2)$, consider that

$$\begin{aligned}\langle f, U_m\left(\bullet \cdot e_\theta\right)\rangle &= \frac{1}{\pi} \int\int_{\mathcal{D}} f(x) U_m\left(x \cdot e_\theta\right) dx_1 dx_2 \\ &= \frac{1}{\pi} \int_{-1}^{1} R_f\left(\theta, s\right) U_m(s) ds.\end{aligned}$$

Similar to the developments given in Chapter 2, we set $s = \cos\alpha$ to obtain the equivalent expression

$$\frac{1}{\pi} \int_{0}^{\pi} R_f\left(\theta, \cos\alpha\right) \sin m\alpha d\alpha.$$

Hence, a discretization is obtained by sampling values of α between 0 and π. Letting $\alpha_k = \pi\left(2K\right)^{-1} + \left(k-1\right)\pi K^{-1}$, where K is the number of nodes used in the sampling, we receive the following approximate discrete identity

$$\langle f, U_m\left(\bullet \cdot e_\theta\right)\rangle \approx \frac{1}{K} \sum_{k=1}^{K} R_f\left(\theta, \cos\alpha_k\right) \sin m\alpha_k.$$

Thus, this naturally leads to

$$f(x) \approx \frac{1}{K} \sum_{m=1}^{M} \sum_{\theta \in \Omega_m} \left(\sum_{k=1}^{K} R_f \left(\theta, \cos \alpha_k \right) \sin m\alpha_k \right) U_m \left(x \cdot e_\theta \right),$$
$$\alpha_k = \frac{\pi}{2K} + (k-1) \frac{\pi}{K}.$$

To finish our computational framework, further sampling is required for values of θ between 0 and π. For $M \geq 1$, we let $\theta_j = j\pi M^{-1}$ for $j = 1, 2, \ldots, M$. Thus, it follows that

$$f(x) \approx \frac{1}{K} \sum_{m=1}^{M} \sum_{j=1}^{m} \left(\sum_{k=1}^{K} R_f \left(\theta_j, \cos \alpha_k \right) \sin m\alpha_k \right) U_m \left(x \cdot e_{\theta_j} \right),$$
$$\theta_j = j\frac{\pi}{M}, \quad \alpha_k = \frac{\pi}{2K} + (k-1) \frac{\pi}{K}. \tag{3.1}$$

The approximate result (3.1) is only suitable for functions $f \in L_2(\mathcal{D})$, where $\mathcal{D}$ is the unit disk. Necessarily, any algorithm making use of this approximate identity would need to allow for functions of more general sizes, say $f : N \times N \to \mathbb{R}$. With this in mind, we wish to scale (3.1) so that it can be applied to functions in the space $L_2(\mathcal{D}_r)$, where $\mathcal{D}_r$ denotes the disk of radius $r = \sqrt{2}N/2$.

We begin with an alternate version of the classical Chebyshev inversion formula and let $f \in L_2(\mathcal{D})$:

$$f = \sum_{m=1}^{\infty} \sum_{j=1}^{\infty} \frac{1}{\pi} \int \int_{\mathcal{D}} f(y) U_m \left(y \cdot e_{\theta_j} \right) dy U_m \left(\bullet \cdot e_{\theta_j} \right).$$

We wish to find $g \in \mathcal{D}_r$ such that $g(ry) = f(y)$, where $y = zr^{-1}$ and $|z| \leq r$. Consider that, for $|x'| \leq 1$,

$$g(rx') = \sum_{m=1}^{\infty} \sum_{j=1}^{\infty} \frac{1}{\pi} \int \int_{\mathcal{D}} g(ry) U_m \left(y \cdot e_{\theta_j} \right) dy U_m \left(x' \cdot e_{\theta_j} \right).$$

Setting $w = ry$, a quick change of variables leads us to a rescaled version of (3.1),

$$g = \sum_{m=1}^{\infty} \sum_{j=1}^{\infty} \frac{1}{\pi r^2} \int \int_{\mathcal{D}_r} g(w) U_m \left(\frac{w}{r} \cdot e_{\theta_j} \right) dw U_m \left(\frac{1}{r} \bullet \cdot e_{\theta_j} \right).$$

By approximating and using the same sampling scheme as in (3.1), we achieve our desired rescaled approximate identity. Namely, for $f \in L_2\left(\mathcal{D}_r\right)$, it follows that

$$f(x) \approx \frac{1}{rK} \sum_{m=1}^{M} \sum_{j=1}^{m} \left(\sum_{k=1}^{K} R_f\left(\theta_j, r\cos\alpha_k\right) \sin m\alpha_k \right) U_m\left(\frac{1}{r} x \cdot e_{\theta_j}\right),$$
$$\theta_j = j\frac{\pi}{M}, \quad \alpha_k = \frac{\pi}{2K} + (k-1)\frac{\pi}{K}. \tag{3.2}$$

This naturally leads to an easy, albeit computationally slow, algorithm with which one could use to compute the reconstruction of an image f from its projection data $R_f\left(\theta_j, r\cos\alpha_k\right)$, where $j = 1, 2, \ldots, M$ and $k = 1, 2, \ldots, K$. Of course, this is evident so long as one has a method with which to compute said projection data. In the next section, we present an informal discussion involving two algorithms which serve as ways of computational data acquisition of the Radon transform, $R_f\left(\theta, s\right)$.

3.2 Data Acquisition Algorithms for the Discrete Radon Transform

Approximating the Radon transform $R_f\left(\theta, s\right)$ of a function f is a matter of approximating the line integral of f along the line parameterized by an angle θ (from the positive x-axis) and a distance s from the origin.

The general idea for approximation then is to begin by considering $f \in L_2\left(\mathcal{D}_r\right)$, where $r = \sqrt{2}N/2$ for $N \geq 1$, as an $N \times N$ image. We consider $f(i,j)$ to be the pixel intensity at the point (i,j), where $-\lceil N/2 \rceil \leq i, j \leq \lceil N/2 \rceil - 1$. Then, the Radon transform $R_f\left(\theta, s\right)$ can be thought of as the sum of the pixel intensities intersected by the line parameterized by (θ, s). The pixel intensities may be weighted in some manner relating to the length of the intersection. As we will discuss later in this section, the algorithms given in A.1 and A.2 differ in the way we emphasize and calculate such weights. For now, to make this idea more concrete, we will introduce some definitions to help in developing a more formal language with which we can discuss these topics.

Definition 3.1. *An $N \times N$ image region is a square whose center is at the origin of the Cartesian plane, and which is subdivided into N^2 equal pixels by an N^2-element grid.*

Definition 3.2. *An image f is a function of two variables whose value in the interior region of any pixel of an N^2-element grid is uniform.*

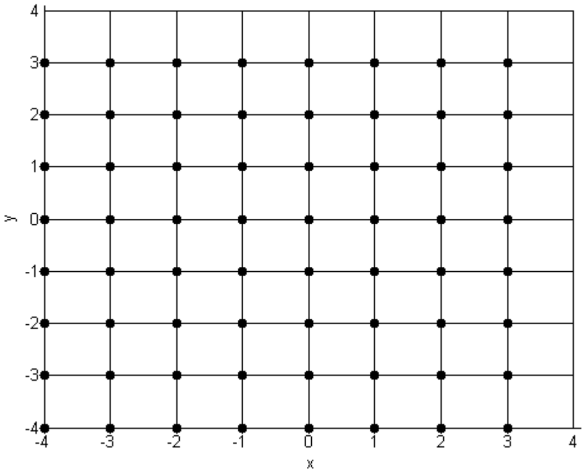


Figure 3.1: A 8×8 image region.

As shown in Figure 3.1, the center pixel $(0, 0)$ of the image region has its lower-left corner located at the origin of the Cartesian plane. For our purposes, we consider gray-scale images f where $f(i, j)$ represents the pixel intensity at the integral point (i, j). The gray-scale Shepp-Logan phantom head, such as in Figure 3.2, can be represented on such an image region. To see what this would look like, refer to Figure 3.3, which shows a 32×32 image being represented on a 32×32 image region. The height of each bar represents the intensity of the pixel at the lower-left corner of the square it is defined on.

Returning our attention to the problem of acquiring the projection data $R_f(\theta, s)$, we will denote by $P_{(\theta,s)}$ the set of pixels intersected by the line with distance s from the origin and angle θ from the positive x-axis. Let $S = \{-\lceil N/2 \rceil, \dots, \lceil N/2 \rceil - 1\}$,

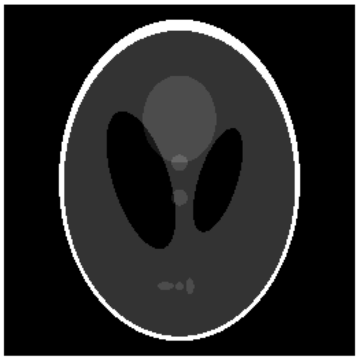


Figure 3.2: A 256 × 256 Shepp-Logan phantom head.

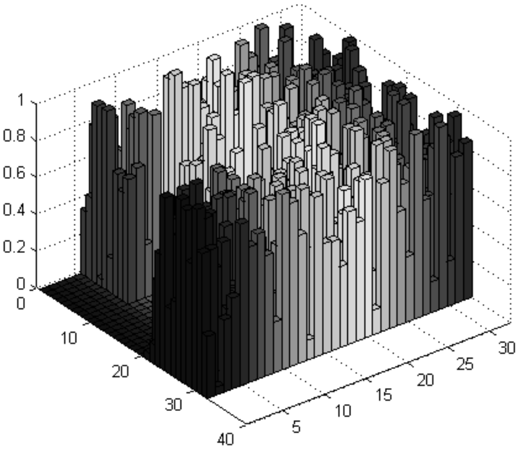


Figure 3.3: A representation of a 32 × 32 image on an image region.

then we define $P_{(\theta,s)}$ as below.

$$P_{(\theta,s)} = \{(x,y) \in \mathbb{Z}^2 \mid x,y \in S, \text{ pixel at (x,y) is intersected}\} \tag{3.3}$$

Figure 3.4 shows an example of a set of pixels being intersected by one possible line. Using this definition, we can define the line integral which will be used in our discrete Radon transform algorithms in the following manner.

Definition 3.3. *Let $P(\theta, s)$ be known within an image region of size $N \times N$. Then, the discrete line integral of an image f along the line parameterized by a distance*

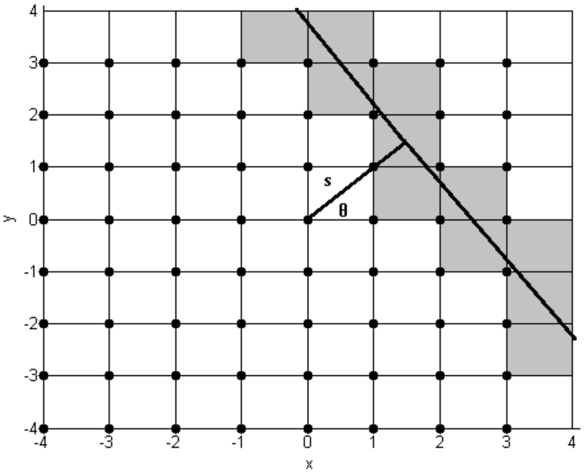


Figure 3.4: The pixels intersected by the line parameterized by (θ, s).

s from the origin and an angle θ from the positive x-axis is approximated by the weighted sum $\hat{L}_f(\theta, s)$ over the pixels in the interior of the image region. Specifically, $\hat{L}_f(\theta, s)$ is defined as

$$\hat{L}_f(\theta, s) := \sum_{i \in S} \sum_{j \in S} w_{(\theta,s)}(i,j)\, f(i,j), \tag{3.4}$$

where

$$w_{(\theta,s)}(i,j) := \begin{cases} l, & \textit{if } (i,j) \in P_{(\theta,s)} \\ 0, & \textit{if } (i,j) \notin P_{(\theta,s)}, \end{cases} \tag{3.5}$$

and l is defined as the length of the intersection.

In this manner, we see that the problem of reconstructing an image f from its projection data is a problem of solving a system of linear equations. Assuming we acquire the projections of f along a set of lines parameterized by a set of angles $\Theta = \{\theta_m\}_{m=1}^{M}$ and distances from the origin $\Lambda = \{s_k\}_{k=1}^{K}$, we can express such projection data as a system of linear equations,

$$\begin{cases} \hat{L}_f\left(\theta_1, s_1\right) & = \sum\limits_{i \in S} \sum\limits_{j \in S} w_{\left(\theta_1, s_1\right)}\left(i, j\right) f(i,j), \\ \hat{L}_f\left(\theta_1, s_2\right) & = \sum\limits_{i \in S} \sum\limits_{j \in S} w_{\left(\theta_1, s_2\right)}\left(i, j\right) f(i,j), \\ \ldots & \\ \hat{L}_f\left(\theta_M, s_{K-1}\right) & = \sum\limits_{i \in S} \sum\limits_{j \in S} w_{\left(\theta_M, s_{K-1}\right)}\left(i, j\right) f(i,j), \\ \hat{L}_f\left(\theta_M, s_K\right) & = \sum\limits_{i \in S} \sum\limits_{j \in S} w_{\left(\theta_M, s_K\right)}\left(i, j\right) f(i,j). \end{cases}$$

Thus, we can equivalently express the acquisition of projection data in terms of matrices. Letting **L** denote the left-hand side of the above equations, **W** denote the $MK \times N^2$ matrix containing the weights, and **F** denote the $N^2 \times 1$ image vector containing the pixel intensity information for f, the above system of equations can be rewritten as the matrix equation

$$\mathbf{L} = \mathbf{WF}. \tag{3.6}$$

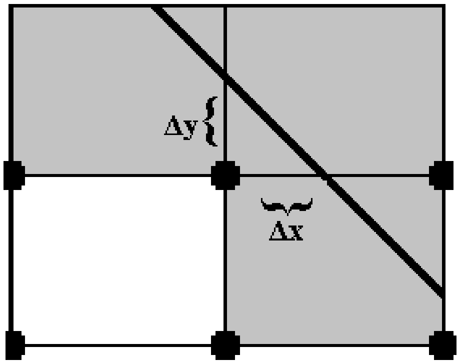


Figure 3.5: The length, $l = \sqrt{\Delta x^2 + \Delta y^2}$, of an intersection.

The basic structure of an algorithm that can be used in order to compute the set of projection data **L** is divided into several steps, detailed below. Note that computation time can be cut by first determining the left, right, top, and bottom most pixels intersected by a line within the image region. We call these pixels the left, right, top, and bottom exits. Computing these locations as a first step makes it so that we can restrict our search in terms of determining the weights of the intersections by eliminating pixels which are most definitely not intersected. Though algorithms A.1 and A.2 differ in a few respects, they both make use of the following general outline.

- Step 1: For $\theta \in \Theta$ and $s \in \Lambda$, compute $P_{(\theta,s)}$.
- Step 2: Determine left, right, top, and bottom exits for the line parameterized by (θ, s).
- Step 3: Scan through pixels between left, right, top, and bottom exits to determine the lengths of intersection.
- Step 4: Multiply the lengths by their respective pixel values (determined by f) and add this to a running sum representing $\hat{L}_f\,(\theta, s)$.
- Step 5: Repeat the process for each $\theta \in \Theta$ and $s \in \Lambda$.

The algorithms we present in A.1 and A.2 serve as ways of computing the discrete line integral of an image f along a line parameterized by (θ, s). Both of these algorithms make use of the fact that for $\theta \in (0, \pi/2)$, the line will be decreasing as we move from the left exit to the right exit and therefore, the top exit will come before the bottom exit. Alternatively, for $\theta \in (\pi/2, \pi)$, the bottom exit will occur before the top exit as we move from the left exit to the right exit. In this manner, the way we search through the potentially intersected pixels is characterized by the range in which θ occurs.

If $\theta = 0$ or $\theta = \pi$, then the line is vertical and we merely use the distance s from the origin to find the “column" in which the line occurs. Every pixel in this column is intersected by the line and the length of the intersection is 1. Similarly, if $\theta = \pi/2$, then the line is horizontal and so we use the distance s to find the “row" in which

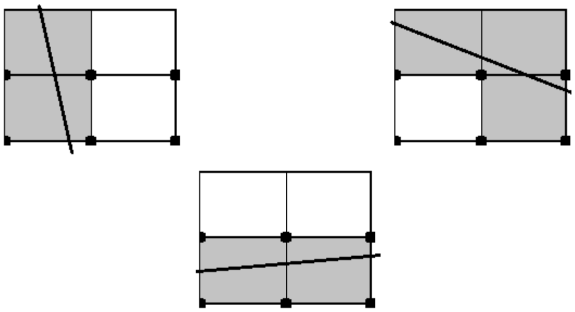


Figure 3.6: Some of the various ways a line can intersect a pixel.

the line occurs. Every pixel in this row is intersected by the line and the length of the intersection is also 1. We should note that as θ gets close to 0 or π, the slope of the line parameterized by θ goes towards infinity. This causes some blurring when reconstructing f using the projection data, as we see in Figure 3.8.

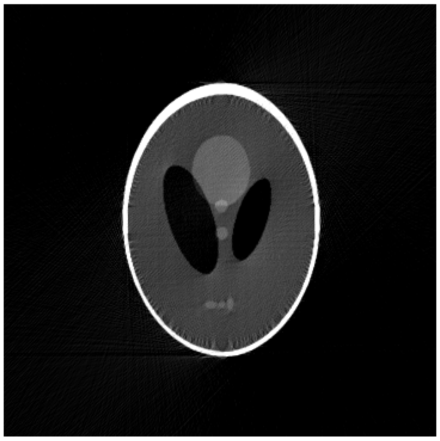


Figure 3.7: Reconstruction of Shepp-Logan Phantom head using data from Algorithm A.1.

Beyond this, the algorithms given in A.1 and A.2 essentially differ in the way we characterized the weights assigned to the lengths of the intersection between a line parameterized by (θ, s) and $P_{(\theta,s)}$. Algorithm A.1 explicitly calculates the lengths of the intersections for every point in $P_{(\theta,s)}$, while Algorithm A.2 makes use of linear interpolation. Algorithm A.2 first finds all of the x or y coordinates of the intersection points, depending on what range θ is in, and then subsequently uses these values to

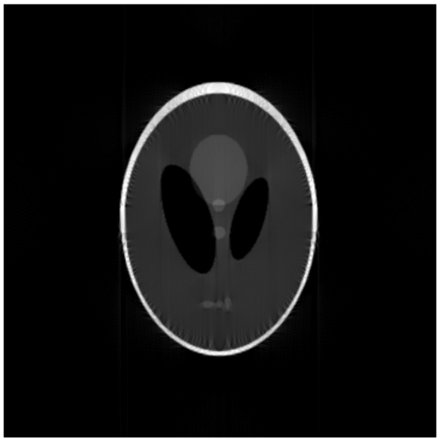


Figure 3.8: Reconstruction of Shepp-Logan Phantom head using data from Algorithm A.2.

determine the y or x coordinates, respectively. From here, the algorithm uses these coordinates to “split" the pixel, and subsequently places the pixel values (weighted by the manner in which they were split) into bins of size 1.

Due to the fact that Algorithm A.1 explicitly calculates the lengths of the intersections, it naturally has to take into account the various ways pixels can be intersected (see Figure 3.6). Hence, further conditionals are placed within the pixel search, which causes computation time to increase. However, it should be noted that while Algorithm A.2 is faster than Algorithm A.1, it is not as accurate due to the way we calculate the weights of the intersections. It was found that this difference in accuracy was mostly evident when $\Lambda = \{s_k\}_{k=1}^{K}$ was chosen in a manner such that the values s_k were not uniformly distributed for $k = 1, 2, \ldots K$.

On a final note, we would like to remind the reader that Algorithms A.1 and A.2 merely only serve the function of calculating the *discrete line integral* of f for a specific line. Algorithm A.3 ties this together by utilizing these algorithms in order to calculate the projection data of a given image f for a given angle set Θ and distance set Λ. Assuming $|\Theta| = M$ and $|\Lambda| = K$, the output of Algorithm A.3 is a $K \times M$ matrix, which can then be converted into a $MK \times 1$ vector serving as a representation of the projection matrix **L** described in (3.6). From this point, an inverse radon transform algorithm can be applied to reconstruct images, such as in Figures 3.7 and 3.8 where MATLAB's built-in inverse radon transform function was used.

Chapter 4

Sub-Exponential Localization Properties of Needlets

In the next chapter, we will introduce a new inversion formula for the Radon transform based on kernels and frames which we refer to as *needlets.* The term *needlet* comes from the fact that kernels of the form (5.1) exhibit subexponential localization properties (i.e. they are *extremely* well-localized). In this chapter, we develop and illustrate these properties in rigorous detail. The discussion begins by introducing the class of functions known as the Schwarz class.

Definition 4.1. *Let $f \in \mathbb{R}^n$, $n \geq 1$. Then, f is said to be rapidly decreasing if for every integer $N \geq 0$, there exists a constant C_N such that*

$$|f(x)| \leq \frac{C_N}{(1+|x|)^N}$$

for all $x \in \mathbb{R}^n$. We denote the space of rapidly decreasing functions by $\mathcal{D}(\mathbb{R}^n)$.

Definition 4.2. *The Schwartz class $\mathcal{S}$ is defined to be*

$$\mathcal{S}(\mathbb{R}^n) := \{f \in C^\infty(\mathbb{R}^n) \mid f, f', f'', \ldots \in \mathcal{D}(\mathbb{R}^n)\}. \tag{4.1}$$

Recalling the form of the kernels of interest,

$$L_n(x,y) = \sum_{j=1}^{\infty} \hat{a}\left(\frac{j}{n}\right) U_j(x) U_j(y),$$

it will take a series of theorems, corollaries, and lemmas to prove the following result, which is the primary claim of this chapter.

Theorem 4.3. *Let $0 \leq \theta, \phi \leq \pi$, and suppose $\hat{a} \in \mathcal{S}(\mathbb{R})$ is an even function. Then, for any $\sigma > 0$, there exists a constant $c_\sigma > 0$ such that*

$$|L_n(\cos\theta, \cos\phi)| \leq \frac{c_\sigma n}{(\sin\theta + n^{-1})(\sin\phi + n^{-1})(1 + n|\theta - \phi|)^\sigma}.$$

It follows as an immediate corollary from our primary claim that for $x, y \in \mathbb{R}$,

$$|L_n(x, y)| \leq \frac{c_\sigma n}{\left(\sqrt{1 - x^2} + n^{-1}\right)\left(\sqrt{1 - y^2} + n^{-1}\right)(1 + n\rho(x, y))^\sigma}, \tag{4.2}$$

where $\rho(x, y) = |\arccos x - \arccos y|$.

Theorem 4.4. *Let P be an arbitrary constant. For appropriate functions g,*

$$\sum_{j \in \mathbb{Z}} g(Pj) = \frac{1}{P} \sum_{j \in \mathbb{Z}} \hat{g}(j).$$

The equation given by Theorem 4.4 is known as the Poisson summation formula and is necessary to prove the following lemma.

Lemma 4.5. *Consider the trigonometric polynomial given by*

$$F_n(\theta) = \sum_{j \in \mathbb{Z}} \hat{a}\left(\frac{j}{n}\right) e^{ij\theta}, \tag{4.3}$$

where $\hat{a} \in C^\infty(\mathbb{R})$ and $\operatorname{supp} \hat{a} \in [-2, 2]$. *Then, for any $\sigma > 0$, there exists $c_\sigma > 0$ such that*

$$|F_n(\theta)| \leq \frac{c_\sigma n}{(1 + n|\theta|)^\sigma}, \quad \theta \in [-\pi, \pi]. \tag{4.4}$$

Proof. Let f be defined in terms of its Fourier transform

$$\hat{f}(\xi) := \hat{a}\left(\frac{\xi}{n}\right) e^{i\xi t}.$$

Then, it follows that

$$\begin{aligned} f(y) &= \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\xi) e^{i\xi y} d\xi \\ &= \frac{1}{2\pi} \int_{\mathbb{R}} \hat{a}\left(\frac{\xi}{n}\right) e^{i\xi t} e^{i\xi y} d\xi \\ &= \frac{1}{2\pi} \int_{\mathbb{R}} \hat{a}\left(\frac{\xi}{n}\right) e^{i\xi(t+y)} d\xi. \end{aligned}$$

Letting $u = \xi/n$, it follows that $d\xi = ndu$ and the former yields

$$f(y) = na\left(n\left(t+y\right)\right). \tag{4.5}$$

Now,

$$\begin{aligned}|F_n(\theta)| &= \left|\frac{1}{2}\sum_{j\in\mathbb{Z}} \hat{a}\left(\frac{j}{n}\right) e^{ij\theta}\right| \\ &= \left|\frac{1}{2}\sum_{j\in\mathbb{Z}} \hat{f}\left(j\right)\right|.\end{aligned}$$

From Theorem 4.4, we obtain

$$|F_n(\theta)| = \pi \sum_{j\in\mathbb{Z}} f(2\pi j).$$

From here, (4.5) yields

$$|F_n(\theta)| = \pi n \sum_{j\in\mathbb{Z}} a\left(n\left(\theta + 2\pi j\right)\right).$$

Now, $\hat{a} \in C^{\infty}\left(\mathbb{R}\right)$ is of compact support and so it follows that $\hat{a} \in \mathcal{S}\left(\mathbb{R}\right)$. To see this, simply note that any derivative of $\hat{a}$ is supported on supp $\left(\hat{a}\right)$ and therefore also has compact support. Hence, any derivative of $\hat{a}$ is thereby bounded by the Extreme Value Theorem. Thus, it follows that for any $\sigma > 0$, there exists $c_\sigma > 0$ such that

$$\begin{aligned}\sum_{j\in\mathbb{Z}} \hat{a}\left(n\left(\theta+2\pi j\right)\right) &\leq \sum_{j\in\mathbb{Z}} \frac{c_\sigma}{\left(1+n\left|\theta+2\pi j\right|\right)^{\sigma}} \\ &= \frac{c_\sigma}{\left(1+n\theta\right)^{\sigma}} + \sum_{j\neq 0} \frac{c_\sigma}{\left(1+n\left|\theta+2\pi j\right|\right)^{\sigma}}.\end{aligned}$$

Recall that $-\pi \leq \theta \leq \pi$. Then, the foregoing becomes

$$\sum_{j\in\mathbb{Z}} \hat{a}\left(n\left(\theta+2\pi j\right)\right) \leq \frac{c_\sigma}{\left(1+n\left|\theta\right|)\right)^{\sigma}} + \sum_{j\neq 0} \frac{c_\sigma}{\left(1+n\left|\pi j\right|\right)^{\sigma}}.$$

From here, simple quantitative arguments yield our desired result:

$$\sum_{j\in\mathbb{Z}} \hat{a}\left(n\left(\theta+2\pi j\right)\right) \leq \frac{c_\sigma}{\left(1+n\left|\theta\right|\right)^{\sigma}} + \sum_{j=1}^{\infty} \frac{c_\sigma}{\left(\left|nj\right|\right)^{\sigma}}.$$

□

Lemma 4.6. *Suppose* $0 \leq \theta, \phi \leq \pi$. *Then, for any* $\sigma > 0$ *there exists a constant* $c_\sigma > 0$ *such that*

$$|F_n(\theta)| \leq \frac{c_\sigma n}{(1 + n|\theta - \phi|)^\sigma}. \tag{4.6}$$

Proof. Let $\sigma > 0$ be given. We consider the following two cases.

- Case 1: Suppose $\theta + \phi \leq \pi$. Then, $\theta + \phi \geq |\theta - \phi|$ and therefore, from (4.4) it follows that

$$\begin{aligned} |F_n(\theta + \phi)| &\leq \frac{c_\sigma n}{(1 + n|\theta + \phi|)^\sigma} \\ &\leq \frac{c_\sigma n}{(1 + n|\theta - \phi|)^\sigma}. \diamond \end{aligned}$$

- Case 2: Suppose $\theta + \phi > \pi$. Then, $0 < 2\pi - \theta - \phi \leq \pi$. Letting $\alpha = \pi - \theta$ and $\beta = \phi - \theta$, it follows that $2\pi - \theta - \phi = \alpha + \beta$. Furthermore, $0 \leq \alpha, \beta \leq \pi$ and $\alpha + \beta \geq |\alpha - \beta| = |\theta - \phi|$. Coupling this with (4.4) and the fact that F_n is an even function yields

$$\begin{aligned} |F_n(\theta + \phi)| &= |F_n(2\pi - \theta - \phi)| \\ &\leq \frac{c_\sigma n}{(1 + n|2\pi - \theta - \phi|)^\sigma} \\ &\leq \frac{c_\sigma n}{(1 + n|\theta - \phi|)^\sigma}. \end{aligned}$$

□

From above, we obtain the following corollary.

Corollary 4.7. *Let* $0 < \theta, \phi < \pi$. *Then, for any* $\sigma > 0$ *there exists a constant* $c_\sigma > 0$ *such that*

$$|L_n(\cos\theta, \cos\phi)| \leq \frac{c_\sigma n}{\sin\theta \sin\phi\, (1 + n|\theta - \phi|)^\sigma} \tag{4.7}$$

Proof. From (5.1), we observe that for $0 < \theta, \phi < \pi$,

$$L_n(\cos\theta, \cos\phi) = \sum_{j=1}^{\infty} \hat{a}\left(\frac{j}{n}\right) \frac{\sin(j\theta)\sin(j\phi)}{\sin\theta \sin\phi}.$$

By adding/subtracting the term $\hat{a}(0)/2$ to the aforementioned expression and subsequently applying a product-to-sum trigonometric identity, we obtain that

$$L_n(\cos\theta, \cos\phi) = \frac{1}{2\sin\theta\sin\phi} \sum_{j=1}^{\infty} \hat{a}\left(\frac{j}{n}\right) \left[\cos\left(j(\theta-\phi)\right) - \cos\left(j(\theta+\phi)\right)\right].$$

Then, by (4.3), the former becomes

$$L_n(\cos\theta, \cos\phi) = \frac{1}{2\sin\theta\sin\phi} \left(F_n\left(\theta-\phi\right) - F_n\left(\theta+\phi\right)\right), \tag{4.8}$$

and therefore, from (4.6), it follows that

$$\left|L_n\left(\cos\theta, \cos\phi\right)\right| \leq \frac{c_\sigma n}{\sin\theta\sin\phi\left(1+n\left|\theta-\phi\right|\right)^\sigma}.$$

□

Lemma 4.8. *Let $0 \leq \theta, \phi \leq \pi$. Then, for any $\sigma > 0$ there exists a constant $c_\sigma > 0$ such that*

$$\left|L_n\left(\cos\theta, \cos\phi\right)\right| \leq \frac{c_\sigma n^3}{\left(1+n\left|\theta-\phi\right|\right)^\sigma}. \tag{4.9}$$

Proof. An important formality we must address is that throughout the proof of this lemma, the constant c_σ may change and, hence, may represent a different constant from instance to instance. However, c_σ will always be dependent *only* on σ. The decision to leave its notation unchanged is a matter of choice based on visual organization and aesthetic, and leaves the validity of the proof intact.

Let G_n be the function defined by

$$G_n(x) := \sum_{j=1}^{\infty} \hat{a}\left(\frac{j}{n}\right) jU_j(x), \tag{4.10}$$

where U_j is a jth Chebyshev Polynomial of the Second Kind as defined in (2.3). Now, from (4.3), it follows that

$$F_n(x) = \frac{\hat{a}(0)}{2} + \sum_{j=1}^{\infty} \hat{a}\left(\frac{j}{n}\right) T_j(x),$$

where T_j is jth Chebyshev Polynomial of the First Kind as defined in (2.2). Then,

$$F_n(\cos\theta) = \frac{\hat{a}(0)}{2} + \sum_{j=1}^{\infty} \hat{a}\left(\frac{j}{n}\right) \cos j\theta.$$

Taking the derivative with respect to θ yields

$$\frac{d}{d\theta}\left[F_n(\cos\theta)\right] = -\sum_{j=1}^{\infty} \hat{a}\left(\frac{j}{n}\right) j \sin j\theta,$$

and so it follows that

$$F_n'(x) = \sum_{j=0}^{\infty} \hat{a}\left(\frac{j}{n}\right) jU_j(x) = G_n(x). \tag{4.11}$$

We now recall the Markov inequality which states that for any $P_m \in \Pi_m$, $m \geq 0$ and $a, b \in \mathbb{R}$,

$$\|P_m'\|_{L_\infty[a,b]} \leq \frac{2n^2}{b-a}\|P_m\|_{L_\infty[a,b]}.$$

Since $F_n \in \Pi_{2n}$, it follows from (4.11) and the above Markov inequality that for $t \in [-1, 1]$,

$$\begin{aligned}\|G_n\|_{L_\infty[-1,t]} &= \|F_n'\|_{L_\infty[-1,t]} \\ &\leq \frac{2n^2}{t+1}\|F_n\|_{L_\infty[-1,t]}.\end{aligned}$$

From (4.6), the foregoing becomes

$$\|G_n\|_{L_\infty[-1,t]} \leq \frac{2n^2}{t+1}\left\|\frac{c_\sigma n}{\left(1+n\sqrt{1-\bullet}\right)^\sigma}\right\|_{L_\infty[-1,t]}.$$

Note that if $t \in [0, 1]$, then the function

$$h(x) := \frac{c_\sigma n}{\left(1+n\sqrt{1-x}\right)^\sigma}$$

attains its maximum at t on the interval $[0,t]$. This follows from the fact that h is a monotonically increasing function on $[-1,1]$. This readily leads us to

$$\begin{aligned}\|G_n\|_{L_\infty[-1,t]} &\leq \frac{2n^2}{t+1}\left\|\frac{c_\sigma n}{\left(1+n\sqrt{1-\bullet}\right)^\sigma}\right\|_{L_\infty[-1,t]}\\ &\leq \frac{2n^2}{t+1}\frac{c_\sigma n}{\left(1+n\sqrt{1-t}\right)^\sigma}\\ &= \frac{c_\sigma n^3}{(t+1)\left(1+n\sqrt{1-t}\right)^\sigma}.\end{aligned}$$

By similar reasoning, for all $t \in [-1,0]$, the aforementioned function h is bounded above on the interval $[-1,t]$ by the case when $t=0$ and so it follows that

$$\begin{aligned}\|G_n\|_{L_\infty[-1,t]} &\leq \frac{c_\sigma n^3}{(1+n)^\sigma}\\ &\leq \frac{c_\sigma n^3}{\left(1+n\sqrt{1-t}\right)^\sigma},\end{aligned}$$

where the last inequality results from scaling of the constant c_σ. Hence, it follows that for all $t \in [-1,1]$,

$$\|G_n\|_{L_\infty[-1,t]} \leq \frac{c_\sigma n^3}{(t+1)\left(1+n\sqrt{1-t}\right)^\sigma}. \tag{4.12}$$

We now refer to a product formula for Gegenbauer polynomials, of which Chebyshev polynomials are a specific type, as presented in [5]. Specifically, it follows from this product formula that for some constant c,

$$\frac{U_j(\cos\theta)\,U_j(\cos\phi)}{U_j(1)} = c\int_{-1}^{1} U_j(\cos\theta\cos\phi + u\sin\theta\sin\phi)\,du,$$

and so therefore,

$$\begin{aligned}L_n(\cos\theta,\cos\phi) &= \sum_{j=1}^{\infty}\hat{a}\left(\frac{j}{n}\right)U_j(\cos\theta)\,U_j(\cos\phi)\\ &= c\int_{-1}^{1}\sum_{j=1}^{\infty}\hat{a}\left(\frac{j}{n}\right)U_j(1)\,U_j(\cos\theta\cos\pi + u\sin\theta\sin\phi)\,du.\end{aligned}$$

Now, $U_j(1) = \lim_{\theta\to 0}(\sin j\theta/\sin\theta) = j$ and so the former becomes

$$L_n(\cos\theta,\cos\phi) = c\int_{-1}^{1}\sum_{j=1}^{\infty}\hat{a}\left(\frac{j}{n}\right)jU_j(\cos\theta\cos\pi + u\sin\theta\sin\phi)\,du.$$

Letting $t(u,\theta,\phi) = \cos\theta\cos\phi + u\sin\theta\sin\phi$ and by using (4.12), we obtain that

$$|L_n(\cos\theta,\cos\phi)| \leq \int_{-1}^{1} \frac{c_\sigma n^3}{\left(1+n\sqrt{1-t(u,\theta,\phi)}\right)^\sigma} du. \tag{4.13}$$

Note that for $0 \leq \theta, \phi \leq \pi$, it follows that $\sin\theta\sin\phi \geq 0$ and so

$$\begin{aligned} 1 - t(u,\theta,\phi) &= 1 - \cos\theta\cos\phi - u\sin\theta\sin\phi \\ &\geq 1 - \cos\theta\cos\phi - \sin\theta\sin\phi \\ &= 1 - \cos(\theta-\phi) \\ &= 2\sin^2(\theta-\phi). \end{aligned}$$

Furthermore, $2\sin^2(\theta-\phi)$ is equivalent to $(\theta-\phi)^2$ (up to some scalar value) when $0 \leq \theta, \phi \leq \pi$. Combining this fact with the above inequality, (4.13) becomes

$$|L_n(\cos\theta,\cos\phi)| \leq \frac{c_\sigma n^3}{(1+n|\theta-\phi|)^\sigma}.$$

□

Lemma 4.9. *Let $\theta, \phi \geq 0$. Then*

$$\left(\theta+n^{-1}\right)\left(\phi+n^{-1}\right) \leq 3\left(\theta\phi+\frac{1}{n^2}\right)(1+n|\theta-\phi|), \quad n \geq 1. \tag{4.14}$$

Proof. Without relevant loss of generality, suppose $\phi \geq \theta$. Then, for some $\lambda \geq 1$, $\phi = \lambda\theta$. Note that proving (4.14) is equivalent to proving the inequality

$$\theta\phi + \frac{1}{n^2} + \frac{\theta+\phi}{n} \leq 3\theta\phi + \frac{3}{n^2} + 3\theta\phi n|\theta-\phi| + \frac{3|\theta-\phi|}{n},$$

and so it suffices to prove that

$$\frac{\theta+\phi}{n} \leq 2\theta\phi + \frac{2}{n^2} + 3\theta\phi n|\theta-\phi| + \frac{3|\theta-\phi|}{n}. \tag{4.15}$$

- Case 1: Suppose $\lambda \geq 3$. Then, $(\theta+\phi)n^{-1} \leq (3|\theta-\phi|)n^{-1}$ if and only if $\lambda + 1 \leq 3(\lambda-1)$. Yet this is true if and only if $4 \leq 2\lambda$, which is equivalent to the condition that $\lambda \geq 2$.

  Therefore, $(\theta+\phi)n^{-1} \leq (3|\theta-\phi|)n^{-1}$ which implies (4.15).

- Case 2: Suppose that $1 \leq \lambda < 3$. Then,

  $$\frac{\theta + \phi}{n} \leq 2\left(\theta\phi + \frac{1}{n^2}\right)$$

  if and only if

  $$\frac{(\lambda + 1)\,\theta}{n} \leq 2\left(\lambda\theta^2 + \frac{1}{n^2}\right).$$

  This inequality holds so long as

  $$\frac{4\theta}{n} \leq 2\left(\theta^2 + \frac{1}{n^2}\right),$$

  which is equivalent to the condition that $\theta^2 - 2\theta n^{-1} + n^{-2} \geq 0$. Factoring this expression tells us that $\left(\theta - n^{-1}\right)^2 \geq 0$, which is clearly true. Therefore,

  $$\frac{\theta + \phi}{n} \leq 2\left(\theta\phi + \frac{1}{n^2}\right),$$

  implying the validity of (4.15).

□

We are finally ready to prove the primary localization result of this chapter.

Proof of Theorem 4.3. The proof is divided into three cases.

- Case 1: Suppose $0 \leq \theta \leq 2\pi/3$ and $0 \leq \phi \leq \pi/2$. Now, $\sin\theta \sin\phi \sim \theta\phi$ (up to some scalar value) and so from (4.7), it follows that

  $$|L_n\left(\cos\theta, \cos\phi\right)| \leq \frac{2c_\sigma n}{2\theta\phi\left(1 + n\left|\theta - \phi\right|\right)^\sigma}. \tag{4.16}$$

  Furthermore, from (4.9), we achieve

  $$|L_n\left(\cos\theta, \cos\phi\right)| \leq \frac{2c_\sigma n}{2n^{-2}\left(1 + n\left|\theta - \phi\right|\right)^\sigma}. \tag{4.17}$$

  - Case 1a: Suppose $\theta\phi \leq n^{-2}$. Then, $\theta\phi + n^{-2} \leq 2n^{-2}$ and so it follows from (4.17) that

    $$|L_n\left(\cos\theta, \cos\phi\right)| \leq \frac{2c_\sigma n}{\left(\theta\phi + n^{-2}\right)\left(1 + n\left|\theta - \phi\right|\right)^\sigma}.$$

- Case 1b: Suppose that $\theta\phi > n^{-2}$. then, $\theta\phi + n^{-2} \leq 2\theta\phi$ and so by (4.16) it follows that

$$|L_n(\cos\theta, \cos\phi)| \leq \frac{2c_\sigma n}{(\theta\phi + n^{-2})(1 + n|\theta - \phi|)^\sigma}.$$

Combining Cases 1a and 1b, we obtain the inequality

$$|L_n(\cos\theta, \cos\phi)| \leq \frac{c_\sigma n}{(\theta\phi + n^{-2})(1 + n|\theta - \phi|)^\sigma},$$

and so therefore,

$$\begin{aligned} |L_n(\cos\theta, \cos\phi)| &\leq \frac{c_\sigma n}{(\theta\phi + n^{-2})(1 + n|\theta - \phi|)^\sigma} \\ &\leq \frac{3c_\sigma n}{3(\theta\phi + n^{-2})(1 + n|\theta - \phi|)(1 + n|\theta - \phi|)^{\sigma - 1}}. \end{aligned}$$

Letting $\sigma' = \sigma - 1$ and using the result given by (4.14), we achieve the following result that

$$|L_n(\cos\theta, \cos\phi)| \leq \frac{c_{\sigma'} n}{(\theta + n^{-1})(\phi + n^{-1})(1 + n|\theta - \phi|)^{\sigma'}}.$$

Since $\sin\theta \sim \theta$, $\sin\phi \sim \phi$, and σ is arbitrary, the foregoing yields

$$|L_n(\cos\theta, \cos\phi)| \leq \frac{c_\sigma n}{(\sin\theta + n^{-1})(\sin\phi + n^{-1})(1 + n|\theta - \phi|)^\sigma}. \diamond$$

- Case 2: Suppose that $\pi/3 \leq \theta \leq \pi$ and $\pi/2 \leq \phi \leq \pi$. Then, $0 \leq \pi - \theta \leq 2\pi/3$ and $0 \leq \pi - \phi \leq \pi/2$. Note that $\cos(\pi - \theta) = -\cos\theta$ and $\cos(\pi - \phi) = -\cos\phi$. Since L_n is an even function in two variables it follows that $L_n(\cos(\pi - \theta), \cos(\pi - \phi)) = L_n(\cos\theta, \cos\phi)$. Hence, the result follows from Case 1. $\diamond$

- Case 3: Suppose that either $0 \leq \phi \leq \pi/3$ and $\pi/2 \leq \theta \leq \pi$ or $2\pi/3 \leq \theta \leq \pi$ and $0 \leq \phi \leq \pi/2$. Then, we have $|\theta - \phi| \geq \pi/6$. Hence, by (4.9) and scaling of the constant c_σ,

$$|L_n(\cos\theta, \cos\phi)| \leq \frac{4c_\sigma n^3}{4(1 + n)^\sigma}.$$

Yet, $(\sin\theta + 1/n)(\sin\phi + 1/n) \leq 4$ and so

$$|L_n(\cos\theta, \cos\phi)| \leq \frac{c_\sigma n}{(\sin\theta + n^{-1})(\sin\phi + n^{-1})(1 + n|\theta - \phi|)^\sigma}.$$

From Cases 1, 2, and 3, the complete result follows. □

As mentioned in the beginning of this chapter, the above proof yields (4.2) as an immediate corollary. That is, for any $\sigma > 0$ there exists a constant $c_\sigma > 0$ such that

$$|L_n(x, y)| \leq \frac{c_\sigma n}{\left(\sqrt{1 - x^2} + n^{-1}\right)\left(\sqrt{1 - y^2} + n^{-1}\right)(1 + n\rho(x, y))^\sigma},$$

where $\rho(x, y) = |\arccos x - \arccos y|$.

Chapter 5

Ridgelet Inversion of the Radon Transform

5.1 Orthogonal Expansion using Chebyshev Polynomials

In this chapter, we are interested in kernels of the form

$$L_n(x,y) = \sum_{j=1}^{\infty} \hat{a}\left(\frac{j}{n}\right) U_j(x)U_j(y), \tag{5.1}$$

where $\hat{a} \in \mathcal{S}(\mathbb{R})$, the class of Schwarz functions on the real line (see (4.1)). The results presented in Chapter 4 showed that kernels of this nature exhibit sub-exponential localization properties. These properties have implications relating to the speed at which we can compute their values. Practically speaking, a Radon inversion formula which is based on kernels of the above form would have greater advantages, in terms of computation time, as opposed to methods which employ standard algebraic reconstruction techniques.

The primary identity we will need is the following.

Theorem 5.1. *Let $f \in L_2(\mathcal{D})$. Then,*

$$f(x) = \sum_{m=1}^{\infty} \frac{m}{2\pi^2} \int_0^{2\pi} \int_{\mathcal{D}} f(y) U_m(y \cdot e_\alpha)\, dy U_m(x \cdot e_\alpha)\, d\alpha.$$

Proof. We will use two intermediate identities to prove the claim. The first of which is an inner product result which we developed in Chapter 2. Using the notation given in (2.1),

$$\frac{1}{\pi} \int_{\mathcal{D}} U_m(x \cdot e_\alpha) U_m(x \cdot e_\beta)\, dx = \frac{\sin m(\alpha - \beta)}{m \sin(\alpha - \beta)} \tag{5.2}$$

The second identity we will use is the following.

$$\frac{1}{2\pi}\int_0^{2\pi} U_m\left(x\cdot e_\alpha\right)\frac{\sin m\left(\beta-\alpha\right)}{\sin\left(\beta-\alpha\right)}d\alpha = U_m\left(x\cdot e_\beta\right). \tag{5.3}$$

Denote the functional Q_m by

$$Q_m f(x) := \frac{m}{2\pi^2}\int_0^{2\pi}\int_{\mathcal{D}} f(y)U_m\left(y\cdot e_\alpha\right)dyU_m\left(x\cdot e_\alpha\right)d\alpha. \tag{5.4}$$

Note that it suffices to prove the claim for the case when $f(x) = U_m\left(x\cdot e_\beta\right)$, since the Chebyshev Polynomials of the Second Kind form an orthonormal basis on the space $L_2(\mathcal{D})$. Now, by (5.2) and (5.3), it follows that

$$\begin{aligned} Q_m f(x) &= \frac{m}{2\pi}\int_0^{2\pi}\frac{1}{\pi}\int_{\mathcal{D}} U_m\left(y\cdot e_\beta\right)U_m\left(y\cdot e_\alpha\right)dyU_m\left(x\cdot e_\alpha\right)d\alpha \\ &= \frac{m}{2\pi}\int_0^{2\pi} U_m\left(x\cdot e_\alpha\right)\frac{\sin m\left(\beta-\alpha\right)}{m\sin\left(\beta-\alpha\right)}d\alpha \\ &= U_m\left(x\cdot e_\beta\right). \end{aligned}$$

Therefore, using an orthogonal reconstruction f, the general result follows:

$$f(x) = \sum_{m=1}^{\infty}\frac{m}{2\pi^2}\int_0^{2\pi}\int_{\mathcal{D}} f(y)U_m\left(e_\alpha\cdot y\right)dyU_m\left(e_\alpha\cdot x\right)d\alpha.$$

□

Applying this to our usage of the Radon transform, observe that

$$\int_{\mathcal{D}} f(y)U_m\left(y\cdot e_\alpha\right)dy = \int_{-1}^{1} R_f\left(\alpha,s\right)U_m(s)ds,$$

and so from the previous theorem,

$$f(x) = \frac{1}{2\pi^2}\sum_{m=1}^{\infty} m\int_0^{2\pi}\left(\int_{-1}^{1} R_f\left(\alpha,s\right)U_m(s)ds\right)U_m\left(x\cdot e_\alpha\right)d\alpha. \tag{5.5}$$

We now propose a computational framework for reconstructing $f \in L_2(\mathcal{D})$ using the Chebyshev orthogonal expansion of f. While the approximate identity proposed in this section *does not* exhibit sub-exponential localization properties, it will be from the ideas presented in this section that will lead us to our desired result in the second part of this chapter.

Now, for $M \geq 1$, consider the kernel given by

$$\Phi_M(s,t) := \sum_{m=1}^{2M} \varphi\left(\frac{m}{M}\right) m U_m(s) U_m(t). \tag{5.6}$$

For our purposes, we will consider a specific cutoff function $\varphi_b \in C^\infty(\mathcal{D})$. Let $\widetilde{\phi}_b$ be defined by

$$\widetilde{\phi}_b(t) := \frac{\pi}{2}\frac{(2b+1)!!}{(2b)!!}\int_0^t \sin^{2b+1}(\pi v)\, dv, \quad b \geq 1$$

where b is an integer parameter.

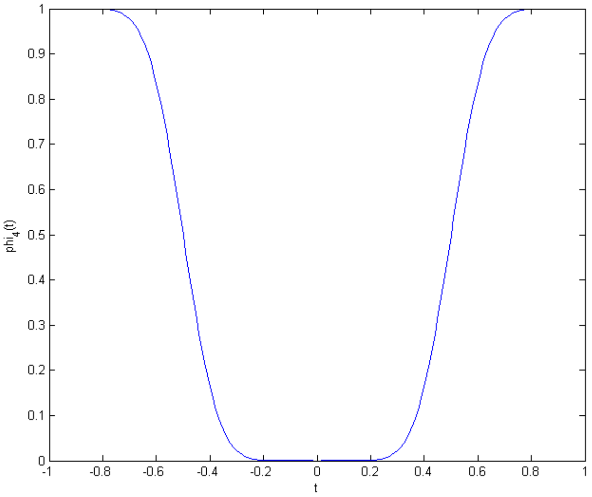


Figure 5.1: Plot of $\phi_4(t)$.

Subsequently denoting by ϕ_b the even extension of $\widetilde{\phi}_b$, for a fixed parameter b we define φ by

$$\varphi(t) := \begin{cases} 1, & 0 \leq t < 1, \\ \phi_b(-t), & 1 \leq t \leq 2. \end{cases} \tag{5.7}$$

Using (5.5), we obtain a reconstruction of the form

$$f(x) = \frac{1}{2\pi^2}\int_0^{2\pi}\int_{-1}^{1}\left[\sum_{m=1}^{\infty} m U_m(s) U_m(x \cdot e_\alpha)\right] R_f(\alpha, s) ds d\alpha, \tag{5.8}$$

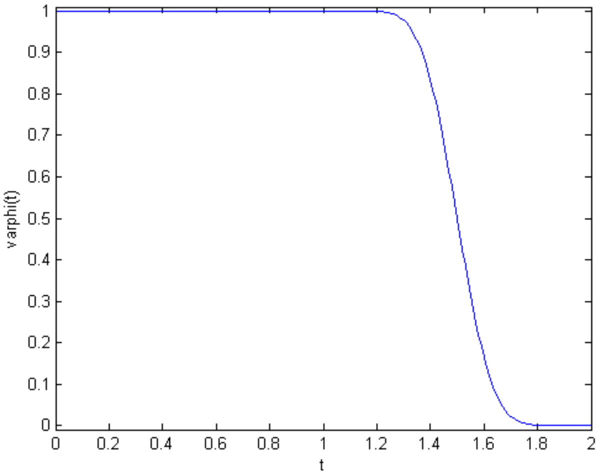


Figure 5.2: Plot of $\varphi(t)$.

and so (5.6) results in the approximate identity

$$f(x) \approx \frac{1}{2\pi^2} \int_0^{2\pi} \int_{-1}^{1} \Phi_M(s, x \cdot e_\alpha) R_f(\alpha, s) ds d\alpha.$$

The discretization of previous expression is as follows. Let $N \geq 1$, then discretizing the outer integral yields

$$f(x) \approx \frac{1}{2\pi^2} \frac{2\pi}{N} \sum_{j=0}^{N-1} \int_{-1}^{1} \Phi_M \left(s, x \cdot e_{\alpha_j}\right) R_f(\alpha_j, s) ds, \quad \alpha_j = \frac{2\pi j}{N}.$$

Letting $s = \cos\theta$, we have that

$$f(x) \approx \frac{1}{\pi N} \sum_{j=0}^{N-1} \int_0^{\pi} \Phi_M \left(\cos\theta, x \cdot e_{\alpha_j}\right) R_f(\alpha_j, \cos\theta) \sin\theta d\theta.$$

The discretization of the remaining integral gives us

$$f(x) \approx \frac{1}{\pi N} \sum_{j=0}^{N-1} \frac{\pi}{K} \sum_{k=0}^{K-1} \Phi_M \left(\cos\theta_k, x \cdot e_{\alpha_j}\right) R_f\left(\alpha_j, \cos\theta_k\right) \sin\theta_k, \quad \theta_k = \frac{\pi k}{K},$$

where $K \geq 1$.

Hence, it follows that

$$f(x) \approx \frac{1}{NK} \sum_{j=0}^{N-1} \sum_{k=0}^{K-1} \Phi_M \left(\cos\theta_k, x \cdot e_{\alpha_j}\right) \sin\theta_k R_f(\alpha_j, \cos\theta_k), \tag{5.9}$$

where $\alpha_j = 2\pi j N^{-1}$ and $\theta_k = \pi k K^{-1}$.

It now remains to develop a framework for computation of the values of Φ_M. Recalling the definition given in (5.6), we have that

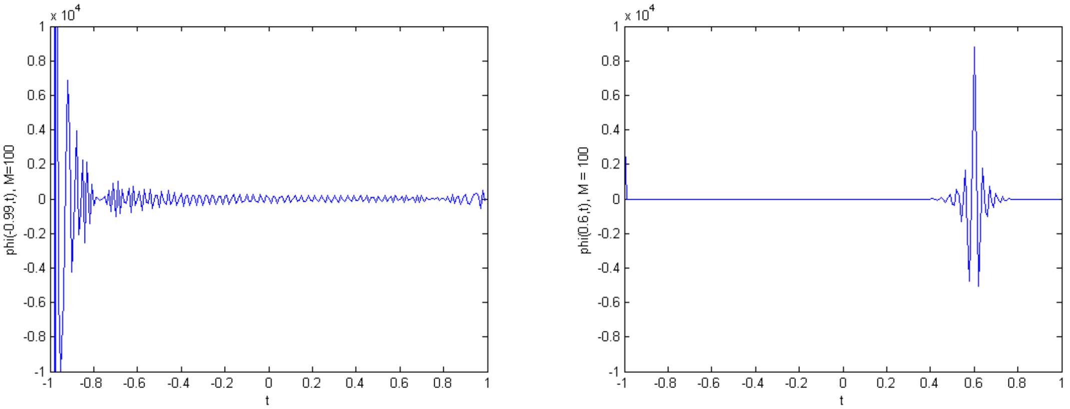
$$\begin{aligned}\Phi_M\left(\cos\theta,\cos\beta\right)\sin\theta &= \sum_{m=1}^{2M}\varphi\left(\frac{m}{M}\right)m\sin m\theta\cdot\frac{\sin m\beta}{\sin\beta}\\ &= \frac{1}{2\sin\beta}\sum_{m=1}^{2M}m\varphi\left(\frac{m}{M}\right)\left(\cos m\left(\theta-\beta\right)-\cos m\left(\theta+\beta\right)\right)\\ &= \frac{1}{2\sin\beta}\left(H_M\left(\theta-\beta\right)-H_M\left(\theta+\beta\right)\right),\end{aligned}$$

where H_M is defined by

$$H_M(t) := \sum_{m=1}^{2M}m\varphi\left(\frac{m}{M}\right)\cos(t).$$

Therefore, from (5.9),

$$\begin{aligned}f(x) &\approx \frac{1}{NK}\sum_{j=0}^{N-1}\sum_{k=0}^{K-1}\Phi_M\left(\cos\theta_k, x\cdot e_{\alpha_j}\right)\sin\theta_k R_f(\alpha_j,\cos\theta_k)\\ &= \frac{1}{2NK}\sum_{j=0}^{N-1}\sum_{k=0}^{K-1}\frac{1}{\sin\left(\arccos x\cdot e_{\alpha_j}\right)}\Big[H_m\left(\theta_k-\arccos\left(x\cdot e_{\alpha_j}\right)\right)\\ &\quad - H_m\left(\theta_k+\arccos\left(x\cdot e_{\alpha_j}\right)\right)\Big]\times R_f\left(\alpha_j,\cos\theta_k\right).\end{aligned}$$



(a) $M = 100$, $s = -0.99$ (near endpoint).

(b) $M = 100$, $s = 0.6$.

Figure 5.3: Localization of Kernel Φ_M.

Hence, it follows that

$$f(x) \approx \frac{1}{2NK} \sum_{j=0}^{N-1} \sum_{k=0}^{K-1} \frac{1}{\sqrt{1-\left(x \cdot e_{\alpha_j}\right)^2}} \Big[H_m \left(\theta_k - \arccos\left(x \cdot e_{\alpha_j}\right)\right)$$

$$- H_m \left(\theta_k + \arccos\left(x \cdot e_{\alpha_j}\right)\right)\Big] \times R_f\left(\alpha_j, \cos\theta_k\right). \tag{5.10}$$

5.2 Construction and Computation of the Ridgelet Inversion Formula

Though we developed an additional Radon inversion formula in the previous section, it doesn't quite satisfy what we've set out to accomplish. Referring back to our early discussions in this chapter, we wanted to obtain a method of reconstructing $f \in L_2(\mathcal{D})$ where the kernels used exhibited *extremely* well-localized properties. In this sense, we are referencing the sub-exponential localization characteristics given in Chapter 4. Using some of the results we presented in the previous section, we will spend the remainder of this paper developing a reconstruction which satisfies our primary goal. Recall the kernel used in (5.10),

$$H_M(t) := \sum_{m=1}^{2M} m\varphi\left(\frac{m}{M}\right)\cos(t).$$

While this kernel exhibits some localization properties, the ability to use the results given in Chapter 4 are ruined by the fact that the function $t\varphi(t)$ is not differentiable at $t = 0$, which then implies that $t\varphi(t) \notin \mathcal{S}$.

Our idea is to make use of φ, as defined (5.7), but instead incorporate a dyadic structure and then subsequently incorporate a recursive definition to develop a new cutoff function. As we will see, this new cutoff function eliminates the issue φ has with smooth differentiability at $t = 0$. The method employed is commonly used when constructing wavelets in various signal theoretic applications, and is well-known under the title *decomposition of unity.*

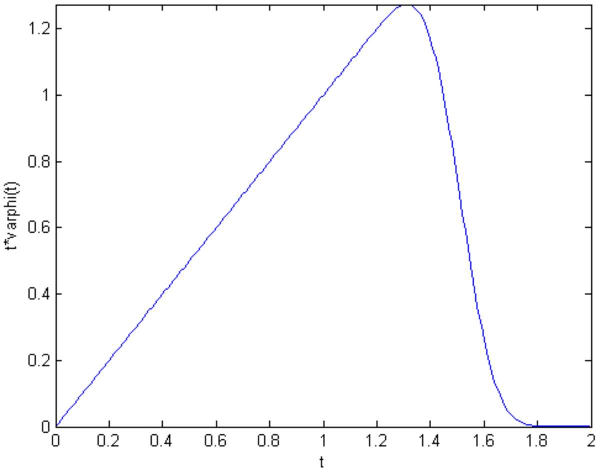


Figure 5.4: Plot of $t\varphi(t)$.

Begin by defining $\psi(t) := \varphi(t/2) - \varphi(t)$. For any integer $\nu \geq 0$ and $t \in \mathbb{R}$, subsequently define

$$\psi_\nu(t) := \begin{cases} \varphi(t), & \nu = 0 \\ \psi\left(t2^{1-\nu}\right), & \nu = 1, 2, \ldots \ . \end{cases} \tag{5.11}$$

We now show that by taking an infinite sum over ν of ψ_ν, the result is equivalent to one.

Lemma 5.2. *For $\nu \geq 0$, let ψ_ν be as defined in (5.11). Then,*

$$\sum_{\nu=0}^{\infty} \psi_\nu(t) = 1.$$

Proof. Consider the mth partial sum,

$$\begin{aligned} \sum_{\nu=0}^{m} \psi_\nu(t) &= \psi_0(t) + \psi_1(t) + \ldots + \psi_m(t) \\ &= \varphi(t) + \left(\varphi\left(\frac{t}{2}\right) - \varphi(t)\right) + \left(\varphi\left(\frac{t}{2^2}\right) - \varphi\left(\frac{t}{2}\right)\right) + \ldots \\ &\quad + \left(\varphi\left(\frac{t}{2^m}\right) - \varphi\left(\frac{t}{2^{m-1}}\right)\right) \\ &= \varphi\left(\frac{t}{2^m}\right). \end{aligned}$$

Therefore, by taking the limit as m tends towards infinity, we obtain

$$\begin{aligned}\sum_{\nu=0}^{\infty} \psi_\nu(t) &= \lim_{m\to\infty} \sum_{\nu=0}^{m} \psi_\nu(t) \\ &= \lim_{m\to\infty} \varphi\left(\frac{t}{2^m}\right).\end{aligned}$$

We argue that the right hand side of the final equality is equivalent to 1. Recall from (5.7) that $\varphi(t)$ is equal to 1 on the interval $[0,1]$. Hence, $\varphi\left(t2^{-m}\right)$ is equal to 1 on the interval $[0,2^m]$. Thus, as m tends towards infinity, $\varphi\left(t2^{-m}\right)$ is equal to 1 on the interval $[0,\infty)$. Being that φ is compactly supported on $[0,2]$, it follows then that for all $t \in \mathbb{R}$,

$$\sum_{\nu=0}^{\infty} \psi_\nu(t) = \lim_{m\to\infty} \varphi\left(\frac{t}{2^m}\right) = 1.$$

□

From here, we define the function Ψ_ν by

$$\Psi_\nu\left(s,t\right) := \sum_{m=2^{\nu-2}}^{2^\nu} \psi_\nu\left(m\right) m U_m\left(s\right) U_m\left(t\right). \tag{5.12}$$

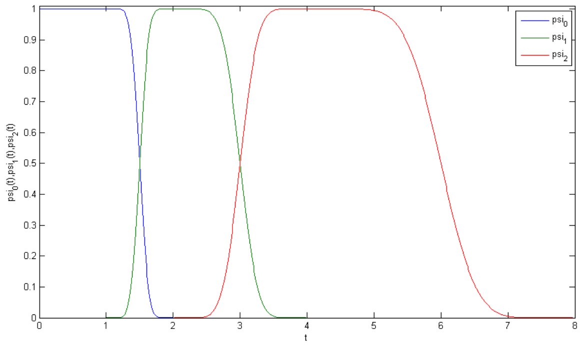


Figure 5.5: Plot of $\psi_0(t), \psi_1(t), \psi_2(t)$.

Denote by Σ_1 the sum

$$\Sigma_1 := \sum_{\nu=0}^{\infty} \sum_{m=2^{\nu-2}}^{2^\nu} \psi_\nu\left(\frac{m}{2^{\nu-1}}\right) m U_m(s) U_m(t), \tag{5.13}$$

and by Σ_2 the sum

$$\Sigma_2 := \sum_{m=0}^{\infty} mU_m(s)U_m(t). \tag{5.14}$$

The following lemma shows that these two summations are, in fact, equal.

Lemma 5.3. *Let Σ_1 and Σ_2 be defined as above. Then, $\Sigma_1 = \Sigma_2$.*

Proof. Note that since $\psi_\nu\left(m2^{1-\nu}\right)$ is compacty supported on the interval $[0, 2^\nu]$, we obtain the following equivalent definition to the one by (5.13),

$$\Sigma_1 = \sum_{\nu=0}^{\infty}\sum_{m=0}^{\infty} \psi_\nu\left(\frac{m}{2^{\nu-1}}\right) mU_m(s)U_m(t).$$

Subsequently changing the order of integration and applying the result given in Lemma 5.2 yields

$$\begin{aligned}\Sigma_1 &= \sum_{m=0}^{\infty}\sum_{\nu=0}^{\infty} \psi_\nu\left(\frac{m}{2^{\nu-1}}\right) mU_m(s)U_m(t)\\ &= \sum_{m=0}^{\infty}\left[\sum_{\nu=0}^{\infty} \psi_\nu\left(\frac{m}{2^{\nu-1}}\right)\right] mU_m(s)U_m(t)\\ &= \sum_{m=0}^{\infty} mU_m(s)U_m(t).\end{aligned}$$

Therefore, it follows from (5.14) that $\Sigma_1 = \Sigma_2$. □

Applying the foregoing result to (5.8), we obtain another reconstruction identity for $f \in L_2(\mathcal{D})$. Namely, for $x \in \mathbb{R}^2$,

$$f(x) = \frac{1}{2\pi^2}\int_0^{2\pi}\int_{-1}^{1}\sum_{\nu=0}^{\infty} \Psi_\nu\left(s, x\cdot e_\alpha\right) R_f\left(\alpha, s\right) ds d\alpha. \tag{5.15}$$

Hence, the above naturally leads to yet another approximate identity for f. For $L \geq 1$,

$$f(x) \approx \frac{1}{2\pi^2}\int_0^{2\pi}\int_{-1}^{1}\sum_{\nu=0}^{L} \Psi_\nu\left(s, x\cdot e_\alpha\right) R_f\left(\alpha, s\right) ds d\alpha. \tag{5.16}$$

To make the discretization of the above result easier, we let $s = \cos\theta$ in order to achieve

$$\begin{aligned} f(x) &\approx \frac{1}{2\pi^2}\int_0^{2\pi}\int_{-1}^{1}\sum_{\nu=0}^{L}\Psi_\nu\left(s, x\cdot e_\alpha\right)R_f\left(\alpha, s\right)dsd\alpha \\ &= \sum_{\nu=0}^{L}\frac{1}{2\pi^2}\int_0^{2\pi}\int_{-1}^{1}\Psi_\nu\left(s, x\cdot e_\alpha\right)R_f\left(\alpha, s\right)dsd\alpha \\ &= \sum_{\nu=0}^{L}\frac{1}{2\pi^2}\int_0^{2\pi}\int_{0}^{\pi}\Psi_\nu\left(\cos\theta, x\cdot e_\alpha\right)R_f\left(\alpha, \cos\theta\right)\sin\theta d\theta d\alpha. \end{aligned} \tag{5.17}$$

The resulting discrete analog to the above approximate identity is obtained by discretizing the two integrals appearing in the expression. By sampling nodes for θ and α, we obtain

$$f(x) \approx \sum_{\nu=0}^{L}\frac{1}{2\pi^2}\frac{2\pi}{2^{\nu+1}}\sum_{j=1}^{2^{\nu+1}}\frac{\pi}{2^{\nu+1}}\sum_{k=1}^{2^{\nu+1}}\Psi_\nu\left(\cos\theta_k, x\cdot e_{\alpha_j}\right)R_f\left(\alpha_j, \cos\theta_k\right)\sin\theta_k,$$

which simplifies to

$$f(x) \approx \sum_{\nu=0}^{L}\frac{1}{2^{2\nu+2}}\sum_{j=1}^{2^{\nu+1}}\sum_{k=1}^{2^{\nu+1}}\Psi_\nu\left(\cos\theta_k, x\cdot e_{\alpha_j}\right)R_f\left(\alpha_j, \cos\theta_k\right)\sin\theta_k. \tag{5.18}$$

We now turn our attention to showing that the kernel

$$\Psi_\nu\left(s,t\right) := \sum_{m=2^{\nu-2}}^{2^\nu}\psi_\nu\left(m\right)mU_m\left(s\right)U_m\left(t\right)$$

exhibits sub-exponential localization properties. Note that we can equivalently express the above as

$$\Psi_\nu\left(s,t\right) = \sum_{m=1}^{\infty}\psi\left(\frac{m}{2^{\nu-1}}\right)mU_m(s)U_m(t),$$

since $\psi\left(m2^{1-\nu}\right)$ is supported on the interval $[1,2]$. Denote by $\omega(t) := t\psi\left(t\right)$. A quick look at the figure below gives evidence that we have been relieved of the smoothness issue $t = 0$.

Therefore, since $\psi(t) \in C^\infty\left(\mathbb{R}\right)$, we have that $\omega(t) = t\psi(t) \in C^\infty\left(\mathbb{R}\right)$. Consequently, we can apply the localization result given by Theorem 4.3 to the kernel Ψ_ν.

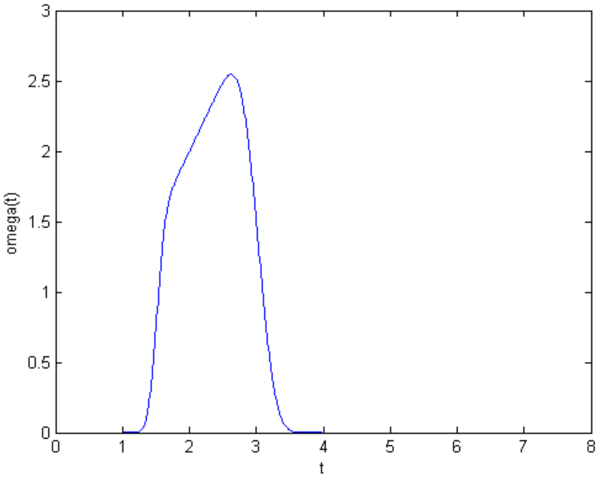


Figure 5.6: Plot of $\omega(t)$.

Recalling Theorem 4.3, it follows that for all $\sigma > 0$ there exists a constant $c_\sigma > 0$ such that

$$\left|\Psi_\nu\left(s,t\right)\right| \leq \frac{c_\sigma 2^{2\nu}}{\left(\sqrt{1-s^2}+2^{-\nu}\right)\left(\sqrt{1-t^2}+2^{-\nu}\right)\left(1+2^\nu\rho\left(s,t\right)\right)^\sigma}, \tag{5.19}$$

where $\rho\left(s,t\right) = \sqrt{\arccos(s)-\arccos(t)}$.

Drawing our attention to (5.18), note the factor $\sin\theta_k$ included in each iteration of the inner sum. Though this factor does not cause any issues with the localization of Ψ_ν (and it in fact helps), we can make use of it to further clean up the aforementioned bound. Consider that for $0 \leq \theta, \phi \leq 2\pi$,

$$\begin{aligned}\Psi_\nu\left(\cos\theta,\sin\phi\right)\sin\theta &= 2^{\nu-1}\sum_{n=1}^{\infty}\omega\left(\frac{m}{2^{\nu-1}}\right)\sin m\theta\frac{\sin m\phi}{\sin\phi}\\ &= \frac{2^{\nu-1}}{\sin\phi}\sum_{m=0}^{\infty}\omega\left(\frac{m}{2^{\nu-1}}\right)\sin m\theta\sin m\phi.\end{aligned}$$

Since $\omega(t) \in C^\infty\left(\mathbb{R}\right)$, it follows from Lemma 4.5 that for all $\sigma > 0$ there exists a cosntant $c_\sigma > 0$ such that

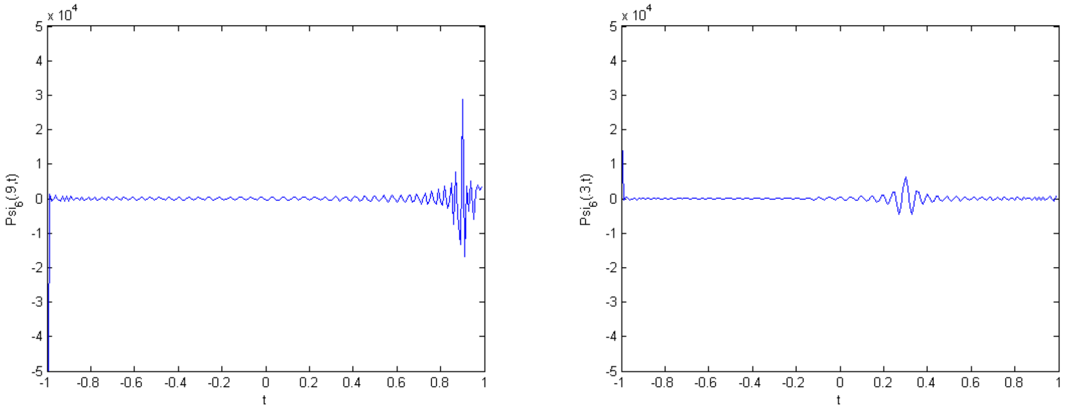
$$\left|\Psi_\nu\left(\cos\theta,\cos\phi\right)\sin\theta\right| \leq \frac{c_\sigma 2^{2\nu}}{\left|\sin\phi\right|\left(1+2^\nu\left|\theta-\phi\right|\right)^\sigma}.$$

Similarly, Lemma 4.8 yields

$$\left|\Psi_\nu\left(\cos\theta, \cos\phi\right)\sin\theta\right| \leq \frac{c_\sigma 2^{3\nu}}{\left(1 + 2^\nu\left|\theta - \phi\right|\right)^\sigma}.$$

Therefore, combining the above two inequalities, we achieve that for all $\sigma > 0$, there exists a constant $c_\sigma > 0$ such that

$$\left|\Psi_\nu\left(\cos\theta, \sin\phi\right)\sin\theta\right| \leq \frac{c_\sigma 2^{2\nu}}{\left(\sin\phi + 2^{-\nu}\right)\left(1 + 2^\nu\left|\theta - \phi\right|\right)^\sigma}.$$



(a) $\nu = 6$, $s = 0.9$ (near endpoint).

(b) $\nu = 6$, $s = 0.3$.

Figure 5.7: Localization of Kernel Ψ_ν.

This concludes our discussions on the localization of the kernel Ψ_ν, which was used in the discrete version of the inverse Radon identity given in (5.18). From (5.19), it follows that when $\cos\theta_k$ and $x \cdot e_{\alpha_j}$ are far apart, $\Psi_\nu\left(\cos\theta_k, x \cdot e_{\alpha_j}\right)$ is extremely small. This speeds up the computation time in evaluating the kernel Ψ_ν since essentially only points where $\cos\theta_k$ and $x \cdot e_{\alpha_j}$ are close together will yield non-zero values. In this sense, Ψ_ν is well-localized which gives the identity proposed in (5.18) advantages over classical Chebyshev reconstruction results.

Bibliography

[1] Ali N Akansu and Richard A Haddad, *Multiresolution signal decomposition: transforms, subbands, and wavelets*, Academic Press, 2001.

[2] Gregory Beylkin, *Discrete Radon transform*, IEEE Trans. Acoust. Speech Signal Process. **35** (1987), no. 2, 162–172. 904961

[3] Martin L. Brady, *A fast discrete approximation algorithm for the Radon transform*, SIAM J. Comput. **27** (1998), no. 1, 107–119 (electronic). 1614880

[4] Charles K. Chui, *An introduction to wavelets*, Wavelet Analysis and its Applications, vol. 1, Academic Press, Inc., Boston, MA, 1992. 1150048

[5] Arthur Erdélyi, Wilhelm Magnus, Fritz Oberhettinger, and Francesco G. Tricomi, *Higher transcendental functions. Vol. II*, Robert E. Krieger Publishing Co., Inc., Melbourne, Fla., 1981. 698780

[6] W. A. Götz and H. J. Druckmüller, *A fast digital Radon transform—an efficient means for evaluating the Hough transform*, Pattern Recognition **28** (1995), no. 12, 1985–1992. 1365481

[7] Carsten Høilund, *The radon transform*, Aalborg University: Vision, Graphics and Interactive Systems (VGIS) **12** (2007).

[8] Kamen Ivanov, Pencho Petrushev, and Yuan Xu, *Sub-exponentially localized kernels and frames induced by orthogonal expansions*, Math. Z. **264** (2010), no. 2, 361–397. 2574981

[9] Jarkko Johansson, *Radon transform in positron emission tomography*, Turku PET Center Image Processing Report Series **2** (2004).

[10] F. Natterer, *The mathematics of computerized tomography*, Classics in Applied Mathematics, vol. 32, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001, Reprint of the 1986 original. 1847845

[11] Pencho Petrushev and Yuan Xu, *Localized polynomial frames on the interval with Jacobi weights*, J. Fourier Anal. Appl. **11** (2005), no. 5, 557–575. 2182635

[12] William H. Press, *Discrete radon transform has an exact, fast inverse and generalizes to operations other than sums along lines*, PNAS **103** (2006), no. 51.

Appendix A

Source Code

A.1 Discrete Line Integral: Algorithm 1

```
function r = dlIntegral1(f,theta,s)
%   Input ~  f, N x N image
%   Input ~  theta, an angle between 0 and 180
%   Input ~  s, a distance from the origin
%   Output ~ r, sum of intensities of pixels intersected
%               by the line, weighted by lengths of the
%               intersection.

N = size(f,1);                % Calculate dimensions of image.
P = -N/2:N/2;                 % Set mesh.
sinVal = sin(theta*pi/180);   % Compute values for sine,
cosVal = cos(theta*pi/180);   % and cosine.
r = 0;                        % Initialize sum line.

% If theta is 0 or 180, find column and sum over rows.
if (theta == 0 || theta == 180)
    if (s >= -N/2 && s < N/2)
        x = floor(s)+N/2+1;
        for y = 1:N
            r = r + f(x,y);
        end
    end
elseif (0 < theta && theta < 90)
    % X and Y coordinates of the line.
    lineX = (s - P*sinVal)/cosVal;
    lineY = (s - P*cosVal)/sinVal;

    % Calculate left, right, top, and bottom exits.
    leftExit = floor(lineX(N+1)+N/2) + 1;
    rightExit =  floor(lineX(1)+N/2) + 1;
```

```
topExit = floor(lineY(1) + N/2)  + 1;
botExit = floor(lineY(N+1) + N/2) + 1;

% Reset left and right exits if out of bounds.
if (leftExit <= 0), leftExit = 1;       end
if (rightExit > N+1), rightExit = N+1; end

% Reset top and bottom exits if out of bounds.
if (topExit > N+1), topExit = N+1;  end
if (botExit <= 0), botExit = 1;      end

% Search for pixels to hit right edges.
for i = leftExit:rightExit
    x = i;
    y = N/2 - floor(lineY(i));

    % Check if pixels are inside the image region.
    if (x > 0 && x <= N && y > 0 && y <= N)
        % Compute dX and dY.
        dX = P(i) - lineX(N+1 - (y-1));
        dY = 1 - lineY(i)+floor(lineY(i));

        % Calculate length of intersection of line and pixel.
        % Reset distances if below 0 or above 1.
        if (dX > 1 || dX < 0); dX = 1; end
        if (dY > 1 || dY < 0); dY = 1; end
        pixLength = sqrt(dX^2+dY^2);

        % Add pixel coordinates and weights.
        r = r + pixLength*f(x,y);
    end
end

% Search for pixels to hit top edges.
for i = topExit:-1:botExit
    x = floor(lineX(i)) + N/2 + 1;
    y = N + 1 - i;

    % Check if pixels are inside the image region.
    if (x > 0 && x <= N && y > 0 && y <= N)
        % Compute dX and dY.
```

```
            dX = lineX(i) - floor(lineX(i));
            dY = lineY(x) - P(i);

            % Calculate length of intersection of line and pixel.
            % Reset distances if below 0 or above 1.
            if (dX > 1 || dX < 0); dX = 1; end
            if (dY > 1 || dY < 0); dY = 1; end
            pixLength = sqrt(dX^2 + dY^2);

            % Add pixel coordinates and weights.
            r = r + pixLength*f(x,y);
        end
    end
elseif (90 < theta && theta < 180)
    % X and Y coordinates on the line.
    lineX = (s - P*sinVal)/cosVal;
    lineY = (s - P*cosVal)/sinVal;

    leftExit = floor(lineX(1)+N/2) + 1;
    rightExit =  floor(lineX(N+1)+N/2) + 1;
    topExit = floor(lineY(N+1) + N/2)  + 1;
    botExit = floor(lineY(1) + N/2) + 1;

    % Reset left and right exits if out of bounds.
    if (leftExit <= 0);    leftExit = 1;     end
    if (rightExit > N+1); rightExit = N+1; end

    % Reset top and bottom exits if out of bounds.
    if (topExit > N+1); topExit = N+1; end
    if (botExit <= 0);   botExit = 1;    end
    for i = leftExit:rightExit
        x = i;
        y = N/2 - floor(lineY(i));

        % Check if pixels are inside the image region.
        if (x > 0 && x <= N && y > 0 && y <= N)
            %Compute dX and dY.
            dX = P(i) - lineX(N+1 - (y-1));
            dY = 1 - lineY(i)+floor(lineY(i));

            % Calculate length of intersection of line and pixel.
            % Reset distances if below 0 or above 1.
            if (dX > 1 || dX < 0); dX = 1; end
            if (dY > 1 || dY < 0); dY = 1; end
            pixLength = sqrt(dX^2+dY^2);

            r = r + pixLength*f(x,y);
        end
    end

    % Search for pixels to hit top edges.
    for i = botExit:topExit
        x = floor(lineX(i)) + N/2 + 1;
        y = N + 1 - i;

        % Check if pixels are inside the image region.
        if (x > 0 && x <= N && y > 0 && y <= N)
            % Compute dX and dY.
            dX = lineX(i) - floor(lineX(i));
            dY = lineY(x) - P(i);

            % Calculate length of intersection of line and pixel.
            % Reset distances if below 0 or above 1.
            if (dX > 1 || dX < 0); dX = 1; end
            if (dY > 1 || dY < 0); dY = 1; end
            pixLength = sqrt(dX^2 + dY^2);

            % Add pixel coordinates and weights.
            r = r + pixLength*f(x,y);
        end
    end
% If theta is 90, find row and sum over columns.
elseif (theta == 90)
    if (s>=-N/2 && s<N/2)
        y = floor(s)+N/2+1;
        for x = 1:N
            r = r+f(x,y);
        end
    end
end
end
```

A.2 Discrete Line Integral: Algorithm 2

```
function r = dlIntegral2(f,theta,s)
%   Input ~  f, N x N image
%   Input ~  theta, an angle between 0 and 180
%   Input ~  s, a distance from the origin
%   Output ~ r, sum of intensities of pixels intersected
%                by the line, weighted by lengths of the
%                intersection.

N = size(f,1);               % Calculate dimensions of image.
n = round(N/2);              % Center of the image.
r = 0;                       % Initialize sum line.

% Calculation of limiting pixel values depends on
% what range the angle is in.  Depending on the theta,
% we may choose to use x or y as a known value and then
% solve for the other.
if (theta~=180 && theta~=0)
    sinVal = sin(theta*pi/180); % Compute values for sine,
    cosVal = cos(theta*pi/180); % and cosine.
    a = -cosVal/sinVal;         %Slope of line
    b = s/sinVal;               %y-intercept

    if theta<=45
        ymax = min(round(-a*n+b), n-1);   % Limit search range
        ymin = max(round(a*n+b), -n);     % for pixel values.

        for y = ymin:ymax
            x = (y-b)/a;
            xfloor = floor(x);
            xup = x - xfloor;
            xlo = 1 - xup;
            x = xfloor;
            x = max(x,-n);
            x = min(x,n-2);
            r = r + xlo*f(y+n+1,x+n+1) + xup*f(y+n+1,x+n+2);
        end

    elseif (theta>45 && theta<=90)
        xmax = min(round((-n-b)/a), n-1);
        xmin = max(round((n-b)/a), -n);

        for x = xmin:xmax
            y = a*x+b;
            yfloor = floor(y);
            yup = y - yfloor;
            ylo = 1 - yup;
            y = yfloor;
            y = max(y,-n);
            y = min(y,n-2);
            r = r + ylo*f(y+n+1,x+n+1) + yup*f(y+n+2,x+n+1);
        end
    elseif (theta > 90 && theta<=135)
        xmax = min(round((n-b)/a), n-1);
        xmin = max(round((-n-b)/a), -n);

        for x = xmin:xmax
                y = a*x+b;
                yfloor = floor(y);
                yup = y - yfloor;
                ylo = 1 - yup;
                y = yfloor;
            y = max(y,-n);
            y = min(y,n-2);
            r = r + ylo*f(y+n+1,x+n+1) + yup*f(y+n+2,x+n+1);
        end

    elseif (theta > 135 && theta < 180)
        ymax = min(round(a*n+b), n-1);
        ymin = max(round(-a*n+b), -n);

        for y = ymin:ymax
            x = (y-b)/a;
            xfloor = floor(x);
            xup = x - xfloor;
            xlo = 1 - xup;
            x = xfloor;
            x = max(x,-n);
            x = min(x,n-2);
            r = r + xlo*f(y+n+1,x+n+1) + xup*f(y+n+1,x+n+2);
        end
    end
% If theta is 0 or 180, find column and sum over rows.
else
    if (s >= -n && s < n)
            y = floor(s)+n+1;
            for x = 1:N
                r = r + f(x,y);
            end
    end
end
end
```

A.3 Discrete Radon Transform

```
function  res  = myradon(f,theta,s,alg)
%   Input ~      f, an N x N image.
%   Input ~ theta, a vector of angles (in degrees).
%   Input ~      s, a vector of distances from origin.
%   Input ~ alg, a case modifier to choose which algorithm to run
%   Output ~  res, A matrix containing the projection data.
%              The rows are the distances and the columns
%              are the angles.

N = size(f, 1);     % Calculate dimensions of image.

%Calculate length of distance and theta vectors.
sL = length(s);
tL = length(theta);

%Initialize radon transform matrix.
res = cast(zeros(sL,tL),'double');

for j=1:tL
    switch alg
        case 1
            for i = 1:sL
                res(i,tL-j+1) = dlIntegral1(f,theta(j),s(i));
            end
        case 2
            for i = 1:sL
                res(i,tL-j+1) = dlIntegral2(f,theta(j),s(i));
            end
        end
    end
end
```

A.4 Classical Chebyshev Inversion Algorithm

```
function coeffMatrix = chebyCoeff(f,M,alg)
%   Input ~ f, a S x S image.
%   Input ~ M, degree of approximation.
%   Input ~ alg, a case modifier to choose which algorithm to run
%   Output ~ coeffMatrix, a matrix containing coefficients for
%                         Chebyshev reconstruction.

S = size(f,1);                              % Compute dimensions of image.
K = 4*M;                                     % Compute step size of alpha sampling.
R = ceil(sqrt(2)*S/2);                       % Calculate radius of enclosing circle.
c = cast(zeros(M,M,K),'double');             % Initialize coefficient matrix.
                                             % Prior to sum.
alpha = pi/(2*K):pi/K:(2*K-1)*pi/(2*K);
s = R*cos(alpha);                            % Sampling distances from origin,

switch alg
    case 1
        for m = 1:M
            theta = 180/m:180/m:180;
            for j=1:m
                for k=1:K
                    c(m,j,k) = dlIntegral1(f,theta(j),s(k))*sin(m*alpha(k));
                end
            end
        end
    case 2
        for m = 1:M
            theta = 180/m:180/m:180;
            for j=1:m
                for k=1:K
                    c(m,j,k) = dlIntegral2(f,theta(j),s(k))*sin(m*alpha(k));
                end
            end
        end
```

```
end

% Coefficient matrix including sum.
coeffMatrix = sum(c,3);
end

function f = iradonCheby(coeffMatrix,N)
%   Input ~ c, a matrix containing Chebyshev coefficients.
%   Input ~ N, size of f.
%   Output ~ f, a reconstruction to original image.
M = size(coeffMatrix,1);  % Calculate degree of approximation.
K = 4*M;          % Calculate step size of alpha sampling.
R = ceil(sqrt(2)*N/2);  % Calculate radius of enclosing circle.
f = cast(zeros(N,N),'double');  % Initialize reconstruction of f.

% Scan through all pixels.
for x = 1:N
    for y = 1:N
        s = x - N/2;     %Offset x and y coordinates.
        l = y - N/2;
        for m = 1:M
            theta = pi/m:pi/m:pi;
            for j = 1:m
                t = s*cos(theta(j))+l*sin(theta(j));
                u = real(chebyshev2nd(m,t/R));
                f(x,y) = f(x,y)+coeffMatrix(m,j)*abs(u);
            end
            f(x,y) = (1/(R*K))*f(x,y);
        end
    end
end
end

% Values of degree m - 1 Chebyshev polynomial of 2nd kind.
function u = chebyshev2nd(m,t)
    u = sin(m*acos(t))/sin(acos(t));
end
```