Math 579 Project 2

Due on Wednesday, May 16, 2018

Phil Warther and Patrick Vaughn

Contents

List of Figures

Tomography

Introduction

Tomography is a method used for the graphic reconstruction of the interior of a three-dimensional region. This is accomplished by taking a sequence of two-dimensional ”slice” measurements of a region. To construct these slices, X-rays are projected through a region, and the intensity of the X-ray is recorded on the other side. The goal of this process is to use the reduction in intensity to determine the amount of X-ray energy absorbed by different parts of the two-dimensional slice and compare it to the known absorption rates of different materials. This will give us the data needed to reconstruct the interior of the region.
Exploring this task is a mathematically rigorous way will require that we make a few assumptions regarding the nature of the X-rays in our measurement:

1. That the X-rays travel in a straight line, and do not refract or diffract as they pass through the region
2. That the X-rays are monochromatic. Meaning they have a single frequency
3. That the intensity, $I(\mathbf{x})$, of the measured X-rays satisfies Beer's Law
$$\frac{dI(\mathbf{x})}{ds} = -\mu(\mathbf{x})I(\mathbf{x}) \tag{1}$$
where s is the arclength along which the X-ray travels, and $\mu(\mathbf{x}) \geq 0$ for all x in our region $\mathcal{D}$.

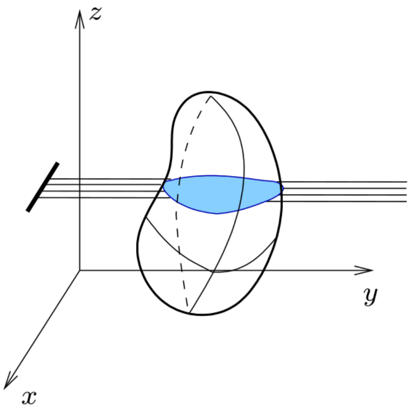


Figure 1: Slice Image (image source:[2])

Our goal becomes determining the function $\mu(\mathbf{s})$, which will be referred to as the attenuation coefficient of $\mathcal{D}$. Accomplishing this will give us an approximation of the shape and size of different materials on the interior of the region. For example, in the case of an X-ray of a human, $\mathbf{x}$ values associated with higher μ values will absorb more X-ray energy, and tend to be things like bone and cartilage. Whereas soft tissue areas would have a much lower μ values, but still higher than that of air. It should be noted that the attenuation coefficient of air is assumed to be zero.

The Radon Transform

For analytical purposes, we will represent the X-rays as a parameterized line $\ell_{t,\omega}$ (see figure 2). For our region $\mathcal{D}$, suppose that $supp(\mathcal{D}) \subset [-L, L] \times [-L, L]$, then $\ell_{\omega,t}$ is defined as $\{t\omega + s\widehat{\omega} | s \in \mathbb{R}\}$ for $0 \leq t \leq L$ and the vector $\omega = (\cos\theta, \sin\theta)^T$ is associated with a measurement angle $\theta \in [0, 2\pi)$ and $\widehat{\omega} = (-\sin\theta, cos\theta)$. The line integral of a function f along $\ell_{t,\omega}$ is defined as such,

$$\mathcal{R}f(t,\omega) \quad = \textstyle\int_{-\infty}^{\infty} f(t\omega + s\widehat{\omega})ds \tag{2}$$

Which is known as the Radon transform of the function f.
Our goal is to use a collection of line integral measurements of the attenuation coefficients of the two-dimensional region to approximate the coefficients over the entire region.
This estimation has its limitations. Suppose that the regions shown in figure (3) all have constant and equal attenuation coefficients. Then the line integral measurements would be equal for all x ray paths in the figure.

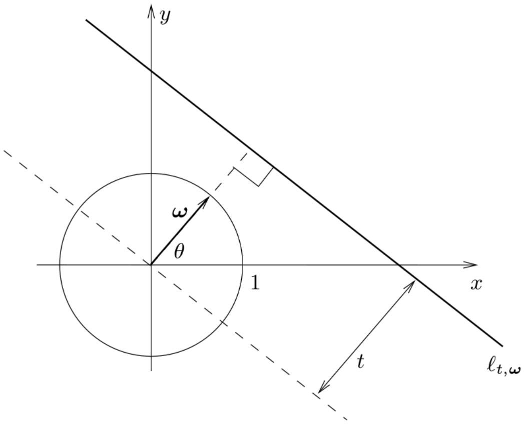


Figure 2: Measurement Angle θ and its Relation to the Parmeterized Line $\ell_{t,\omega}$ (image source:[2])

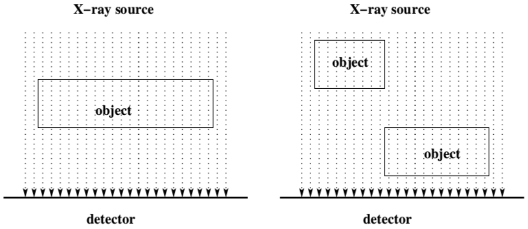


Figure 3: Line Integral Paths for one (left) versus multiple (right) regions taken at some measurement angle θ_0 (image source: [1] pg. 40)

This is problematic in that we will not be able to distinguish between the left and right images of figure(3). If we take a measurement at a second angle (figure 4) then we are able to not only differentiate between the two objects, but also we will receive different readings for different line integrals, especially near the edges of the support for these two regions.

From this it is clear that we will require measurements from multiple angles to get a meaningful estimation of the attenuation coefficients of the entire region.

Sinogram

Towards our goal of gathering X-ray information over multiple measurements, suppose for $N, M \in \mathbb{Z}^+$ we collect X-ray data at $2N-1$ points for each of M measurement angles. Remaining consistent with the parameterization of the line $\ell_{t,\omega}$ (figure 2) each measurement can be represented by the Radon transform $\mathcal{R}f(t_n, \omega_{(\theta_m)})$ where $t_n = n\frac{L}{N}$ for $n = -N, -N+1, \ldots, 0, \ldots, N-2, N-1$ and $\theta_m = m\frac{\pi}{M}$ for $m = 0, 1, 2, \ldots, M-1$. Then our data can be represented in the following way:

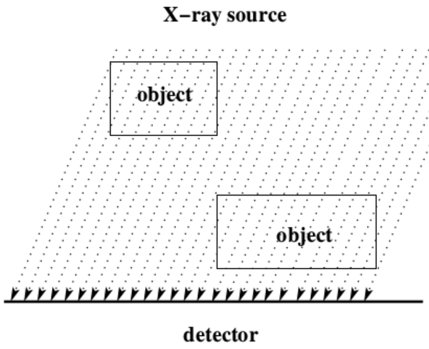


Figure 4: Line Integral Paths for X-rays taken at a second measurement angle θ_1 for the objects on right side of figure(3). (image source: [1]pg.40)

$$S \;=\begin{bmatrix} \mathcal{R}f(t_{-N},\omega_{(\theta_0)}) & \mathcal{R}f(t_{-N},\omega_{(\theta_1)}) & \dots & \mathcal{R}f(t_{-N},\omega_{(\theta_{M-1})}) \\ \mathcal{R}f(t_{-N+1},\omega_{(\theta_0)}) & \mathcal{R}f(t_{-N+1},\omega_{(\theta_1)}) & \dots & \mathcal{R}f(t_{-N+1},\omega_{(\theta_{M-1})}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{R}f(t_{N-1},\omega_{(\theta_0)}) & \mathcal{R}f(t_{N-1},\omega_{(\theta_1)}) & \dots & \mathcal{R}f(t_{N-1},\omega_{(\theta_{M-1})}) \end{bmatrix} \tag{3}$$

Central Slice Theorem

The Central Slice Theorem is an integral part of our reconstruction methods. The theorem states: for an $f \in L^1(\mathbb{R}^2)$, which is absolutely integrable over every line in $\mathbb{R}^2$,

$$\widetilde{\mathcal{R}f}(r,\boldsymbol{w}) = \int_{-\infty}^{\infty} \mathcal{R}f(t,\boldsymbol{w})e^{-itr}dt = \mathcal{F}f(r\boldsymbol{w}) \tag{4}$$

For all $r \in \mathbb{R}$ and $\boldsymbol{w} \in S^1$.

Where $\widetilde{\mathcal{R}f}(r,\boldsymbol{w})$ is the 1-D Fourier transform of $\mathcal{R}f(t,\boldsymbol{w})$ in the t variable, and $\mathcal{F}f(r\boldsymbol{w})$ is the 2-D Fourier transform of $f(t\boldsymbol{w})$.

What this allows us to do is reconstruct the Fourier transform of the region using the Radon transform information found in the sinogram.

Filtered Back Projection (FBP)

Derivation

Suppose that the function f represents our attenuation coefficients, we also the sinogram data containing $\mathcal{R}f(t,\omega_{(\theta)})$, the Radon transform of the coefficient data. The Filtered Back Projection algorithm inverts the Radon transform information to give us the function f. Consider the Fourier inversion formula for $f \in L^1(\mathbb{R}^2)$:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \widehat{f}(\xi)e^{-i\langle \mathbf{x},\xi\rangle}d\xi \tag{5}$$

By switching to polar coordinates and applying the Central Slice Theorem (4), we obtain the following.

$$
\begin{aligned}
f(\mathbf{x}) &= \frac{1}{(2\pi)^2}\int_0^{2\pi}\int_0^{\infty}\widehat{f}(r,\omega_{(\theta)})e^{-i\langle\mathbf{x},\omega_{(\theta)}\rangle r}rdrd\theta \\
&= \frac{1}{(2\pi)^2}\int_0^{2\pi}\int_0^{\infty}\widetilde{\mathcal{R}f}(r,\omega_{(\theta)})e^{-i\langle\mathbf{x},\omega_{(\theta)}\rangle r}rdrd\theta \\
&= \frac{1}{(2\pi)^2}\int_0^{\pi}\int_{-\infty}^{\infty}\widetilde{\mathcal{R}f}(r,\omega_{(\theta)})e^{-i\langle\mathbf{x},\omega_{(\theta)}\rangle r}|r|drd\theta \qquad (6)
\end{aligned}
$$

This can be used directly to compute f.
If we consider

$$
\mathcal{G}(\mathcal{R}f)(t,\omega_{(\theta)}) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\widetilde{\mathcal{R}f}(r,\omega_{(\theta)})e^{irt}|r|dr \qquad (7)
$$

to be a shift invariant filter in the t variable, we get the following:

$$
f(\mathbf{x}) = \frac{1}{2\pi}\int_0^{\pi}\mathcal{G}(\mathcal{R}f)(\langle\mathbf{x},\omega_{(\theta)}\rangle, \omega_{(\theta)})d\theta \qquad (8)
$$

Which is the Filtered Back Projection formula.

Implementation

To implement Filtered Back Projection we will us the following numerical approximation of equation (8). So we have

$$
\begin{aligned}
f(\mathbf{x}) &\approx \frac{1}{2\pi}\sum_{k=0}^{M-1}\mathcal{G}(\mathcal{R}f)(\langle\mathbf{x},\omega_{(\theta)}\rangle, \omega_{(\theta)})\frac{\pi}{M} \\
&\approx \frac{1}{2M}\sum_{k=0}^{M-1}\mathcal{G}(\mathcal{R}f)(\langle\mathbf{x},\omega_{(\theta)}\rangle, \omega_{(\theta)}) \qquad (9)
\end{aligned}
$$

Implementing the filter described in equation (7) is rather difficult to apply, additionally, the fact that this filter will accentuate very high frequencies which will create errors when confronted with high frequency noise. To address this we filter $\widetilde{\mathcal{R}f}(r,\omega_{(\theta)}$ with a band-limited filter of our choosing as an approximation of $|r|$, which we will define as $\widehat{\phi}(r)$. So by the Convolution Theorem and Fourier inversion, we can approximate (7) as

$$
\begin{aligned}
\mathcal{G}(\mathcal{R}f)(t,\omega_{(\theta)}) &\approx \frac{1}{2\pi}\int_{-\infty}^{\infty}\widetilde{\mathcal{R}f}(r,\omega_{(\theta)})\widehat{\phi}(r)e^{irt}dr \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}\mathcal{F}(\mathcal{R}f(r,\omega_{(\theta)} * \phi(r))e^{irt}dr \\
&= \Big(\mathcal{R}f * \phi\Big)(t,\omega_{(\theta)}) \qquad (10)
\end{aligned}
$$

Equation (11) can be approximated by the discrete convolution formula

$$
\Big(\mathcal{R}f * \phi\Big)(t,\omega_{(\theta)}) \approx \sum_{j=-N}^{N-1}\mathcal{R}f(s_j,\omega_{(\theta)})\phi(t-s_j) \qquad (11)
$$

In practice, however, we will compute this using the Convolution theorem for the discrete Fourier transform. So our filtered approximation, $Q_\phi f(t,\omega)$ is defined

$$
\begin{aligned}
Q_\phi f(t_n,\omega_{\theta_k}) &= \Big(\mathcal{R}f \star \phi\Big)_n \\
&= \mathcal{F}^{-1}\Big(\Big\langle\widetilde{\mathcal{R}f}_j \cdot \widehat{\phi}_j\Big\rangle_{j=-N}^{N-1}\Big)_n \qquad (12)
\end{aligned}
$$

Where $\widehat{\mathcal{R}f}_j$ and ϕ_j are the j^{th} entry of the sequence of the discrete Fourier transform of Radon transform and ϕ function respectively.
The matlab function filteredBackProj.m applies these methods in the following manner

1. Calculate the $\widehat{\phi}(r)$ as a vector of discrete values, $\widehat{\phi}$, over our frequency range. By the construction of $\widehat{\phi}$ this is considered to be the discrete Fourier transform (DFT) of our filter function.

2. Compute $Q_\phi f$ by taking the DFT of the Radon transform values given in the Sinogram, and doing an element-wise multiplication with $\widehat{\phi}$ vector (12).

3. Apply back projection to the filtered data. This is done by averaging the values computed at some $\mathbf{x} \in \mathcal{D}$ over the M measurement angles θ_k. These values become the approximation of $f(\mathbf{x})$, our attenuation coefficients

Applied Filters

The filtering funtion $\widehat{\phi}(r)$ is defined

$$\widehat{\phi}(r) = A(r)|r| \tag{13}$$

Where $A(r)$ is known as the apodizing function. For some bandlimit, B, three apodizing functions were chosen for our experiments.

- **Ram-Lak Filter**: $A(r) = \chi_{[-B,B]}(r)$
- **Hanning Filter**: $A(r) = \cos^2(\frac{\pi r}{2B})\chi_{[-B,B]}(r)$
- **Shepp-Logan Filter**: $A(r) = |\text{sinc}(\frac{rd}{2})|^3$ where $d = \frac{\pi}{B}$

Where the function $\chi_{[-B,B]}(r)$ is an indicator function we use to enforce band-limitation, and is defined

$$\chi_{[-B,B]}(r) = \begin{cases} 1 & \text{if } -B \leq r \leq B \\ 0 & \text{else} \end{cases} \tag{14}$$

The frequency response of these filters can be seen in figure (5).

Implementation Issues

Since our discrete measurement angles were between 0 and $\pi - \frac{\pi}{M}$, the final back projection iteration left errors in the form of shadows in the final image. This problem was addressed using the following symmetric property of the Radon transform and, consequently, the sinogram:

$$\mathcal{R}f(t,\theta) = \mathcal{R}f(-t,-\theta) = \mathcal{R}f(-t,\theta+\pi)$$

This property allowed us to concatenate the sinogram matrix with an up-down flipped version of itself to simulate more measurements. This redundancy in the data allowed the elimination shadows in the final approximation of the attenuation coefficients.
Another issue that arose was that of negative attenuation coefficients, which were very prone to occurrence outside of the support of $\mathcal{D}$. While this was not problematic to the algorithm itself, by the non-negative nature of the attenuation coefficients that we are attempting to recover, any negative resulting values can be assumed assumed as aliasing and numerical error. So all negative values were set to zero in the final $f(\mathbf{x})$ output.

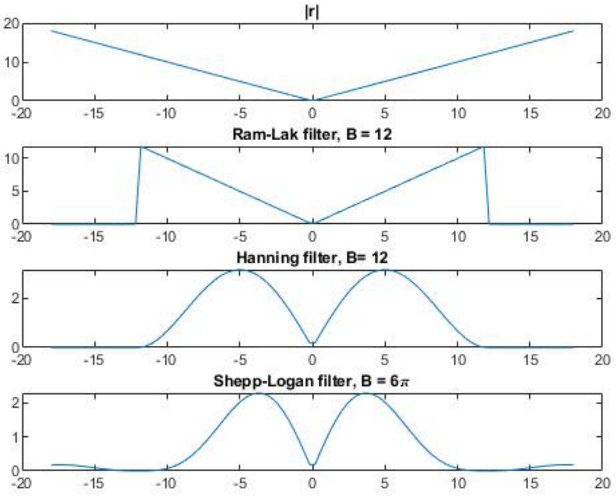


Figure 5: Comparison between different filter frequency responses.

Direct Fourier Inversion (DFI)

DFI Algorithm

The method for Direct Fourier Inversion is derived from the Central Slice Theorem. We want to use eq. 4 in order to reconstruct f from our sinogram:

$$f = \mathcal{F}^{-1}\widetilde{\mathcal{R}f}$$

Using our assumption that $\mathcal{R}f(t, \boldsymbol{w}$ is L-Periodic in t, and the definition for Recall from $\widetilde{\mathcal{R}f}(r, \boldsymbol{w})$ from eq 4, we reduce to:

$$\widetilde{\mathcal{R}f}(r, \boldsymbol{w}) = \int_{-\infty}^{\infty} \mathcal{R}f(t, \boldsymbol{w}) e^{-itr} dt$$
$$\widetilde{\mathcal{R}f}(r, \boldsymbol{w}) = \int_{-L}^{L} \mathcal{R}f(t, \boldsymbol{w}) e^{-itr} dt$$

We only have a discrete sample of the continuous Radon transform, represented in our sinogram data. Our sample data takes the form: $\mathcal{R}f(t_j, \boldsymbol{w}(\theta_k))$, where $t_j = \frac{jL}{N}$ uniformly discretizes [-L,L], j = -N, ..., N, with time step $\frac{L}{N}$ and $_k = \frac{k\pi}{M}$ uniformly discretizes [0, 2π] and k = 0, 1, ... 2M. Since the Radon transform is 2π-periodic and odd symmetric across π, we only need to use k = 0, 1, ... M data. We use a left Riemann sum to approximate $\widetilde{\mathcal{R}f}$ as follows:

$$\widetilde{\mathcal{R}f}(r, \theta_k) = \sum_{j=-N}^{N-1} \mathcal{R}f(t_j, \theta_k) e^{-it_j r} \frac{L}{N} \tag{15}$$

Since $t_j = \frac{jL}{N}$, we want to sample r in a way that we can use the FFT algorithm in our code. Let $r_n = \frac{n\pi}{L}$. Then above reduces to:

$$\begin{aligned}
\widetilde{\mathcal{R}f}(r_n,\theta_k) &\approx \frac{L}{N}\sum_{j=-N}^{N-1}\mathcal{R}f(t_j,\theta_k)e^{-it_jr_n}\\
&= \frac{L}{N}\sum_{j=-N}^{N-1}\mathcal{R}f(t_j,\theta_k)e^{\frac{-i\pi jn}{N}}\\
&= \frac{L}{N}\sum_{j=-N}^{N-1}\mathcal{R}f(t_j,\theta_k)e^{\frac{-i2\pi jn}{2N}} \qquad (16)\\
&= \frac{L}{N}DFT(\mathcal{R}f(t_j,\theta_k))
\end{aligned}$$

Where DFT is the Discrete Fourier transform. Then in order to undo the 2-D Fourier transform, we seek to apply a numerical scheme to $\widetilde{\mathcal{R}f}(r_n,\theta_k)$. Not having such an inversion formula that works on polar coordinate data, we need to approximate our polar data into a uniform grid of Cartesian coordinate data. We want our new coordinate mesh to fit on $[\frac{-N\pi}{L},\frac{N\pi}{L}]\times[\frac{-N\pi}{L},\frac{N\pi}{L}]$.

In general the 2-D Inverse Fourier transform of $\widetilde{\mathcal{R}f}(\xi,\eta)$ is:

$$f(x,y)=\frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\widetilde{\mathcal{R}f}(\xi,\eta)e^{i(x\xi+y\eta)}d\xi d\eta$$

Since we are assuming our data is 2L-periodic, we can restrict the domains:

$$f(x,y)=\frac{1}{4\pi^2}\int_{\frac{-N\pi}{L}}^{\frac{N\pi}{L}}\int_{\frac{-N\pi}{L}}^{\frac{N\pi}{L}}\widetilde{\mathcal{R}f}(\xi,\eta)e^{i(x\xi+y\eta}d\xi d\eta$$

We approximate this with a Riemann sum, we discretize x to $x_j=\frac{jL}{N}$ and y to $y_k=\frac{kL}{N}$.

$$\begin{aligned}
f(x_j,y_k) &\approx \frac{1}{4\pi^2}\sum_{n=-N}^{N}\sum_{l=-N}^{N}\widetilde{\mathcal{R}f}(\xi_n,\eta_l)e^{i(x_j\xi_n+y_k)\eta_l}\frac{\pi}{L}\frac{\pi}{L}\\
&= \frac{1}{4L^2}\sum_{n=-N}^{N}\sum_{l=-N}^{N}\widetilde{\mathcal{R}f}(\xi_n,\eta_l)e^{i(\frac{jLn\pi}{NL}+\frac{kLl\pi}{NL})}\\
&= \frac{1}{4L^2}\sum_{n=-N}^{N}\sum_{l=-N}^{N}\widetilde{\mathcal{R}f}(\xi_n,\eta_l)e^{\frac{i\pi(jn+kl)}{N}}\\
&= \frac{1}{4L^2}\sum_{n=-N}^{N}\frac{2N}{2N}\sum_{l=-N}^{N}\frac{2N}{2N}\widetilde{\mathcal{R}f}(\xi_n,\eta_l)e^{\frac{i2\pi(jn+kl)}{2N}}\\
&= \frac{N^2}{L^2}\sum_{n=-N}^{N}\frac{1}{2N}\sum_{l=-N}^{N}\frac{1}{2N}\widetilde{\mathcal{R}f}(\xi_n,\eta_l)e^{\frac{i2\pi(jn+kl)}{2N}} \qquad (17)\\
&= \frac{N^2}{L^2}IDFT(\widetilde{\mathcal{R}f}(\xi_n,\eta_l))
\end{aligned}$$

IDFT is the Inverse Discrete Fourier Transform.

The Algorithm is outlined as follows:

1. 1-D Fourier transform the sinogram data.
2. Convert from polar to Cartesian coordinates and interpolate.
3. 2-D Fourier transform the Cartesianized data.

DFI and FBP Results

In order to compare DFI and FBP, we will apply them to the following image's sinogram data, shown in Figure 6. S1, not shown, only differs to S2 by having less angles taken in the Discrete Radon transform. Note that most figures in this section use a color map to represent intensity value, with a scale where blue represents a small value and red represents a large value.

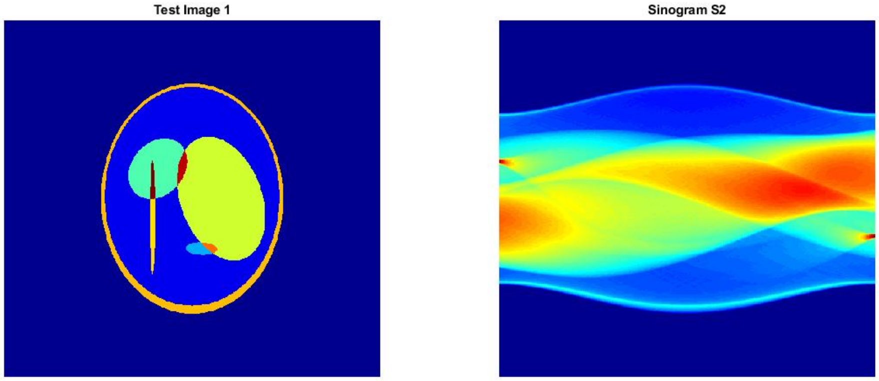


Figure 6: (1) An image of attenuation coefficients 513 × 513 pixels. (2) The Discrete Radon Transform of (1), 513 × 180 pixels.

Our DFI code produced the images shown in Figure 7. Artifacting can be seen more clearly in the negatives. Notice that refining the θ mesh did not clear up the artifacting at all. This demonstrates some of the difficulties the method has with precision, probably from having to interpolate the polar grid data onto a Cartesian mesh.

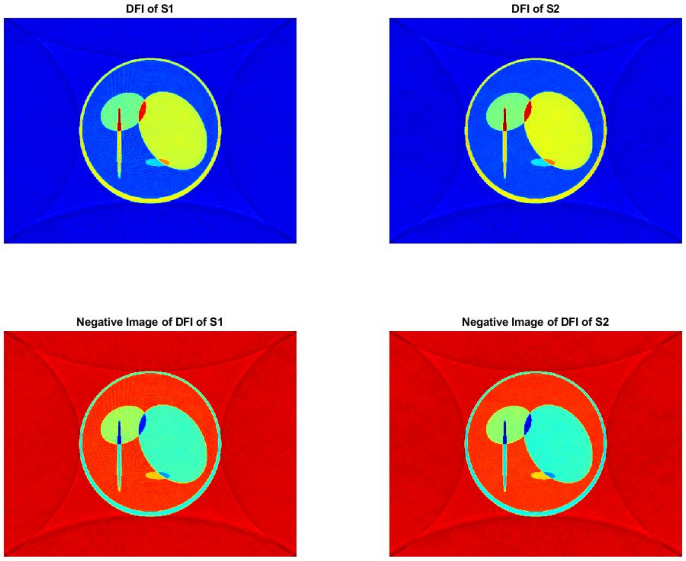


Figure 7: (1) The DFI of S1, (2) The DFI of S2, (3) Negative image of (1), (4) Negative image of (2).

Next we take a look at reconstructions with the FBP method. Figure 8, 9, and 10 we can see the effects of parameter B. If B is too small, the frequency is too cut off and we lose a sharpness is features. This is because sharp features have higher frequencies that soft features, so by cutting off too much of the high frequency data with the filter in the frequency domain, we lose sharpness of the image after the IDFT. Also notice the wavy behavior of the Ram-Lak reconstructions in Figure 8. Since the Ram-Lak filter doesn't have a smooth IDFT, we see aliasing in the higher value regions of the image. Hanning and Shepp-Logan filters don't have this issue. The Hanning filter reconstructions are not a crisp as ones produced using the Shepp-Logan filter.

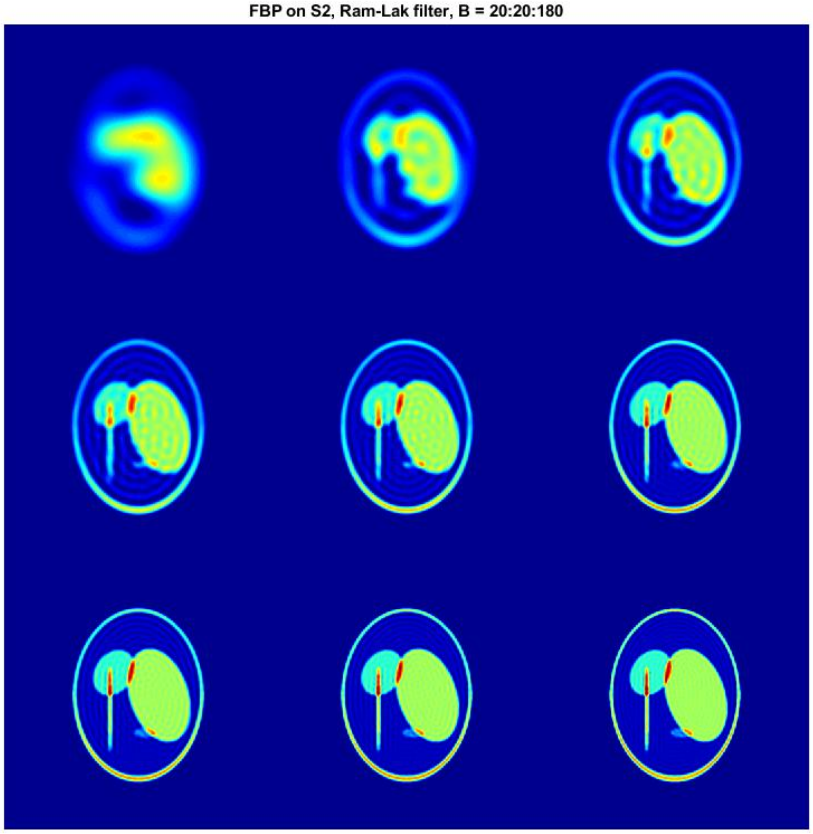
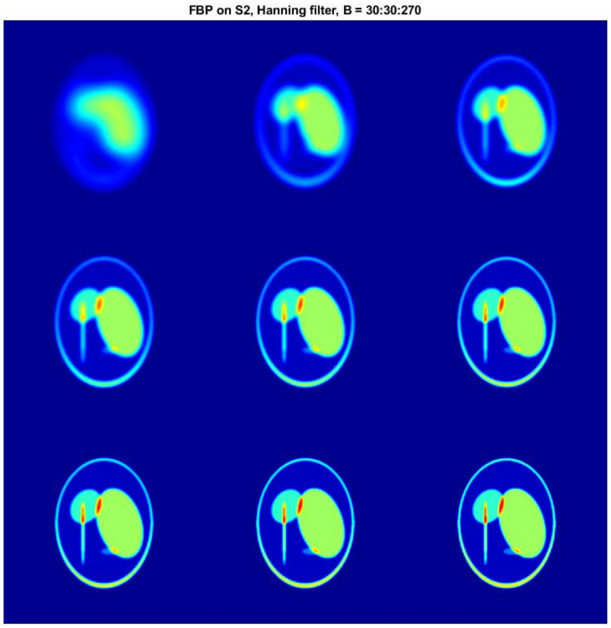
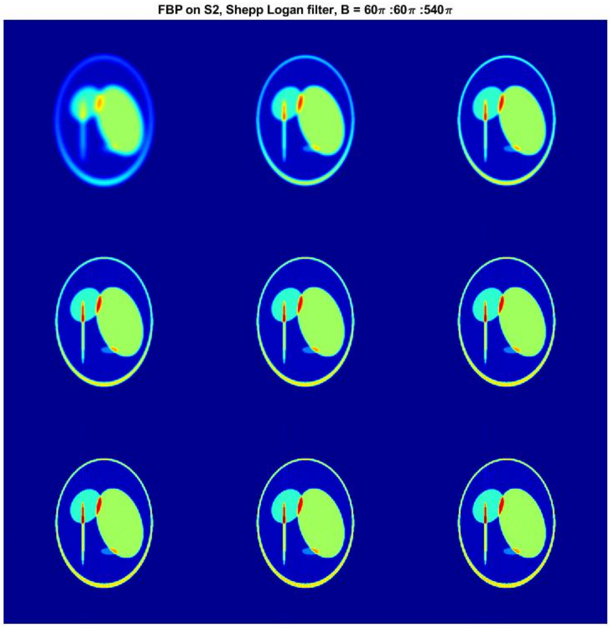


Figure 8: 9 reconstructions from S2 via FBP with Ram-Lak filter using B ranging from 20 to 180.



FBP on S2, Hanning filter, B = 30:30:270

Figure 9: 9 reconstructions from S2 via FBP with Hanning filter using B ranging from 30 to 270.



FBP on S2, Shepp Logan filter, B = 60π :60π :540π

Figure 10: 9 reconstructions from S2 via FBP with Shepp-Logan filter using B ranging from 60π to 540π.

Figure 11 and 12 compare the absolute errors between the original image and the different reconstructions from S1 and S2, Discrete Radon transforms of the image. The L^1 and L^∞ errors are included in the figures as well. Overall the DFI reconstructions have the least error. Shepp-Logan filter reconstructions outperform the Hanning and Ram-Lak FBP reconstructions. Notice that the artifacting issue of the DFI reconstructions are noticeable on the error plots. Doubling the sampling rate for the sinogram over decreased the errors for the DFI reconstruction, but had almost no effect for FBP errors. This means the error terms for FBP are lead by some other factor. From the DFI algorithm, more θ measurements leads to better interpolations from the polar to Cartesian step. In FBP, more θ measurements effect the back projection step. Back projection is an averaging process and increasing the number of estimates being averaged does not increase the overall accuracy if the estimates themselves don't increase in accuracy.

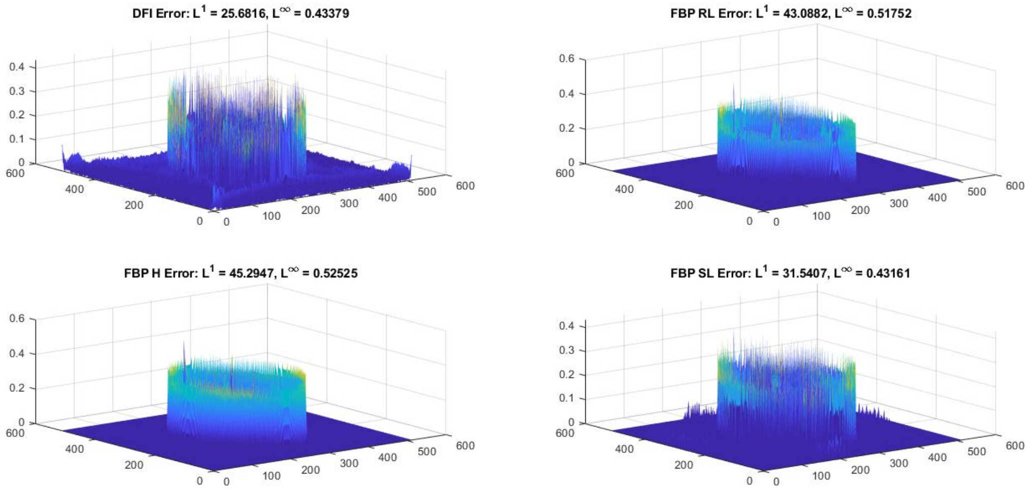


Figure 11: Plots of error between different reconstructions from S1: (1) DFI error, (2) FBP error with Ram-Lak filter, (3) FBP error with Hanning filter,(4) FBP error with Shepp-Logan filter.

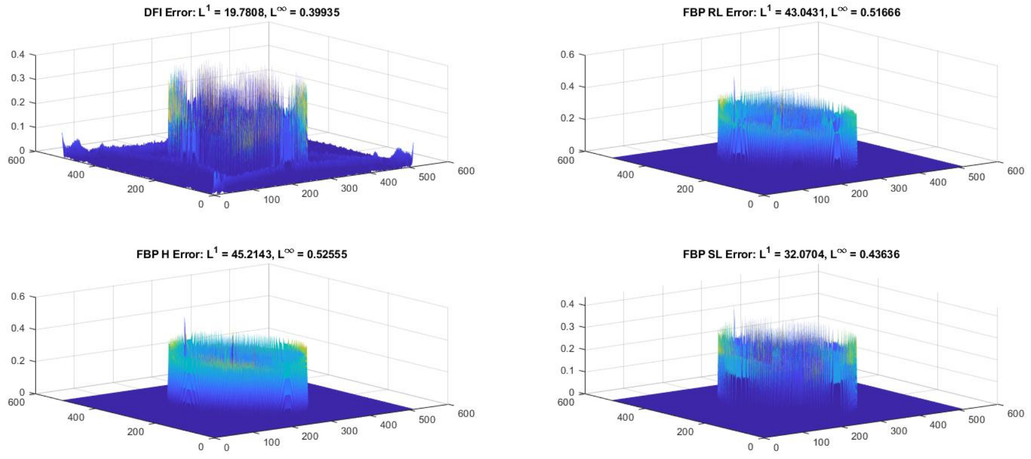


Figure 12: Plots of error between different reconstructions from S2: (1) DFI error, (2) FBP error with Ram-Lak filter, (3) FBP error with Hanning filter,(4) FBP error with Shepp-Logan filter.

Notice the sharp edges of the original image. In order for FBP to capture the very high frequency data of the cript edges, B would need to be very large. Even with a large B value, from the algorithm, we are using a DFT and an IDFT, which will have a finite support anyways, so increases the B value will only increase the accuracy by so much until it exceeds the dimensions of our t discretization. Thus for images like this, we can only do so much to improve the accuracy of the reconstructions before we need to refine the sampling mesh of the Radon transform further. Also, it is dangerous to use very large B values due to them enhancing noise.

In order to compare the methods further, consider the 5 sinograms displayed in Figure 13.

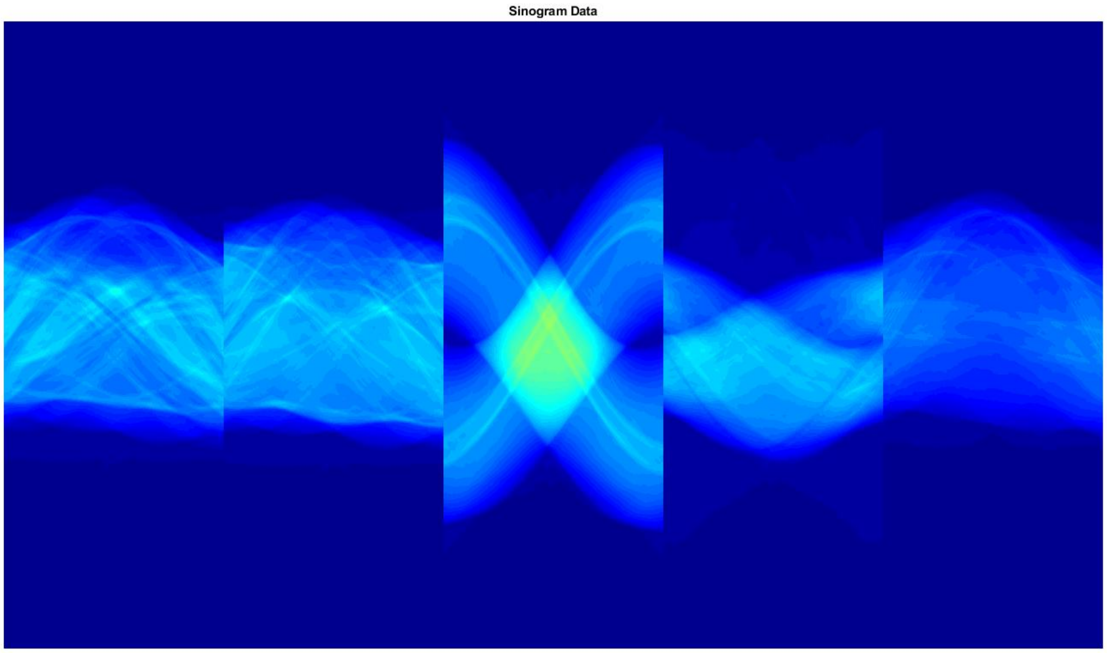


Figure 13: 5 sinograms: S1, S2, S3, S4, and S5 each with dimension 513 × 180 pixels.

Figures 14, 15, 16, 17, and 18 display reconstruction attained from the 5 different sinograms respectively. Figures 14, 15, and 18 look like different cross sections of a persons head, Figure 16 could be the cross section of a pair of legs, and Figure 17 might be the cross section of a pair of feet. Artifacting can be seen in each DFI reconstruction, most apparent in the skull cross sections as a shadow.

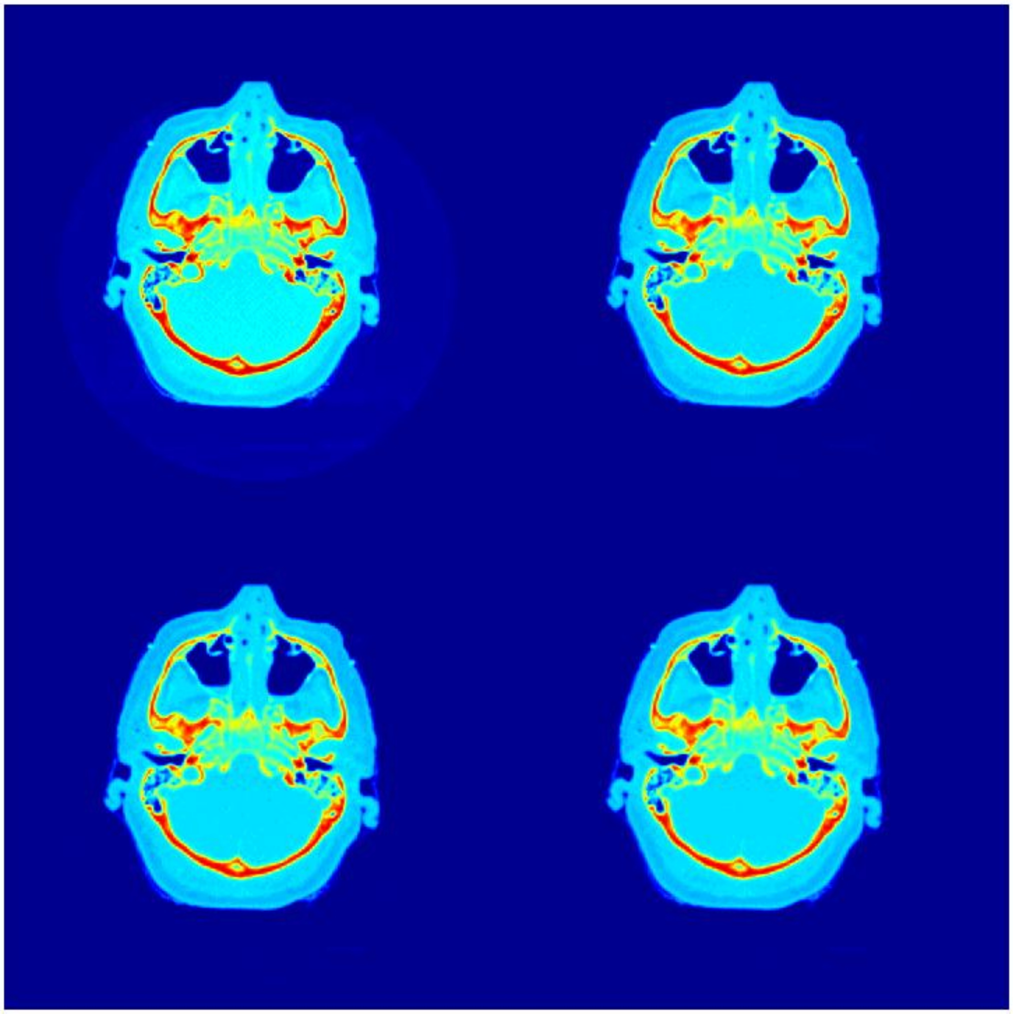


Figure 14: S1 reconstructions: (1) DFI, (2) FBP with Ram-Lak filter, (3) FBP with Hanning filter,(4) FBP with Shepp-Logan filter. Each image is zoomed and centered.

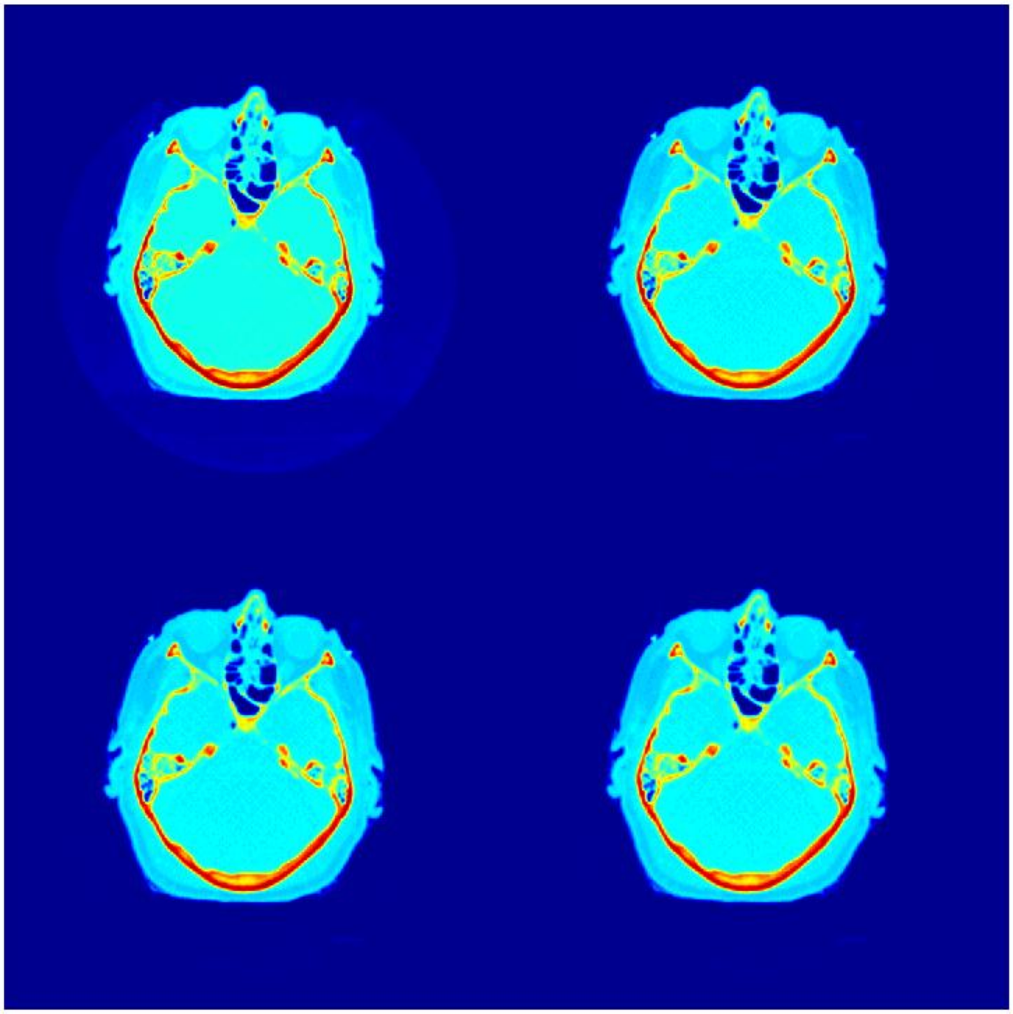


Figure 15: S2 reconstructions: (1) DFI, (2) FBP with Ram-Lak filter, (3) FBP with Hanning filter,(4) FBP with Shepp-Logan filter. Each image is zoomed and centered.

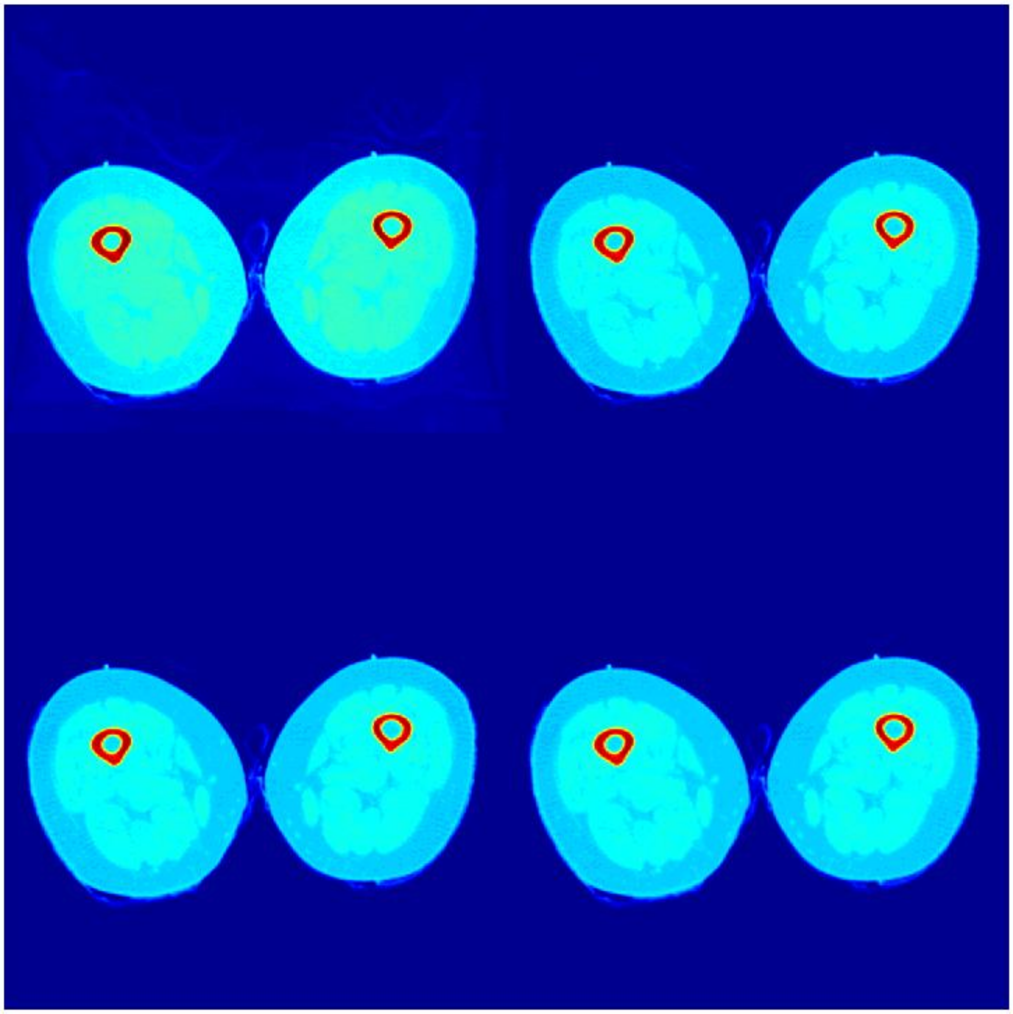


Figure 16: S3 reconstructions: (1) DFI, (2) FBP with Ram-Lak filter, (3) FBP with Hanning filter,(4) FBP with Shepp-Logan filter. Each image is zoomed and centered.

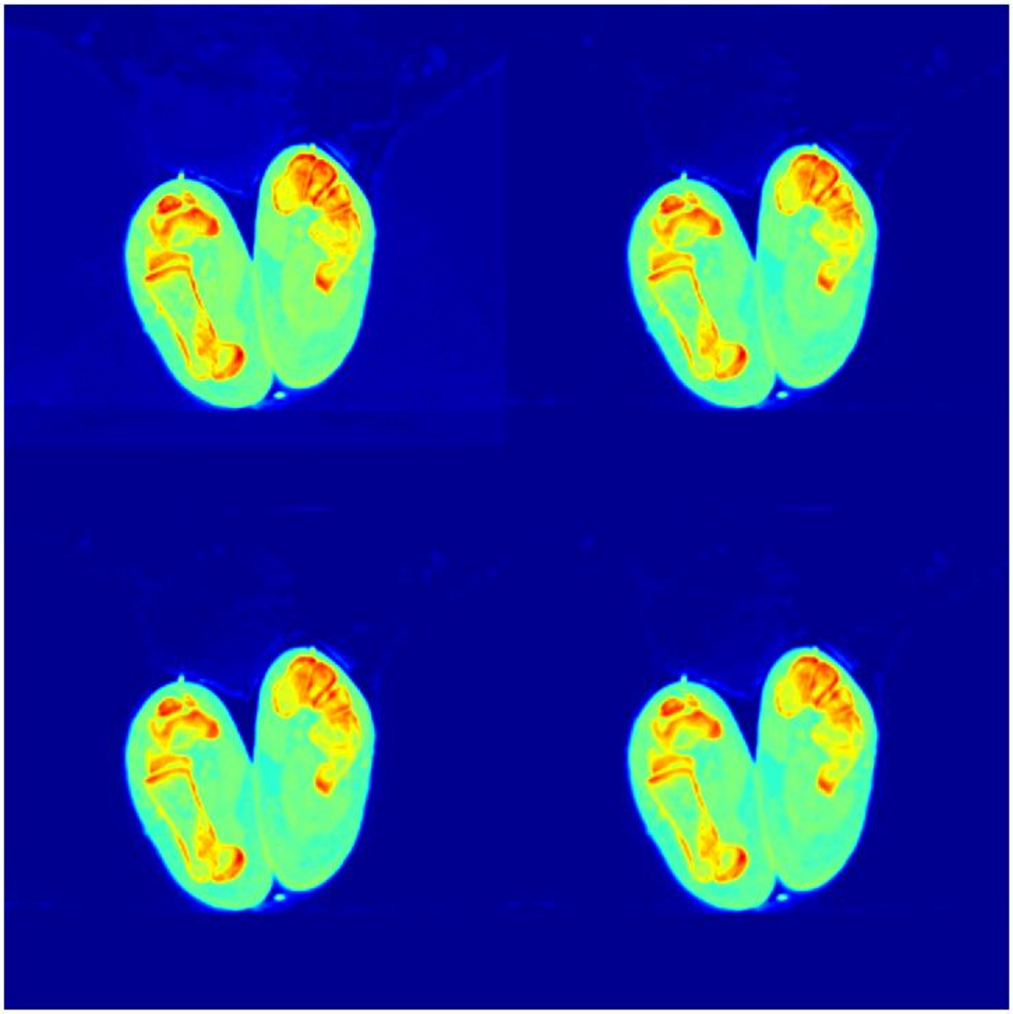


Figure 17: S4 reconstructions: (1) DFI, (2) FBP with Ram-Lak filter, (3) FBP with Hanning filter,(4) FBP with Shepp-Logan filter. Each zoomed image is and centered.

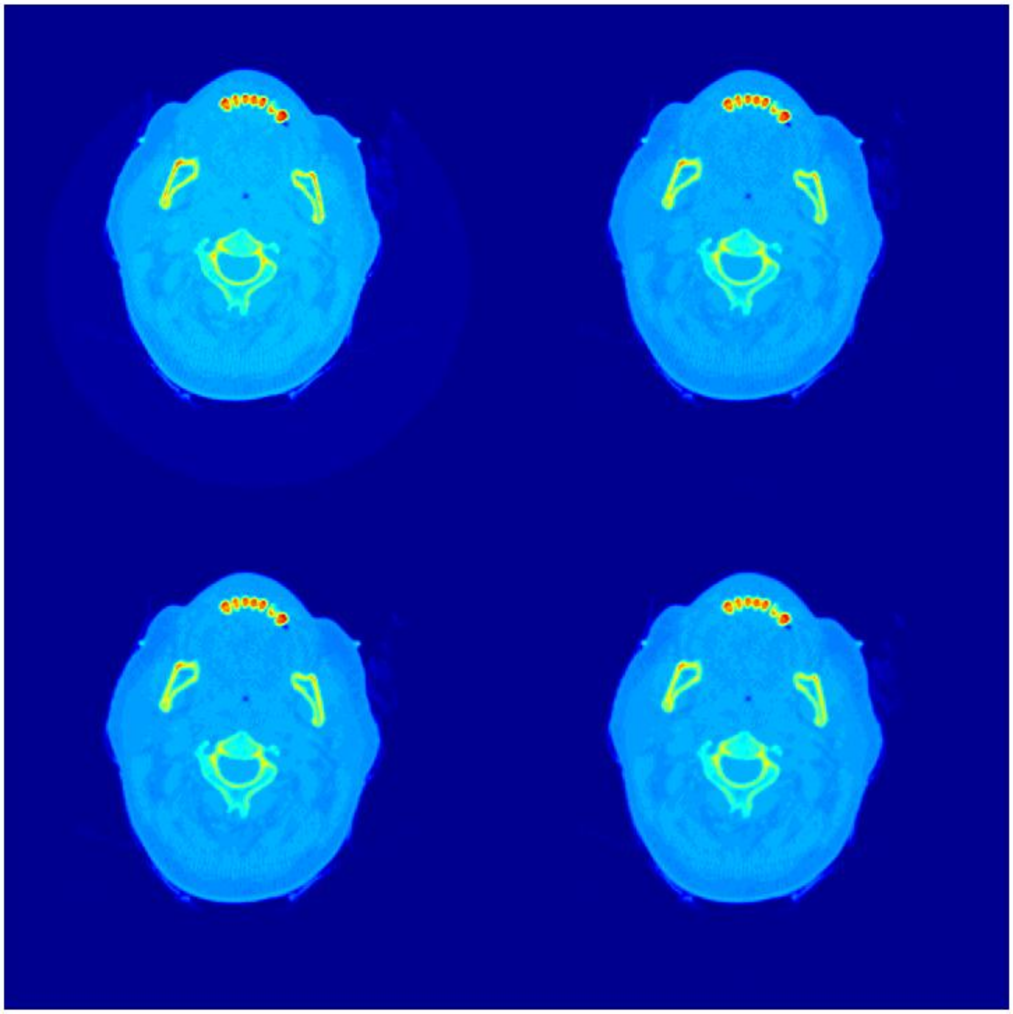


Figure 18: S5 reconstructions: (1) DFI, (2) FBP with Ram-Lak filter, (3) FBP with Hanning filter,(4) FBP with Shepp-Logan filter. Each image is zoomed and centered.

It was difficult to see the artifacting issues caused by DFI, so we took the absolute difference between the DFI reconstructions and the Shepp-Logan filter reconstructions, and re-scaled for clarity. Figure 19 shows more clearly the undesired arc patterns that were seen in Figure 7. It also shows more severe error the shadowy region just outside of each scanned objects support.

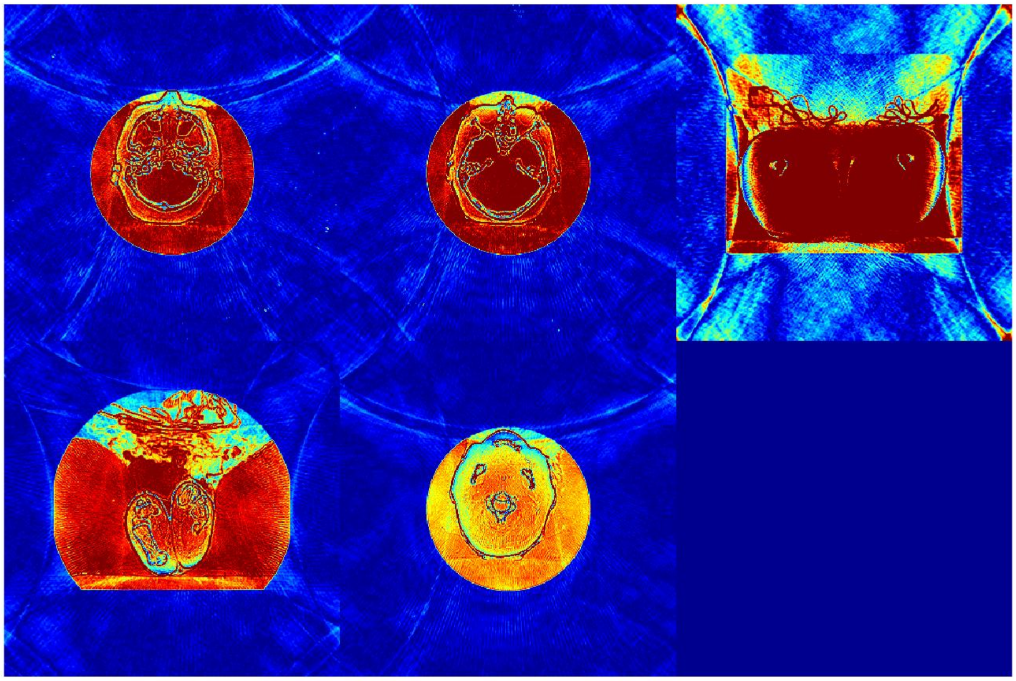


Figure 19: Absolute difference between DFI and FBP - Shepp Logan reconstructions.

Since one of the advantages of FBP over DFI is noise reduction, we compared the methods on some noisy data. We took the sinogram S1 and introduced some salt and pepper noise. The image was corrupted with between 5% and 20% noise and the resulting images displayed in Figure 20

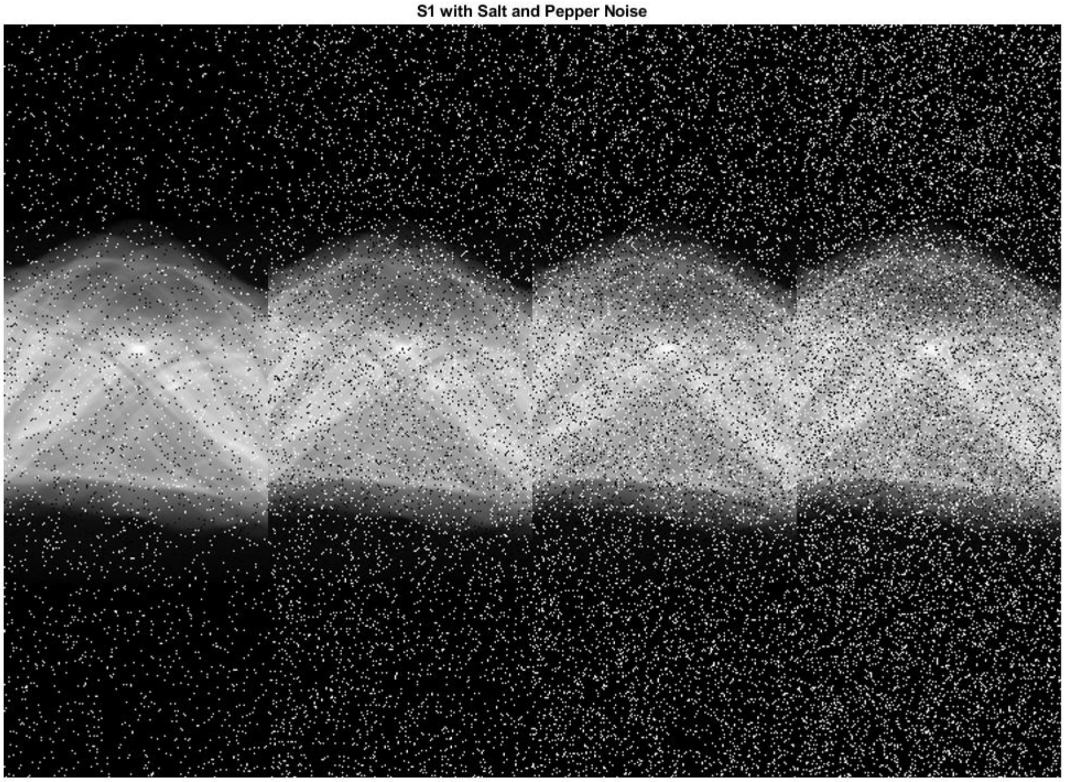


Figure 20: S1 corrupted by varying degrees of salt and pepper noise, corruption rates: (1) 0.05, (2) 0.10, (3) 0.15, (4) 0.20.

After applying the noise to the sinograms, we reconstructed the image with DFI and FBP. Figure 21 displays the results. The first row, the reconstructions from DFI, are all very noisy. The method has no inherent features to mitigate the noise introduced to the data. The second row, data processed by FBP with the Ram-Lak filter, reduces the noise, but it is still present, and at greater corruption levels its harder to see the image. Hanning and Shepp-Logan FBP results have the least noisy reconstructions of the image. However, even though noise is reduced around the area of interest, the shape itself is still blurred, due to having to have their B parameter low enough to combat the noise. Also, with 15% or more noise, all of the reconstructions become very noisy.

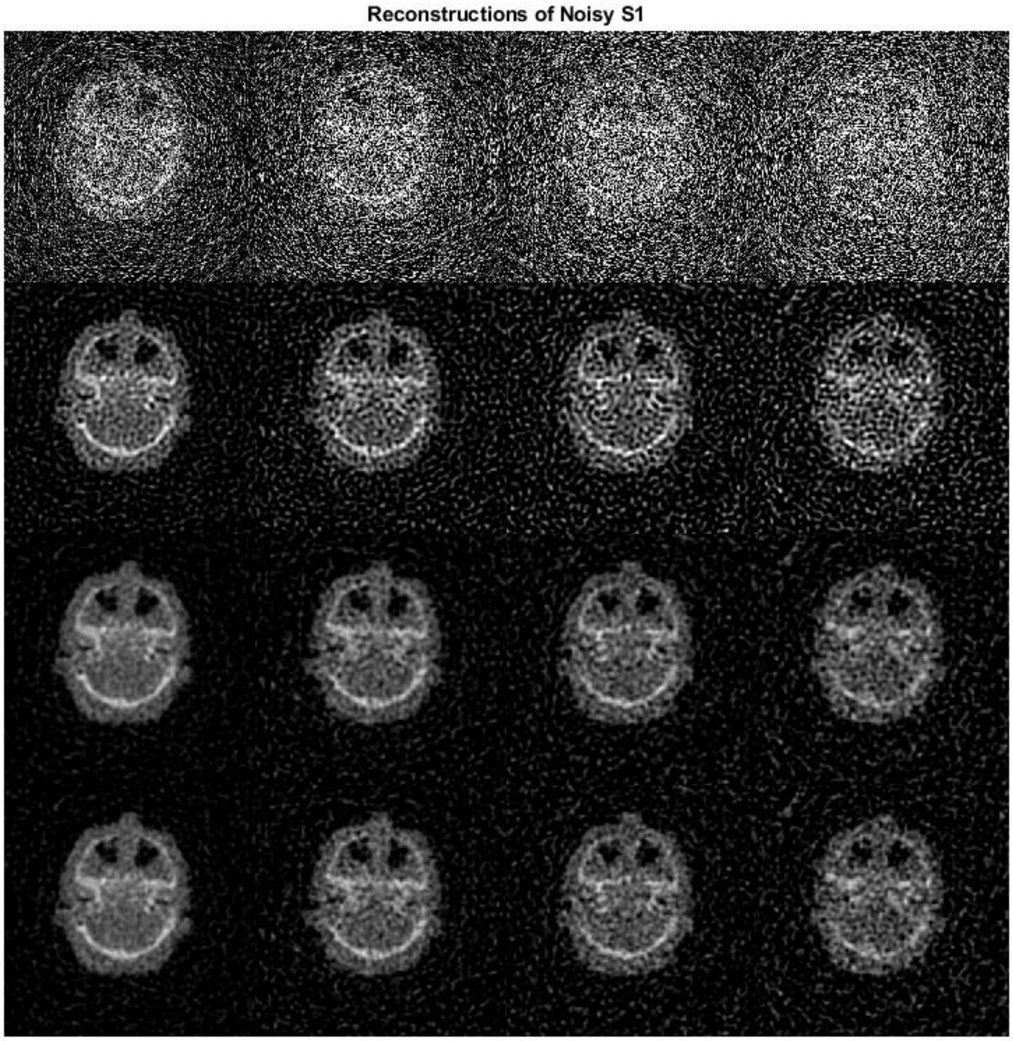


Figure 21: Noisy S1 reconstructions: (Row 1) DFI, (Row 2) FBP with Ram-Lak filter, (Row 3) FBP with Hanning filter,(Row 4) FBP with Shepp-Logan filter.

Conclusions

Both methods created recognizable reconstructions of the first test image. When comparing FBP and DFI with a test image, DFI appeared to have smaller errors than FBP. However, DFI produced regions with non zero attenuation outside of the support of the original image. So even though it appeared to be more accurate, it induced attenuation in places we expected no attenuation. FBP seemed to produce the best results with the Shepp-Logan filter. With a large enough B parameter on well behaved data, all of the filters produced visually clean images. For smaller B values, each filter produced blurry images, and in the Ram-Lak filter case had aliasing issues. Even with very large B values, the FBP methods smoothed the edges of features in the images. Overall the Shepp-Logan FBP reconstructions had the sharpest features, and were more resistant to noisy sinogram data. When dealing with noise, DFI failed miserably. Overall the Shepp-Logan FBP scheme performed the best.

Appendix I: Code

FBP: filteredBackProj.m

```
function [f] = filteredBackProj(S,theta,t,filter,B)
[n,~]   = size(t);
[m,~]   = size(theta);
N       = (n-1)/2;
L       = max(t);
r       = (-N:N)*pi/L;

%% Compute phi_hat (the DFT of the apodizing function) for designated filter
if strcmp(filter,'Ram-Lak')
    phi_hat = RamLak(ifftshift(r),B)';
elseif strcmp(filter,'Hanning')
    phi_hat = Hanning(ifftshift(r),B)';
elseif strcmp(filter,'Shepp-Logan')
    phi_hat = SheppLogan(ifftshift(r),B)';
elseif strcmp(filter,'None')
    phi_hat = ones(n,1);
else
    error('Invalid filter selction. Choose from: "Ram-Lak", "Hanning", "Shepp-Logan", or "None"')
end

%% Convolving with filter
S_shifted   = ifftshift(S);         %   Shift the domain to [0,2L) for use
                                    %   with fft function (by periodic-ness)

RF_tilde    = fft(S_shifted);       %  Compute the DFT of the radon transform

conv        = RF_tilde.*phi_hat;    %   Convolution of RF and phi is equivalent
                                    %   to this element-wise multiplication of
                                    %   the DFTs.

S_filtered  = fftshift(ifft(conv,'symmetric'));% Computing the filtered sinogram
f = backProj(S_filtered,theta,t, filter);   %   Computing the backprojection of the
                                    %   filtered data to get the attenution

                                    %   coefficient, f.
f(f<0) = 0;
%% Displaying data

% figure(1);
% subplot(3,1,1);plot(t,S(:,1)); title('Initial RF value for \Theta_0')
% subplot(3,1,2);plot(r, fftshift(phi_hat)); title([filter ' filter'])
% subplot(3,1,3);plot(t,S_filtered(:,1)); title('Filtered RF for \Theta_0')

end
%% Filter Functions

function [phi_hat] = RamLak(r, B)
% Applies a Ram-Lak Filter to a vector of frequencies, r, with bandlimit B
phi_hat = zeros(size(r));
idx = find(abs(r)>B);
phi_hat = abs(r);
phi_hat(idx) = 0;

end

function [phi_hat] = Hanning(r,B)
```

```
% Applies a Hanning Filter to a vector of frequencies, r, with bandlimit B
phi_hat         = zeros(size(r));
idx             = find(abs(r)>B);
phi_hat         = abs(r).*(cos((pi.*r)/(2*B)).^2);
phi_hat(idx)    = 0;

end

function [phi_hat] = SheppLogan(r,B)
% Applies B-low pass Shepp-Logan filter over the frequency domain r
phi_hat         = zeros(size(r));
d = pi/B;
phi_hat         = abs(sinc(r*d/2)).^3.*abs(r);
end

%% Backprojection Function
function [A] = backProj(S, theta, t, filter)

% This function performs Back Projection on a sinogram and
% reconstructs the image of a two-dimensional region with non-constant
% attenuation coefficients.
%
% ----------INPUTS-------------
% S:        A M (rows) x N (columns) array corresponding to a sinogram with
%           N theta measurement
% theta:    An N X 1 array of the measurement angles (domain of sinogram),
%           [0,pi-e] **NOTE: the measurment angles are prependicular to the
%           incoming rays, so the computed A matrix will need to be rotated
%           90 degrees for the final image**
% t :       An M X 1 array of the distance values (range of the sinogram)
%
% ----------OUTPUTS -----------
% A:        M X M matrix of attenutation coefficients associated with the
%           two-dimensional region

% %%%%%%%%%%%%%%%%%%%%%%%%%%%% DESCRIPTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [1]   This function will apply the values of the radon transform data to
%       all entries of the matrix temp matrix corresponding to the line
%       used to compute the the transform values.
%
% [2]   To do this for all measurments, the temp matrix will be rotated
%       according to the values of the measurement angles.
%
% [3]   The temp matrix values will be accumulated to the output A
%
% [4]   To attain the attenuation coefficients at each discrete entry in
%       the region, the values of A are averaged over the N measurements
%
% [5]   Since our measurement angles in theta are perpendicular to the
%       incoming waves, the output A is rotated 90 degrees to match the
%       measured region

if strcmp(filter,'whateven')
temp    = S(:,1);
S = [S flipud(temp)];
theta = [theta; pi];
else
S       = [S flipud(S)];
theta   = [theta; theta+ pi];
end
[M,N]   = size(S);
A_old   = zeros(M);
```

```
A       = zeros(M);
%for j = 1:2
% figure(2)
for i = 1:N
    temp = repmat(S(:,i),[1,M]);                     %[1]
    temp = imrotate(temp,theta(i)*(180/pi)+90,'bilinear','crop'); %[2] [5]
%     subplot(1,2,1);imagesc(flipud(A_old));colormap('gray');
%     axis('equal'); axis off;
%     subplot(1,2,2);spy(temp);
%     drawnow;

    A       = A_old + temp;                                 %[3]
    if i > 1 % zero out unaffected entries
        A(A==A_old) = 0;
        A(A==temp)  = 0;
    end
    A_old = A;
end
A   = (1/(2*N))*A;   %[4]
A   = flipud(A);
% Displaying the results


% a = num2str(t(1)); b = num2str(t(end));
% figure(3);
% imagesc(f;colormap('gray');axis('equal')
% axis off;
% title(['Attenuation Coefficients over [' a ' , ' b ']\times[' a ' , ' b ']']);
% figure(4);
% subplot(1,2,1);surf(t,t,f,'edgecolor','none');
% subplot(1,2,2);surf(t
end
```

DFI: dirFourierInv.m

```
function [f] = dirFourierInv(R, theta, t)
%{
Inverse Radon Transform by Direct Fourier Inversion

R: Sinogram data on meshgrid(t,theta)
t: vector of shift values from discrete radon transform
theta: vector of angle values from discrete radon transform

F: function approximations over meshgrid(x,y)

Steps:
1) 1-D Fourier Coef. R
2) Convert from Polar to Cartesian Coordinates
2.5) Enforce symetry condition?
3) Inverse 2-D Fourier Transform
%}

%initial stuff

if mod(size(R,1),2) == 1
   R = R(1:end-1,:);%using left Reiman sum, so the largest t val row of R wont be used
end
L = (max(t) - min(t))/2;
[N,M] = size(R);
N = (N)/2;

%Step 1
```

```

% Find fft of sinogram in polar coords:
%1) shift DC for t to (1,1)
%2) fft-1D
%3) shift DC for r to center
%4) scale by timestep

F_pol = L*fftshift(fft(ifftshift(R,1)),1)/N;
r = -N:N;
r = r*pi/L;

%Step 2
[theta_m, r_m] = meshgrid(theta,r);
[cartx, carty] = pol2cart(theta_m, r_m);
F_pol = [F_pol ; conj(F_pol(1,:))]; %extend F_pol to include its r_n = Npi/L row

[x,y] = meshgrid(r,r);

F_cart = griddata( carty, cartx , F_pol, x, y, 'cubic');

dummy = zeros(2*N+1);%replace nans from griddata with zeros
b = isnan(F_cart);
F_cart(b) = dummy(b);

%Step 2.5
F_cart = (F_cart + conj(rot90(F_cart,2)))/2;%impose forier symmetry
F_cart = ifftshift(F_cart); %shift DC of r back to (1,1)


%Step 3
%ifft
f = N*N/(L*L)*ifft2(F_cart, 'symmetric');
f = fftshift(f); %undo fftshift
f = rot90(f,-1);%fix weird rotation issues
f = fliplr(f);

end
```

References

[1] Epstein, Charles L. (2001) *The Mathematics of Medial Imaging*. Philadelphia, Pennsylvania: University of Pennsylvania

[2] Lee, C. (2018) Lectures and Outlines on Tomography