**Computed Tomography Notes, Part 1**

**Challenges with Projection X-ray Systems**

The equation that governs the image intensity in projection imaging is:

\[
I_o(x, y) = I_0 \exp\left(-\int \mu(x, y, z) dz\right)
\]

Projection x-ray systems are the most inexpensive and widespread medical imaging device, but there are some major drawbacks:

- There is no depth (z) information in the images – we can’t tell where along a particular line where a lesion is located.
- Lack of contrast – large changes in attenuation coefficient may results in very small changes in image intensity:

![Diagram](image)

\[
\text{Contrast in } \mathcal{M}: \frac{I - I_0}{I_0} = \frac{e^{-1} - e^{-11}}{e^{-11}} = \frac{0.03}{0.3} = 0.1
\]

**Solution: Computed Tomography** (see Macovski, pp. 113-141)

Definition: Tomography is the generation of cross sectional images of anatomy or structure.

First, we will reduce the dimensionality of the problem through collimation of the x-ray source to a single slice through the object (choose a single z location):
The intensity along this 1D row of detectors is now:
\[ I_d(y) = I_0 \exp\left(-\int \mu(x,y) dx \right) \]

We now define a new function:
\[ g(y) = \ln \frac{I_0}{I_d(y)} = \int \mu(x,y) dx = \int f(x,y) dx \]

where \( g \) is the “projection” through some unknown function \( f \) that we wish to determine. We can also describe \( g \) as the “line integral” through \( f \) in the \( x \) direction:

Another way of writing the line integral is:
\[ g(R) = \iint f(x, y) \delta(y - R) dxdy = \int f(x, R) dx \]

where \( y = R \) defines a line along with the integration is to occur (this is the only place where the delta function is non-zero).

We can now describe the line integral at an arbitrary angle, \( \theta \):
with the following expression:

\[ g_{\theta}(R) = \iint f(x, y) \delta(x \cos \theta + y \sin \theta - R) \, dx \, dy \]

This collection of projections \( g_{\theta}(R) \) is known as the \textit{Radon transform} of \( f(x,y) \).

**The Central Section Theorem (projection-slice theorem)**

Perhaps the most important theorem in computed tomography is the central section theorem, which says:

The 1D FT of a projection \( g_{\theta}(R) \) is the 2D FT of \( f(x,y) \) evaluated at angle \( \theta \).

Taking the 1D FT of the projection, we get:

\[
G_{\theta} (\rho) = F_{1D(R)} \{ g_{\theta}(R) \} = \iint f(x, y) \delta(x \cos \theta + y \sin \theta - R) \exp(-i 2\pi \rho R) \, dx \, dy \, dR
\]

\[
= \iiint f(x, y) \exp(-i 2\pi \rho (x \cos \theta + y \sin \theta)) \, dx \, dy \\
= \iiint f(x, y) \exp(-i 2\pi (\rho \cos \theta x + \rho \sin \theta y)) \, dx \, dy
\]

Observing that the 2D FT of \( f(x,y) \) is:

\[ F(u, v) = \iint f(x, y) \exp(-i 2\pi (ux + vy)) \, dx \, dy \]

and that \((u,v)\) in polar coordinates is \((\rho \cos \theta, \rho \sin \theta)\), we can see that:

\[
G_{\theta} (\rho) = F(u, v) \bigg|_{u=\rho \cos \theta, v=\rho \sin \theta}
\]

\[ = F(\rho, \theta) \]
To make an image, then, we can acquire projections at many different angles over \((0, \pi)\) to fill in the \(F(u,v)\) space and then inverse 2D FT to get the input image \(f(x,y)\):

\[
f(x, y) = \iiint F(u, v) \exp(i2\pi (ux + vy)) du dv
\]

\[
= \int_0^{2\pi} \int_0^{\infty} G_\theta(\rho) \exp(i2\pi (\rho \cos \theta x + \rho \sin \theta y)) \rho d\rho d\theta
\]

**Example 1**

Suppose we have an object that has the same projection at all angles:

\[
g_\theta (R) = 2 \text{sinc} (2R)
\]

\[
F(\rho, \theta) = F_{1D} \{2 \text{sinc} (2R)\} = \text{rect}(\rho / 2)
\]

We can see that that:

\[
F(\rho) = F(\rho, \theta) = \text{circ} (\rho)
\]

and therefore,

\[
f(r) = \text{jinc} (r) = \frac{J_1(2\pi r)}{r}
\]
Thus, if we project through a circularly symmetric jinc function, we will get a sinc function:

Example 2

We can also use the central section theorem to determine projections through a known object. For example, suppose we wanted to know the projection through 
$f(x,y) = \text{rect}(x)\text{rect}(y)$ at and angle of $\theta = \pi/4$.

$F(u,v) = \text{sinc}(u)\text{sinc}(v)$

$G_\theta(\rho) = \text{sinc}(\rho \cos \theta)\text{sinc}(\rho \sin \theta)$

$= \text{sinc}(\rho / \sqrt{2})\text{sinc}(\rho / \sqrt{2}) = \text{sinc}^2(\rho / \sqrt{2})$

$g_\theta(r) = F_{1D}^{-1}\left\{\text{sinc}^2(\rho / \sqrt{2})\right\} = \sqrt{2} \text{ tri}(\sqrt{2}r)$
Sinograms

In general, we have data for \( g_\theta(R) \) for many different angles \( \theta \) that can be placed into a big matrix that we call a “sinogram.” For example, let’s take some point object at \((x_0, y_0)\), e.g. \( \delta(x-x_0, y-y_0) \), then:

\[
g_\theta(R) = \int \int \delta(x-x_0) \delta(y-y_0) \delta(x \cos \theta + y \sin \theta - R) \, dx \, dy
= \delta(x_0 \cos \theta + y_0 \sin \theta - R)
\]

and letting \( y_0 = 0 \), then:

\[
g_\theta(R) = \delta(x_0 \cos \theta - R)
\]

a delta function located at \( x_0 \cos \theta \). That is, a point traces out a sinusoid in the \( R-\theta \) space and thus the name sinogram. This is also known as Radon space.

For a more complex object…. 
In the sinogram, the maximum deviation describes an object’s distance from the origin and the point of peak deviation describes the angular location of object. The three objects above, from smallest to largest are located at \((r, \theta) = (0,0), (113, \pi/4), \text{and} (-200,0) \text{ or } (200,\pi)\).

Finally, observe the symmetry: \( p(r, \theta) = p(-r, \theta + \pi) \).
Methods for Image Reconstruction

Image reconstruction from a set of projections is a classic inverse problem. This a particularly rich problem in that there are many different ways to approach this problem and we will present several of these below.

1. Direct Fourier Interpolation Method

This method makes direct use of the central section theorem. The steps in the image reconstruction are:

1. 1D FT each of the projections: \( F_{1D} \{ g_\theta (R) \} = G_\theta (\rho ) = F(\theta , \rho ) \)

2. Interpolate \( F(\rho , \theta ) \) to \( \hat{F}(u, v) \) (polar to rectangular coordinates – e.g. you could use Matlab functions interp2 or griddata)

3. Inverse 2D FT

2. Backprojection-Filtering

Backprojection means that we “smear” the projection data back across the object space.

The backprojection operator for a single projection looks like this:

\[
b_\theta (x, y) = \int_{-\infty}^{\infty} g_\theta (R) \delta (x \cos \theta + y \sin \theta - R) dR
\]
and the total backprojected image is the integral (sum) of this over all angles:

\[ f_b(x, y) = \int_0^\pi b_b(x, y) d\theta \]

\[ = \int_0^\infty \int_0^\infty g_\theta(R) \delta(x \cos \theta + y \sin \theta - R) Rd\theta d\theta \]

Now consider that:

\[ g_\theta(R) = F_{1D, \rho}^{-1} \{ F(\rho, \theta) \} = \int_0^\infty F(\rho, \theta) \exp(i2\pi \rho R) d\rho \]

then the backprojected image can be written as:

\[ f_b(x, y) = \int_0^\infty \int_0^\infty \int_0^\infty \int_0^\infty F(\rho, \theta) \delta(x \cos \theta + y \sin \theta - R) \exp(i2\pi \rho R) dR d\theta d\rho d\phi \]

\[ = \int_0^\pi \int_0^\infty F(\rho, \theta) \exp(i2\pi \rho (x \cos \theta + y \sin \theta)) d\theta d\rho \]

This is nearly the inverse FT formulation of in polar coordinates, but two changes are needed:

a. limits of integration should be (0, 2\pi) and (0, \infty), and

b. we need \( rdrd\theta \) for an integration in polar coordinates

We can address the first issue by recognizing that \( F(-\rho, \theta) = F(\rho, \theta+\pi) \) and we can address the second issue multiplying and dividing by \( \rho \).

\[ f_b(x, y) = \int_0^{2\pi} \int_0^\infty \frac{F(\rho, \theta)}{\rho} \exp(i2\pi \rho (x \cos \theta + y \sin \theta)) \rho d\rho d\theta \]

\[ = F_{2D}^{-1} \left\{ \frac{F(\rho, \theta)}{\rho} \right\} \]

\[ = f(x, y) * * F_{2D}^{-1} \left\{ \frac{1}{\rho} \right\} \]

\[ = f(x, y) * * \frac{1}{r} \]

This says that the backprojected image is equal to the desired image convolved with a 1/r blurring function.
Up until this point, we’ve only done the backprojection. In order to get the final image, we need to undo this blurring function. Thus, the steps in backprojection-filter method are

1. Backproject all projections, $g_{\theta}(R)$ to get $f_{b}(x,y)$.

2. Forward 2D FT to get $\frac{F(\rho,\theta)}{\rho}$.
3. Filter with $\rho$ (or $|\rho|$) to get $F(\rho, \theta)$. This is a “cone”-like weighting, $H(u, v) = \sqrt{u^2 + v^2}$, applied to $F(u, v)$.

![Cone Filter Diagram]

4. Inverse 2D FT to get $\tilde{f}(x, y)$.

One disadvantage to this method is that it is often necessary to backproject across an extended matrix because the blurred image extends beyond the original object due to the long extent of the $1/r$ function. In addition, the deblurring filtering done in the Fourier domain will lead to artifact if the object isn’t padded out.

3. Direct Fourier Superposition and Filtering Method

This method makes use of the fact that backprojection is mathematically equivalent to adding a line to the Fourier data. We show this by examining looking at the backprojection operator in a rotated coordinate system where:

$$x_r = x \cos \theta + y \sin \theta$$
$$y_r = -x \sin \theta + y \cos \theta$$

![Rotated Object and Fourier Spaces]

The backprojection operator is:
\[ b_\theta(x, y) = \int_{-\infty}^{\infty} g_\theta(R) \delta(x \cos \theta + y \sin \theta - R) dR \]
\[ b_\theta(x_r, y_r) = \int_{-\infty}^{\infty} g_\theta(R) \delta(x_r - R) dR \]
\[ = g_\theta(x_r) \cdot 1 \]

The 2D FT in the rotated frame is:
\[ B_\theta(u_r, v_r) = G_\theta(u_r) \delta(v_r) \]

and substituting back to the standard coordinate system, we get:
\[ B_\theta(u, v) = G_\theta(u \cos \theta + v \sin \theta) \delta(-u \sin \theta + v \cos \theta) \]

This is the FT of the projection, \( G_\theta(\rho) \), positioned along a delta line at angle \( \theta \). The steps in this method are then:
1. 1D FT each projection to get \( G_\theta(\rho) \).
2. Place data directly into Fourier matrix (add or superimpose each \( G_\theta(\rho) \)).
3. Filter with \( \rho \) (or \( |\rho| \)) to get \( F(\rho, \theta) \).
4. Inverse 2D FT to get \( \tilde{f}(x, y) \).

4. Filtered Backprojection Method

In this method, we reverse the order of backprojection and filtering. The steps are:
1. Filter the backprojection with a \( |\rho| \) filter. This is sometimes called a “ramp” filter.
   a. Fourier method:
      \[ g_\theta'(R) = F_{1D}^{-1}\left\{ \mathcal{F}\left\{ g_\theta(R) \right\} \right\} \]
   b. Convolution method
      \[ g_\theta'(R) = g_\theta(R) * c(R) \]
      where \( c(R) \approx F_{1D}\left\{ \rho \right\} \)
2. Backproject for all angles to get \( \tilde{f}(x, y) \).

Putting it all together, we get:
\[
\hat{f}(x, y) = \int_0^\infty \int_0^\infty f_{1D}^{-1}(\rho) F_{1D} \{ g_\theta (R) \} \delta (x \cos \theta + y \sin \theta - R) dR d\theta \\
= \int_0^\infty \int_0^\infty F_{1D}^{-1}(\rho) F(\rho, \theta) \delta (x \cos \theta + y \sin \theta - R) dR d\theta \\
= \int_0^\infty \int_0^\infty |\rho| F(\rho, \theta) \exp(i2\pi \rho R) \delta (x \cos \theta + y \sin \theta - R) d\rho dR d\theta \\
= \int_0^\infty \int_0^\infty F(\rho, \theta) \exp(i2\pi \rho (x \cos \theta + y \sin \theta)) |\rho| d\rho d\theta \\
\]

and changing the limits of integration to \((0, 2\pi)\) and \((0, \infty)\), we get:

\[
\hat{f}(x, y) = \int_0^{2\pi} \int_0^\infty F(\rho, \theta) \exp(i2\pi \rho (x \cos \theta + y \sin \theta)) \rho d\rho d\theta \\
= F_{2D}^{-1}\{ F(\rho, \theta) \} = f(x, y)
\]
One example:

And then backprojecting…

Now, let’s look at the convolution function, \( c(R) \).

\[
c(R) = F^{-1} \{ p \}
\]
does not exist. However, we can find the FT of a variety of functions that approach $|\rho|$ in the limit. For example:

$$c(R) = \lim_{\epsilon \to 0} F^{-1}\{ |\rho| \exp(-\epsilon|\rho|) \}$$

$$= \lim_{\epsilon \to 0} \frac{2(\epsilon^2 - 4\pi^2 R^2)}{(\epsilon^2 + 4\pi^2 R^2)^2}$$

Thus,
- for small $R$, $c(R)$ will approach $2/e^2$
- for large $R$, $c(R)$ will approach $-1/2\pi^2 R^2$

Above, the function $\exp(-\epsilon|\rho|)$ clips of the high spatial frequency parts of $|\rho|$, with an approximate cutoff frequency of $\rho_0$. There are numerous other functions that can do that. For example, we could clip off the high spatial frequencies using a rect function.
This filter has substantial ringing artifact. One can also apply a Gaussian or Hanning filter:

\[ C(\rho) = |\rho| \text{rect}\left(\frac{\rho}{2\rho_0}\right) \]

\[ = \rho_0 \left[ \text{rect}\left(\frac{\rho}{2\rho_0}\right) - \text{tr}\left(\frac{\rho}{\rho_0}\right) \right] \]

\[ c(R) = \rho_0^2 \left( 2\text{sinc}^2(2\rho_0 R) - \text{sinc}^2(\rho_0 R) \right) \]
The filter resulting from a Hanning apodized ramp filter is:

\[
C(\rho) = |\rho| \exp \left( -\pi \left( \frac{\rho}{\rho_0} \right)^2 \right)
\]

\[
C(\rho) = |\rho| \text{rect} \left( \frac{\rho}{2\rho_0} \right) \left( \frac{1}{2} + \frac{1}{2} \sin \left( \pi \frac{\rho}{\rho_0} \right) \right)
\]

5. Algebraic Reconstruction Technique (ART)

This is like regular backprojection, but this method uses iterative corrections. There are numerous variants, but I will discuss the simplest “additive” ART. Here, we let \( g_i \) be the measured projections and \( f_{ij}^q \) be the image at iteration \( q \). Then:

\[
f_{ij}^{q+1} = f_{ij}^q + \frac{g_i + \sum_j f_{ij}^q}{N}
\]
Here’s an example for a noise free object. Consider the object with 4 pixel values and 6 pieces of projection information. We initialize the data to all zeros (e.g. $f_1^0=0, f_2^0=0$, ...). Looking at the projections from top to bottom, we can find the pixel values at $q = 1$:

$$f_1^1 = 0 + \frac{11-0}{2} = 5.5; \quad f_3^1 = 0 + \frac{9-0}{2} = 4.5$$

$$f_2^1 = 0 + \frac{11-0}{2} = 5.5; \quad f_4^1 = 0 + \frac{9-0}{2} = 4.5$$

Now, looking at the left-right projections:

$$f_1^2 = 5.5 + \frac{12-(5.5+4.5)}{2} = 6.5; \quad f_3^2 = 4.5 + \frac{12-(5.5+4.5)}{2} = 5.5$$

$$f_2^2 = 5.5 + \frac{8-(5.5+4.5)}{2} = 4.5; \quad f_4^2 = 4.5 + \frac{8-(5.5+4.5)}{2} = 3.5$$

Now, we look at the diagonal projections:

$$f_1^3 = 6.5 + \frac{7-(6.5+3.5)}{2} = 5; \quad f_3^3 = 5.5 + \frac{13-(5.5+4.5)}{2} = 7$$

$$f_2^3 = 4.5 + \frac{13-(5.5+4.5)}{2} = 6; \quad f_4^3 = 3.5 + \frac{7-(6.5+3.5)}{2} = 2$$
which we can see is consistent with all of the projections. While this procedure may look fine, in the presence of noise (that is, the projections aren’t exactly consistent), this method converges very slowly, and in some cases, not at all. Usually, one goes through the entire set of projections multiple times.