Computed Tomography Notes, Part 1

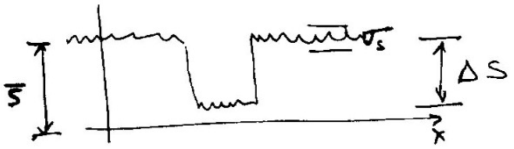
Challenges with Projection X-ray Systems

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

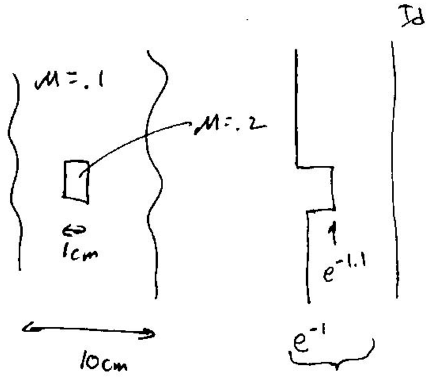
$$I_d(x, y) = I_0 \exp\left(-\int \mu(x, y, z)dz\right)$$

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. We define contrast as the change in intensity relative to the background intensity:



e.g. we can define a contrast $C = \Delta S / \bar{S}$ and a contrast to noise ratio $CNR = \Delta S / \sigma_s$. High contrast in the attenuation coefficient might yield low contrast in the projection image.

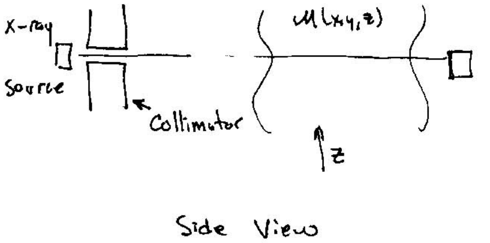


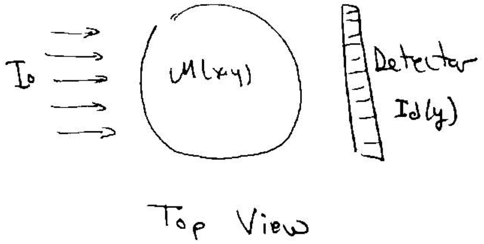
Contrast in μ $\quad \frac{.2 - .1}{.1} = \underline{\underline{1}}$ $\quad C = \frac{\Delta S}{S} = \frac{e^{-1} - e^{-1.1}}{e^{-1}} = \frac{0.03}{0.3} \approx \underline{\underline{0.1}}$

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):





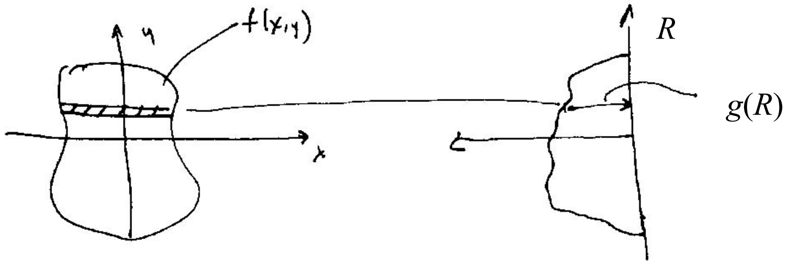
The intensity along this 1D row of detectors is now:

$$I_d(y) = I_0 \exp\left(-\int \mu(x,y)dx\right)$$

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):

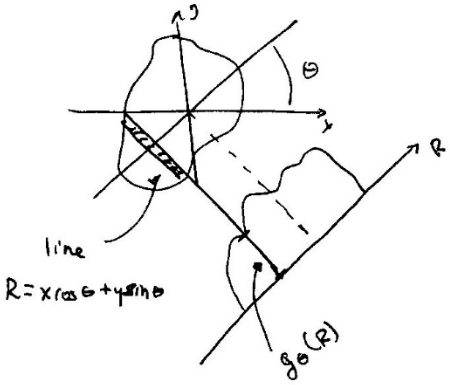


Another way of writing the line integral is:

$$g(R) = \iint f(x,y)\delta(y-R)dxdy = \int f(x,R)dx$$

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, θ:



with the following expression:

$$g_\theta(R) = \iint f(x,y)\delta(x\cos\theta + y\sin\theta - R)dxdy$$

This collection of projections $g_\theta(R)$ is known as the *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 θ.

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

$$\begin{aligned} G_\theta(\rho) &= F_{1D(R)}\{g_\theta(R)\} = \iiint f(x,y)\delta(x\cos\theta + y\sin\theta - R)\exp(-i2\pi\rho R)dxdydR \\ &= \iint f(x,y)\exp(-i2\pi\rho(x\cos\theta + y\sin\theta))dxdy \\ &= \iint f(x,y)\exp(-i2\pi(\rho\cos\theta x + \rho\sin\theta y))dxdy \end{aligned}$$

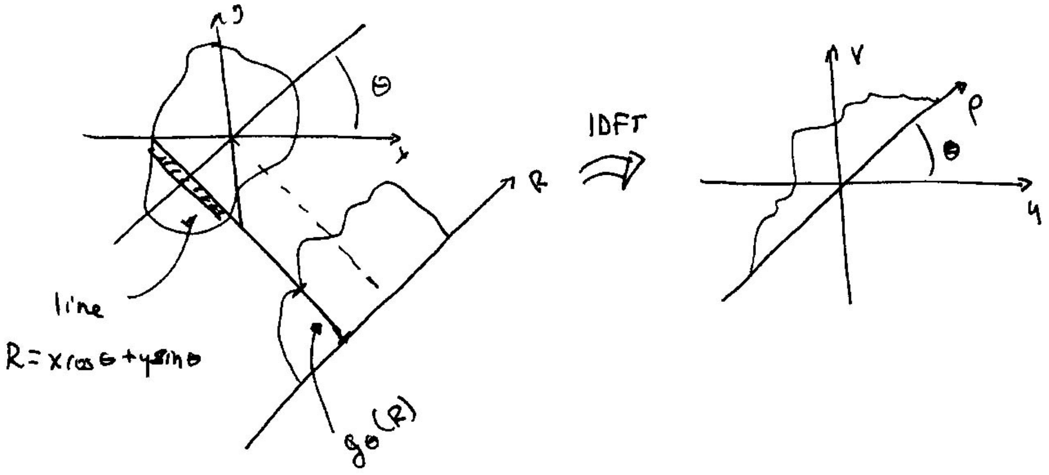
Observing that the 2D FT of $f(x,y)$ is:

$$F(u,v) = \iint f(x,y)\exp(-i2\pi(ux+vy))dxdy$$

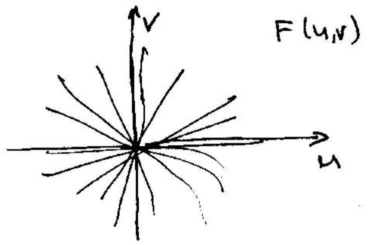
and that (u,v) in polar coordinates is $(\rho\cos\theta, \rho\sin\theta)$, we can see that:

$$G_\theta(\rho) = F(u,v)\Big|_{u=\rho\cos\theta, v=\rho\sin\theta}$$

$$= F(\rho,\theta)$$



To make an image, then, we can acquired 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 $f(x,y)$:



$$f(x,y) = \iint F(u,v)\exp(i2\pi(ux+vy))dudv$$

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

Example 1

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

$$g_\theta(R) = 2\,\mathrm{sinc}(2R)$$

$$F(\rho,\theta) = F_{1D}\{2\,\mathrm{sinc}(2R)\} = \mathrm{rect}(\rho/2)$$

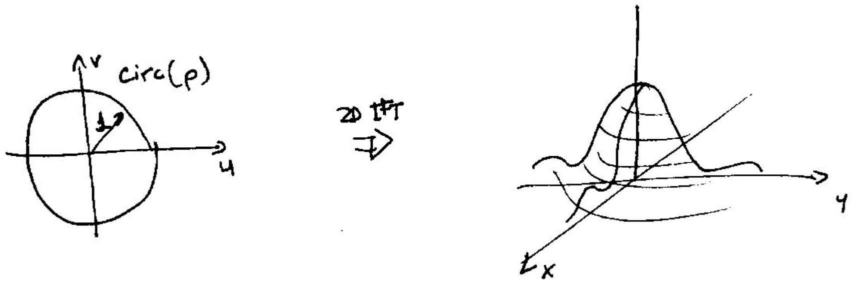
We can see that that:

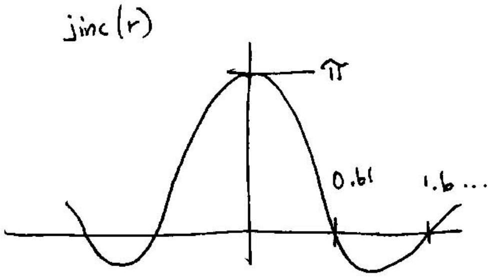
$$F(\rho) = F(\rho,\theta) = \mathrm{circ}(\rho)$$

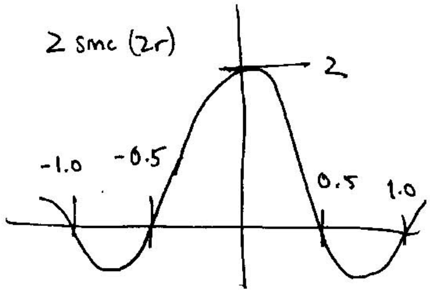
and therefore,

$$f(r) = 4\,\mathrm{jinc}(2r) = \frac{J_1(2\pi r)}{r}$$

Thus, if we project through a circularly symmetric jinc function, we will get a sinc function:







Example 2

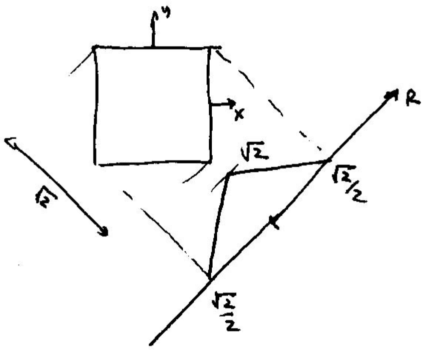
We can also use the central section theorem to determine projections through a known object. For example, suppose we wanted to know the projection through $f(x,y)$ = rect(x)rect(y) at and angle of $\theta = \pi/4$.

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

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

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

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



Sinograms

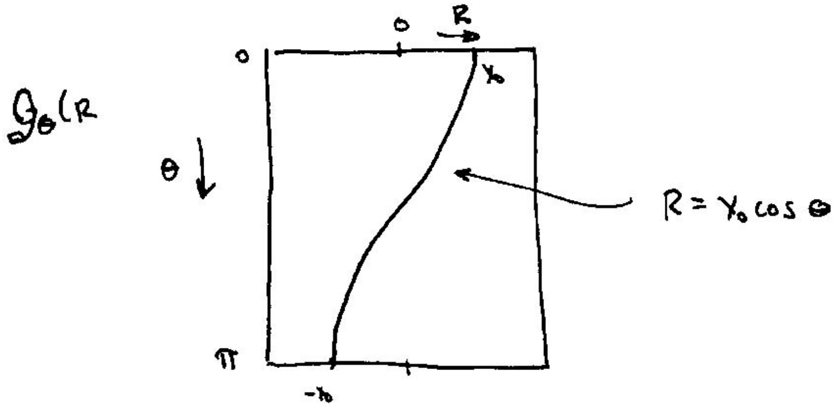
In general, we have data for $g_\theta(R)$ for many different angles θ that can be placed into a big matrix that we call a "sinogram." For example, let's take some point object at (x_0,y_0), e.g. $\delta(x-x_0,y-y_0)$, then:

$$g_\theta(R) = \iint \delta(x-x_0)\delta(y-y_0)\delta(x\cos\theta + y\sin\theta - R)dxdy$$
$$= \delta(x_0\cos\theta + y_0\sin\theta - R)$$

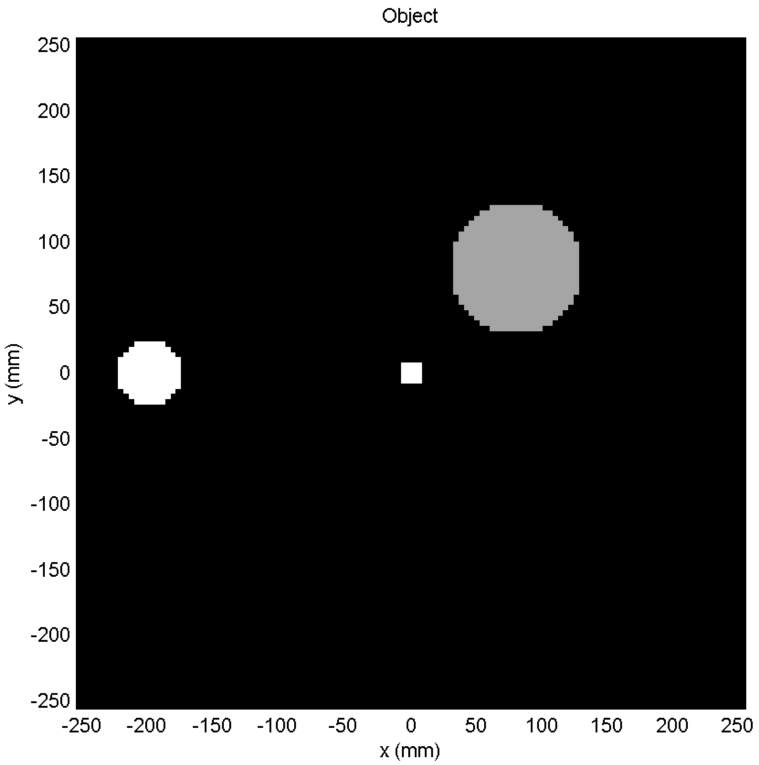
and letting $y_0 = 0$, then:

$$g_\theta(R) = \delta(x_0\cos\theta - R)$$

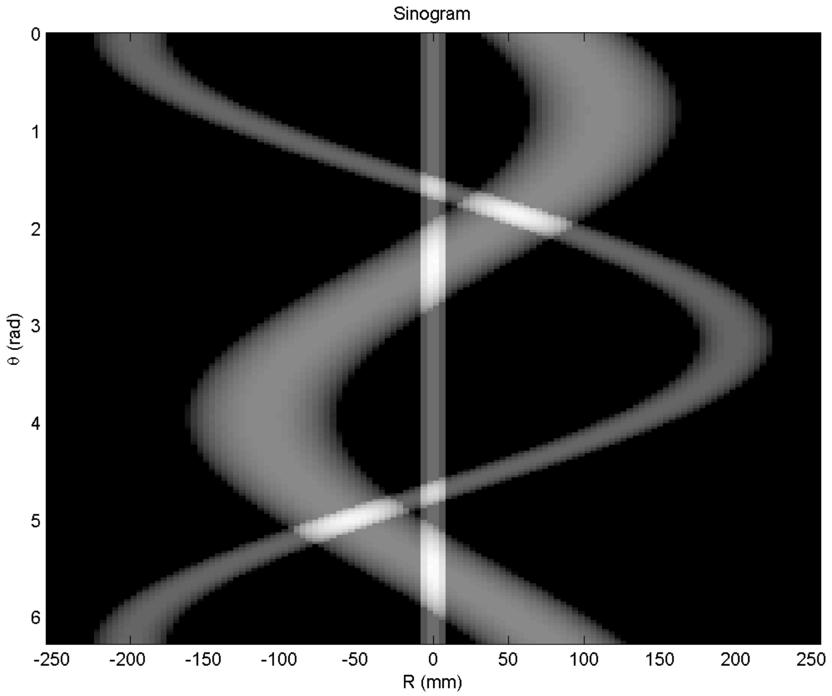
a delta function located at $x_0\cos\theta$. That is, a point traces out a sinusoid in the R-θ space and thus the name sinogram. This is also known as *Radon* space.



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 $(r,\theta) = (0,0)$, $(113, \pi/4)$, and $(-200,0)$ or $(200,\pi)$.



Finally, observe the symmetry: $p(r,\theta) = p(-r,\theta+\pi)$.

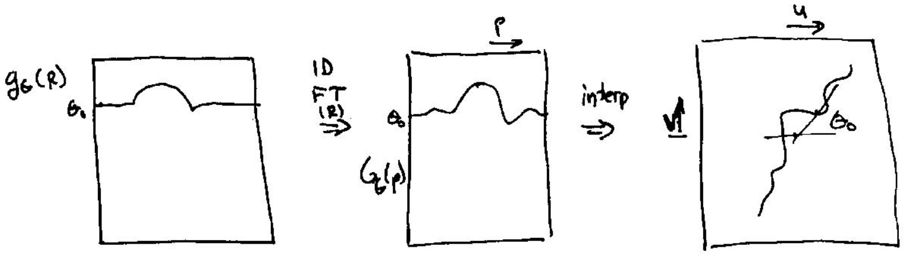
Methods for Image Reconstruction

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

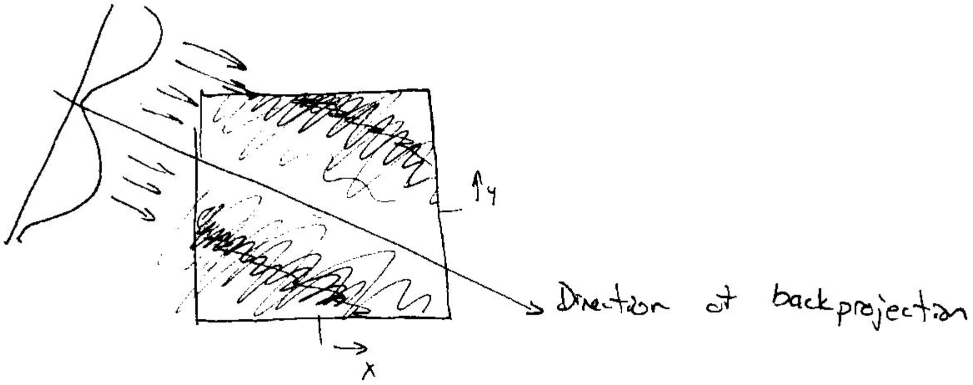
This method makes direct use of the central section theorem. The steps in the image reconstruction are:

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:

$$f_b(x,y) = \int_0^{\pi} b_\theta(x,y)d\theta$$
$$= \int_0^{\pi}\int_{-\infty}^{\infty} g_\theta(R)\delta(x\cos\theta + y\sin\theta - R)dRd\theta$$

Now consider that:

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

then the backprojected image can be written as:

$$f_b(x,y) = \int_{-\infty}^{\infty}\int_0^{\pi}\int_{-\infty}^{\infty} F(\rho,\theta)\delta(x\cos\theta + y\sin\theta - R)\exp(i2\pi\rho R)dRd\theta d\rho$$
$$= \int_0^{\pi}\int_{-\infty}^{\infty} F(\rho,\theta)\exp(i2\pi\rho(x\cos\theta + y\sin\theta))d\theta d\rho$$

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

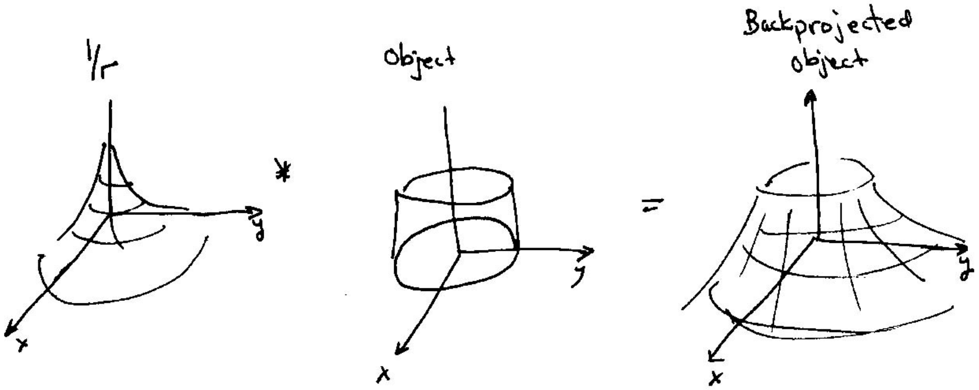
a. limits of integration should be (0, 2π) and (0, ∞), and

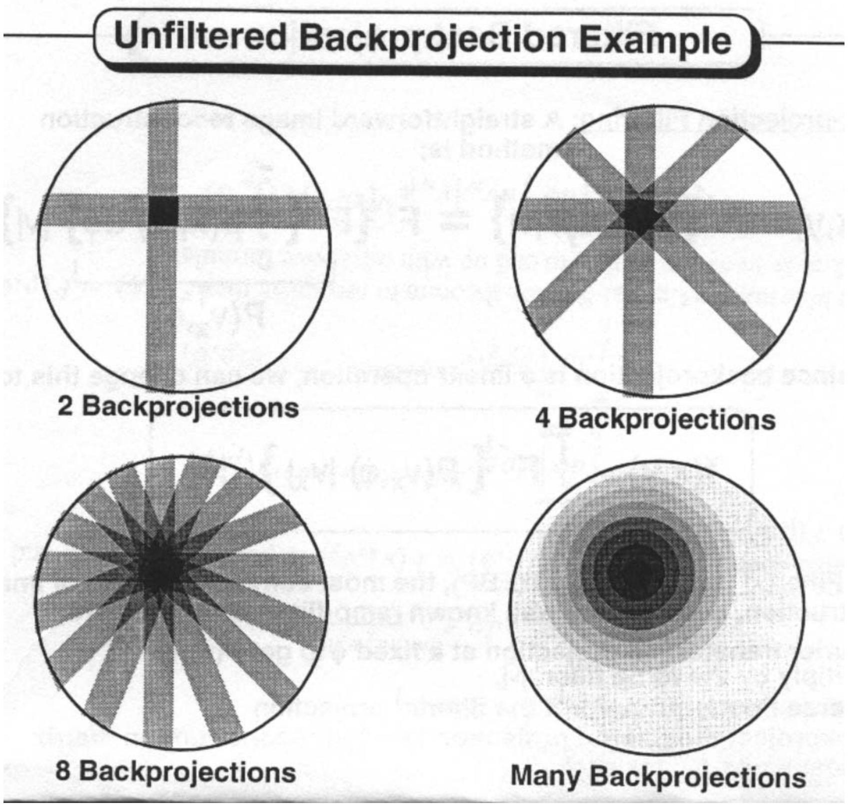
b. we need $\rho d\rho d\theta$ for an integration in polar coordinates

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 ρ.

$$f_b(x,y) = \int_0^{2\pi}\int_0^{\infty} \frac{F(\rho,\theta)}{\rho}\exp(i2\pi\rho(x\cos\theta + y\sin\theta))\rho d\rho d\theta$$
$$= F_{2D}^{-1}\left\{\frac{F(\rho,\theta)}{\rho}\right\}$$
$$= f(x,y) ** F_{2D}^{-1}\left\{\frac{1}{\rho}\right\}$$
$$= f(x,y) ** \frac{1}{r}$$

This says that the backprojected image is equal to the desired image convolved with a 1/*r* blurring function.

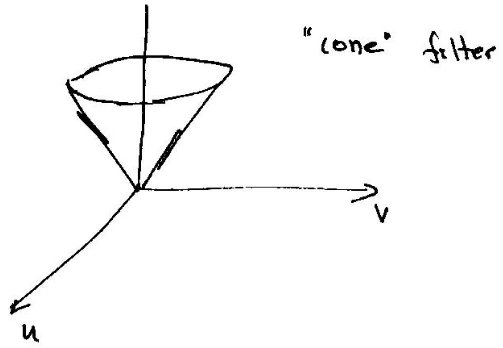




Up until this point, we've only done the backprojection. In order to get the final image, we need to undo this blurring function. Thus, the steps in backprojection-filter method are

1. Backproject all projections, $g_\theta(R)$ to get $f_b(x,y)$.
2. Forward 2D FT to get $\dfrac{F(\rho,\theta)}{\rho}$

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



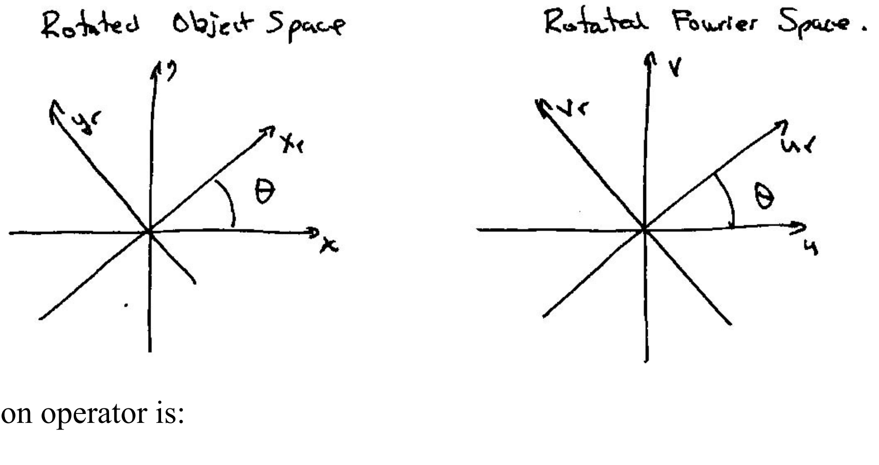
4. Inverse 2D FT to get $\widehat{f}(x, y)$.

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:

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



The backprojection operator is:

$$b_\theta(x,y) = \int_{-\infty}^{\infty} g_\theta(R)\delta(x\cos\theta + y\sin\theta - R)dR$$

$$\begin{aligned} b_\theta(x_r,y_r) &= \int_{-\infty}^{\infty} g_\theta(R)\delta(x_r - R)dR \\ &= g_\theta(x_r)\cdot 1 \end{aligned}$$

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 θ. 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 ρ (or $|\rho|$) to get $F(\rho, \theta)$.
4. Inverse 2D FT to get $\hat{f}(x,y)$.

4. Filtered Backprojection Method

In this method, we reverse the order of backprojection and filtering. The steps are:

1. Filter the projection with a $|\rho|$ filter. This is sometimes called a “ramp” filter.
    a. Fourier method:

    $$g'_\theta(R) = F_{1D}^{-1}\{|\rho| F_{1D}\{g_\theta(R)\}\}$$

    b. Convolution method

    $$g'_\theta(R) = g_\theta(R) * c(R)$$
    $$\text{where } c(R) \approx F_{1D}\{|\rho|\}$$

2. Backproject for all angles to get $\hat{f}(x,y)$.

Putting it all together, we get:

$$\hat{f}(x,y)=\int_0^{\pi}\int_{-\infty}^{\infty}F_{1D}^{-1}\left\{|\rho|F_{1D}\{g_\theta(R)\}\right\}\delta(x\cos\theta+y\sin\theta-R)\,dR\,d\theta$$

$$=\int_0^{\pi}\int_{-\infty}^{\infty}F_{1D}^{-1}\left\{|\rho|F(\rho,\theta)\right\}\delta(x\cos\theta+y\sin\theta-R)\,dR\,d\theta$$

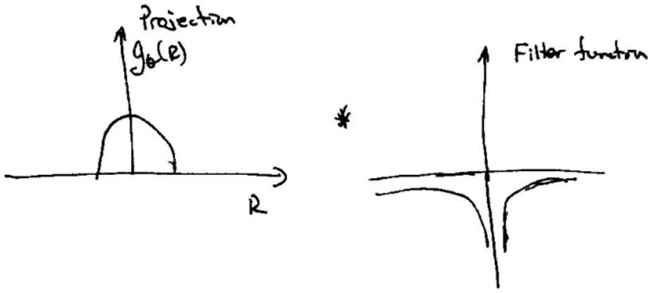
$$=\int_{-\infty}^{\infty}\int_0^{\pi}\int_{-\infty}^{\infty}|\rho|F(\rho,\theta)\exp(i2\pi\rho R)\delta(x\cos\theta+y\sin\theta-R)\,d\rho\,dR\,d\theta$$

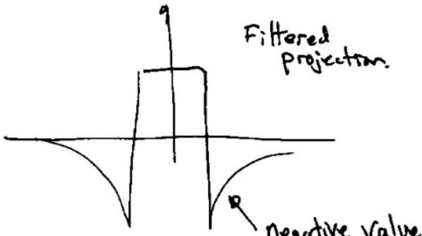
$$=\int_0^{\pi}\int_{-\infty}^{\infty}F(\rho,\theta)\exp(i2\pi\rho(x\cos\theta+y\sin\theta))|\rho|\,d\rho\,d\theta$$

and changing the limits of integration to (0, 2π) and (0, ∞), we get:

$$\hat{f}(x,y)=\int_0^{2\pi}\int_0^{\infty}F(\rho,\theta)\exp(i2\pi\rho(x\cos\theta+y\sin\theta))\rho\,d\rho\,d\theta$$

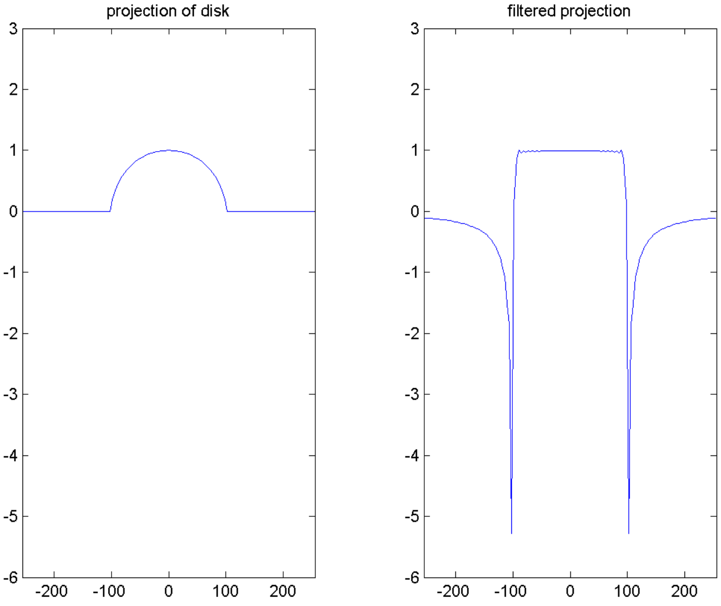
$$=F_{2D}^{-1}\{F(\rho,\theta)\}=f(x,y)$$



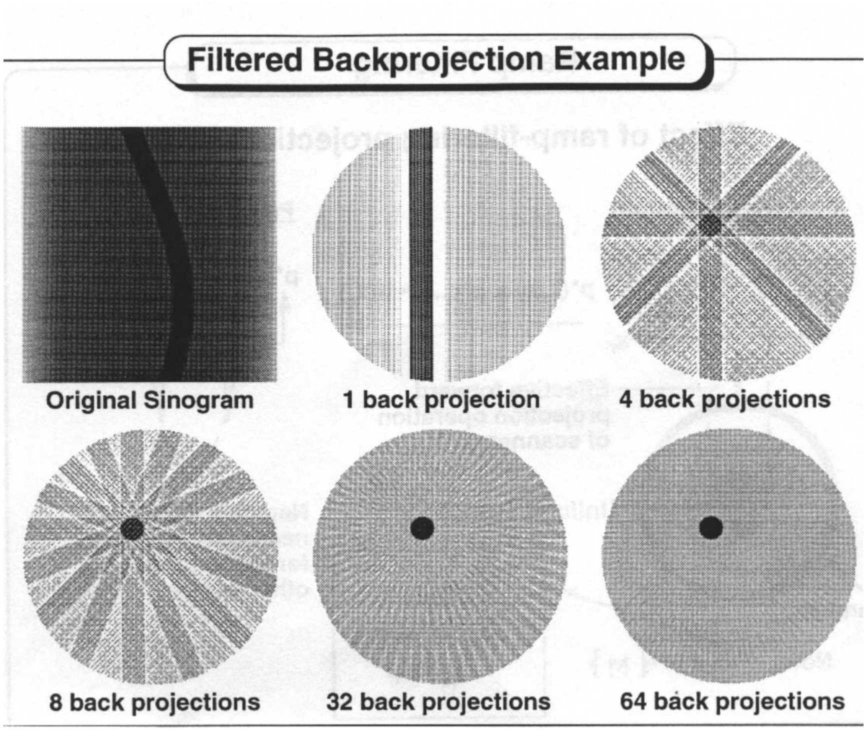


negative values are necessary to correct for projections from other angles

One example:



And then backprojecting…



Now, let's look at the convolution function, $c(R)$.

$$c(R) = F^{-1}\{|\rho|\}$$

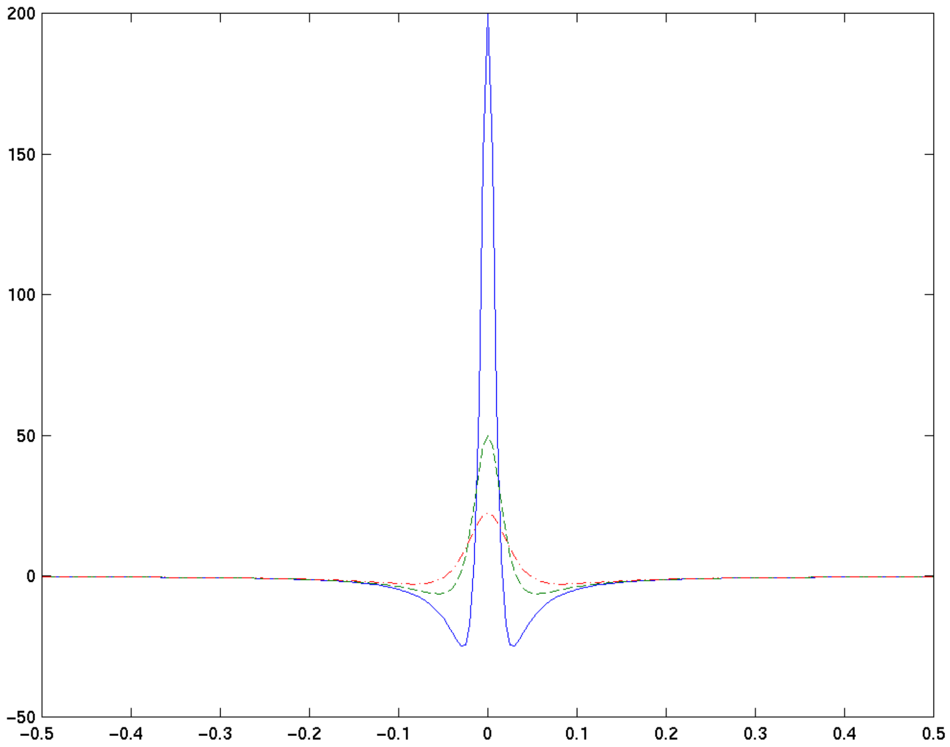
does not exist. However, we can find the FT of a variety of functions that approach $|\rho|$ in the limit. For example:

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

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

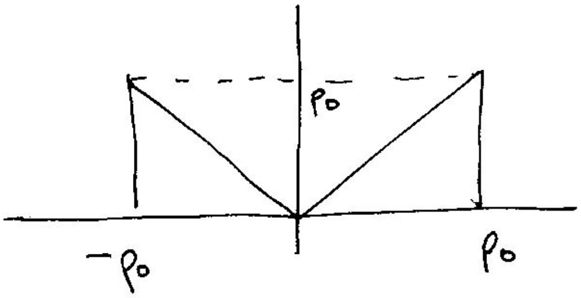
Thus,

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

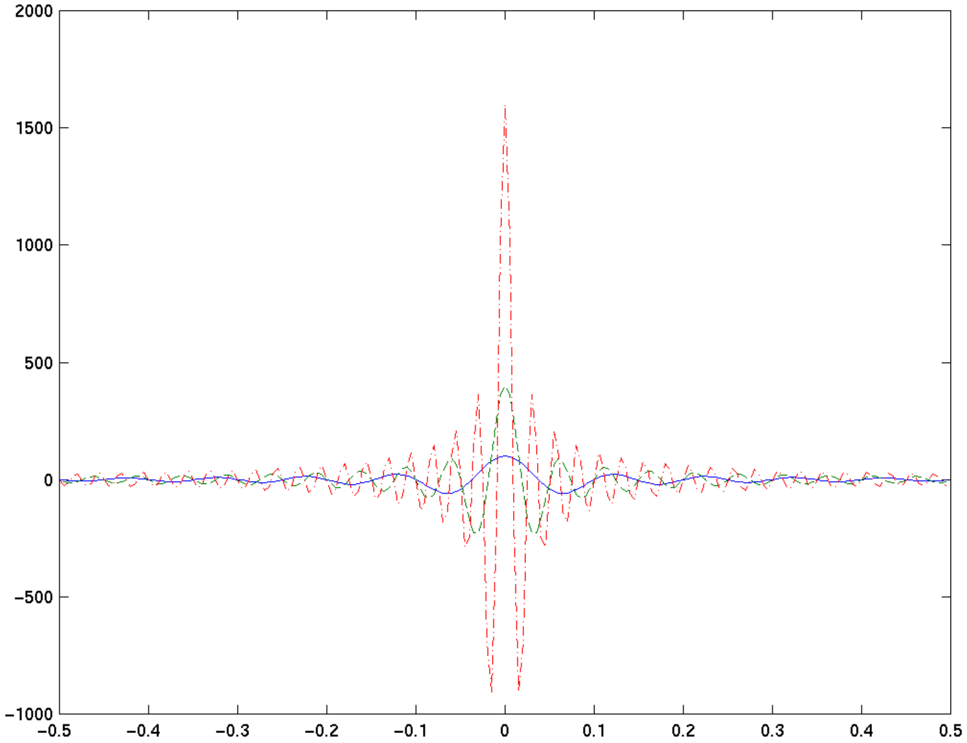


Above, the function $\exp(-\varepsilon|\rho|)$ clips of the high spatial frequency parts of $|\rho|$, with an approximate cutoff frequency of ρ_0. There are numerous other functions that can do that. For example, we could clip off the high spatial frequencies using a rect function.

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



$$c(R) = \rho_0^2\left(2\text{sinc}(2\rho_0 R) - \text{sinc}^2(\rho_0 R)\right)$$

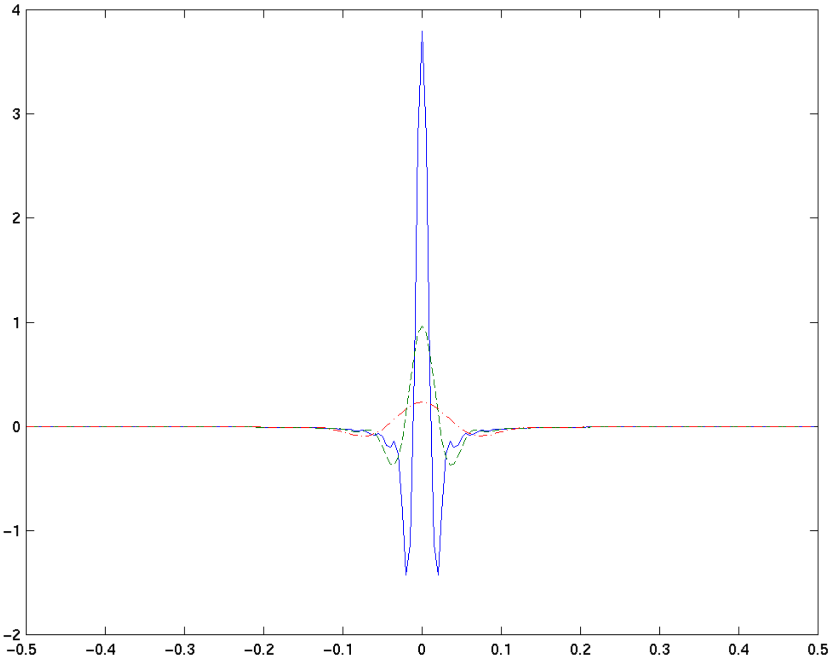


This filter has substantial ringing artifact. One can also apply a Gaussian or Hanning filter:

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

$$(\rho) = |\rho| \operatorname{rect}\left(\frac{\rho}{2\rho_0}\right)\left(\frac{1}{2} + \frac{1}{2}\cos\left(\pi\frac{\rho}{\rho_0}\right)\right)$$

Filters resulting from a Hanning apodized ramp filters are:



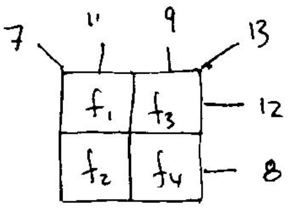
5. Algebraic Reconstruction Technique (ART)

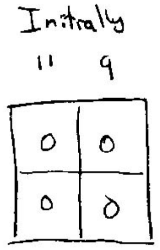
This is like regular backprojection, but this method uses iterative corrections. There are numerous variants, but I will discuss the simplest "additive" ART. Here, we let g_i be the measured projections (i is the projection number and g_i is a vector) and f_j^q be the estimated image at iteration *q* (*j* indexes all pixels in the image). Then:

$$f_j^{q+1} = f_j^q + \lambda\frac{g_i - \sum_j w_{ij} f_j^q}{\sum_j w_{ij}}$$

The weights, w_{ij}, give the contribution of a pixel j to g_i. The term $\sum_j w_{ij} f_j^q$ represents $\hat{g}_i^q$, our best guess of the projection i at iteration q, and then the numerator represents a correction term to move the solution closer to the an image that is consistent with the measured projections. λ is a regularization term to stabilize the algorithm.

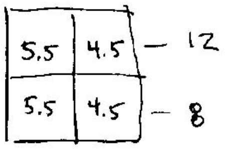
Here's an example for a noise free object. Consider the object with 4 pixel values and 3 projections – vertical, horizontal and the diagonals – with 6 pieces of projection information (over-determined). We initialize the data to all zeros (e.g. $f_1^0=0, f_2^0=0, \ldots$). In this simplified example, $\lambda = 1$ and $w_{ij} = 1$. Looking at the vertical projections, we can find the pixel values at iteration $q = 1$:





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

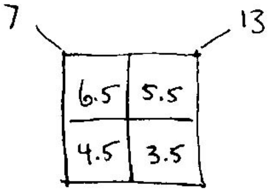
$$f_2^1 = 0 + \frac{11-0}{2} = 5.5; \quad f_4^1 = 0 + \frac{9-0}{2} = 4.5$$



Now, looking at the left-right projections:

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

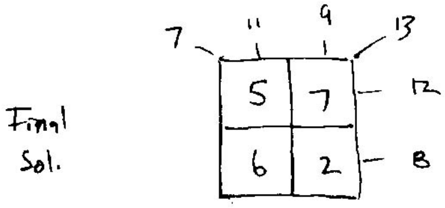
$$f_2^2 = 5.5 + \frac{8-(5.5+4.5)}{2} = 4.5; \quad f_4^2 = 4.5 + \frac{8-(5.5+4.5)}{2} = 3.5$$



Now, we look at the diagonal projections:

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

$$f_2^3 = 4.5 + \frac{13-(5.5+4.5)}{2} = 6; \quad f_4^3 = 3.5 + \frac{7-(6.5+3.5)}{2} = 2$$

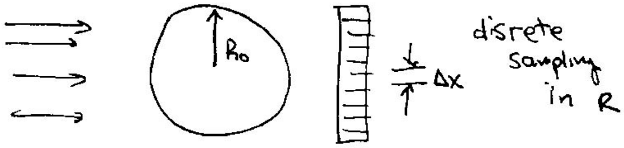


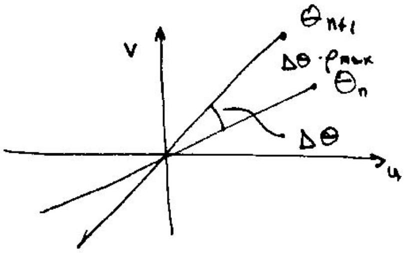
which we can see is consistent with all of the projections. While this procedure may look fine, in the presence of noise (that is, the projections aren't exactly consistent), this method converges very slowly, and in many cases, not at all – that is, the solution will continue to bounce around. Usually, one goes thought the entire set of projections multiple times. There are entire books written on image reconstruction in CT.

Practical Considerations in CT

Sampling

CT (the "C" meaning "computed") relies on the sampling of projection data for processing on a computer. There are 2 kinds of sampling – radial sampling (along the projection in *R*) and angular sampling (*θ*).





discrete sampling in θ

Sampling in R (e.g. Δx) depends on the spatial frequency content that you wish to represent in your final image. Subsampling in this domain results in the relatively benign spectral aliasing. Sampling in θ is a more complex, though, since it is really a sampling in the Fourier domain of the object and as such subsampling will results in aliasing of spatial information in the image.

Recall, that the mimimum sampling distance in the Fourier domain is dictated by the field of view. For example,

$$\Delta k_{\max} \le \frac{1}{FOV} = \frac{1}{2R_0}$$

where R_0 is the maximum extent of the object. Also, recognize that

$$\Delta k_{\max} = \Delta\theta \cdot \rho_{\max}$$

but the maximum spatial frequency that can be represented by a discretely sampled projection is:

$$\rho_{\max} = \frac{1}{2\Delta x}$$

and thus:

$$\Delta\theta \le \frac{\Delta x}{R_0}$$

For example, if our device has 512 samples across the field of view $\left(\frac{2R_0}{\Delta x} = 512\right)$, then:

$$\Delta\theta \le \frac{1}{256}$$

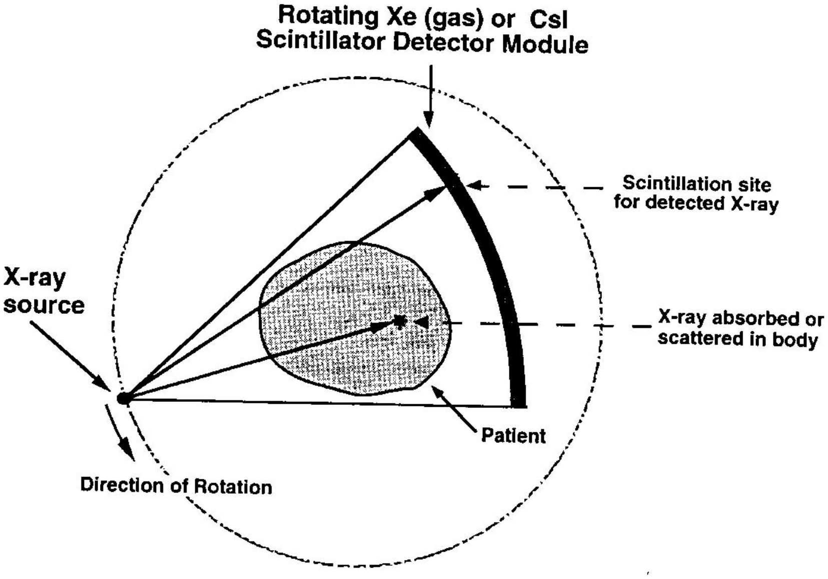
and the number of projections will be:

$$N_{proj} = \frac{\theta_{\max} - \theta_{\min}}{\Delta\theta} = 256\pi \approx 804$$

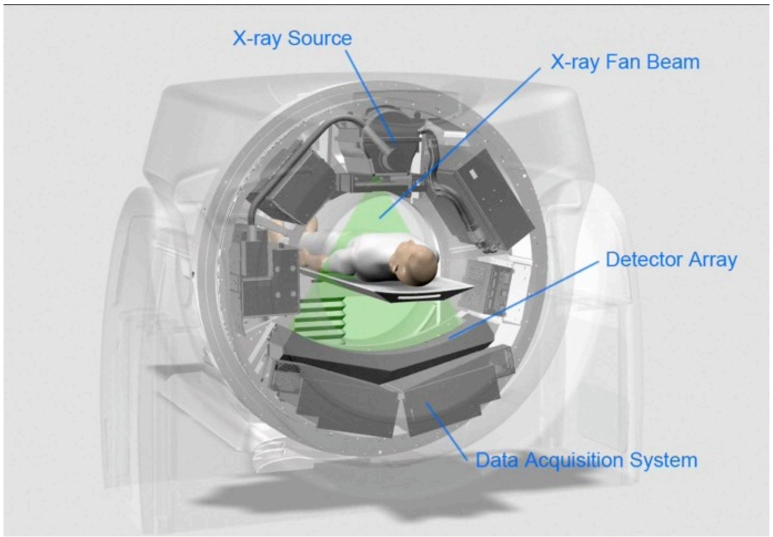
Thus approximately 804 projections would be required to fully sample the object for a final image of size roughly 512 by 512. Of course, we can always sample fewer angles and filter (smooth) the data to reduce $\rho_{\max}$ which reduces spatial resolution.

Fan-Beam Geometry

Our analysis of CT to date was for a parallel ray x-ray source, while in practice, we have a small source that projects a fan of x-rays:

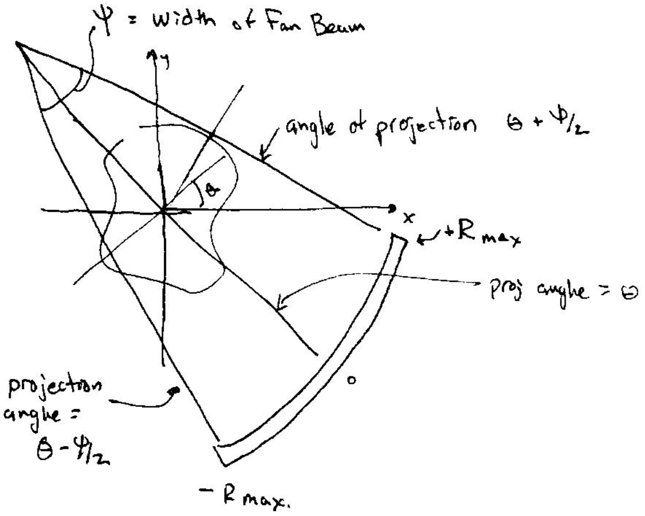


Macovski

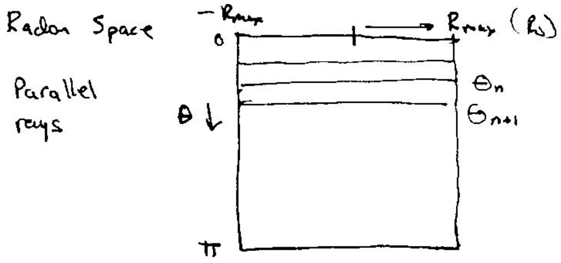


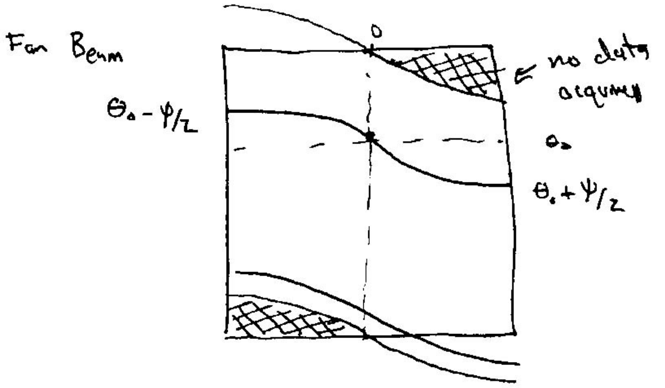
Analogic.com

For a given angle of the CT gantry, we actually collect a projection where were the angle of projection varies as a function of R. Observe that if the fan width is φ then the maximal R location will actually have an angle of $(\theta + \varphi/2)$, while the $R = 0$, will have an angle of θ and the minimal R will have an angle of $(\theta - \varphi/2)$:


ψ = width of Fan Beam
y
x
θ
angle of projection $\theta + \psi/2$
$+R_{max}$
proj angle = θ
0
projection angle = $\theta - \psi/2$
$-R_{max}$.

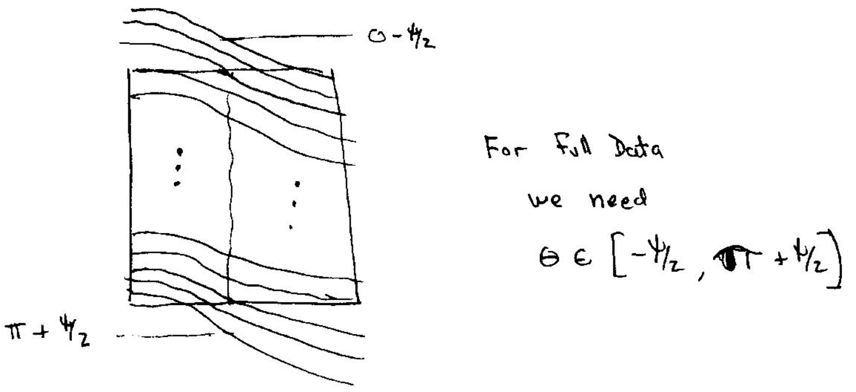
Question: Does the 1D FT of this projection result in a curved line in the Fourier domain? No – it results in a curved line in the sinogram space (Radon space):



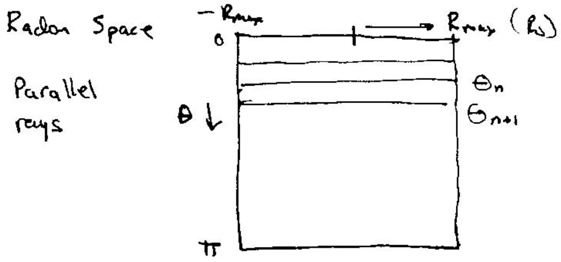


If we sample from $\theta \in [0, \pi)$ we dont acquire in the ▨ areas.

In order to be able to reconstruct the image we need to fill in the Radon space completely (recall the FT of horizontal lines will give lines in the Fourier domain and incomplete lines will limit our ability to fill in the Fourier domain). In order to fully sample the Radon space, we will need to sample projections over angles $\theta \in [-\varphi/2, \pi+\varphi/2]$ or over a ($\pi + \varphi$) range.



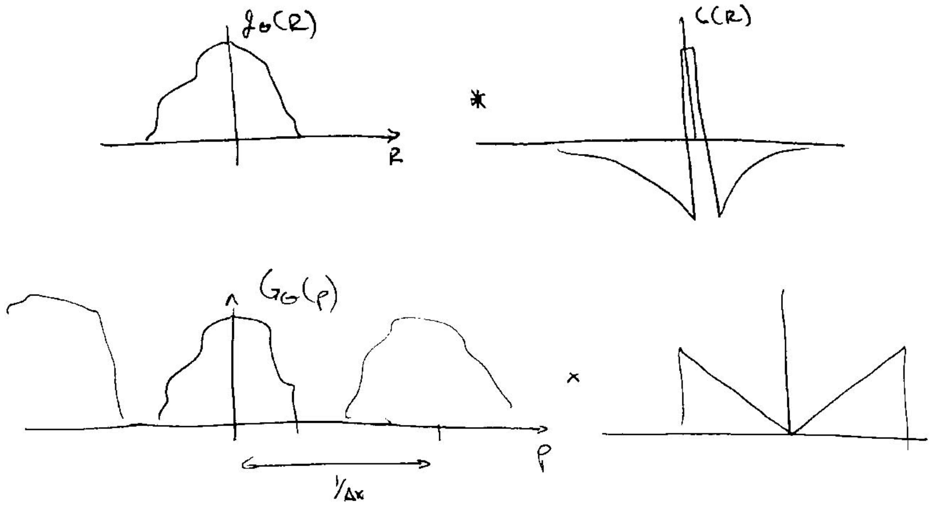
One way to reconstruct this data will be to resample the above to:



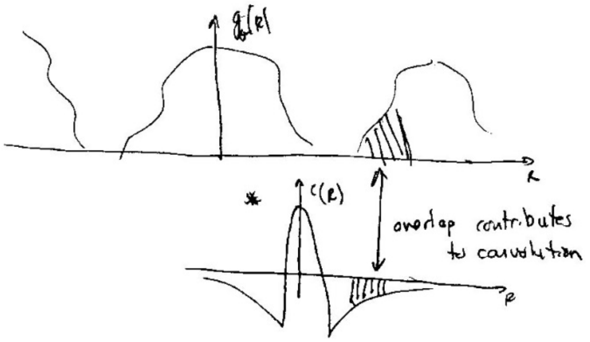
and then reconstruct using the usual methods. There are also filtered backprojection methods and iterative (e.g. ART) formulations that deal directly with fan beam data (see Kak or Zeng books, for example). For example, in filtered backprojection, instead of backprojecting along parallel rays, one backprojects along the fan beam geometry. There some corrections that relate to changing beam width along the path, but it works very similarly to the parallel beam filtered backprojection.

Circular Convolution

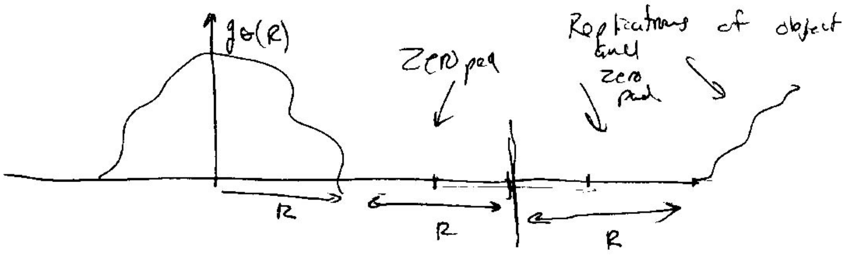
When performing filtering, it is often convenient to do this filtering in the Fourier domain.



One problem that arises is circular convolution. Because the DFT assumes that the object is periodic, the convolution function (which has long tails) will extend into the replicated versions of the object:

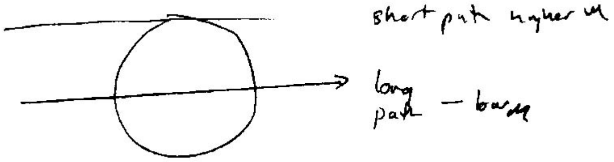


The typical solution to this problem is to zero pad the object prior to Fourier transformation for filtering:

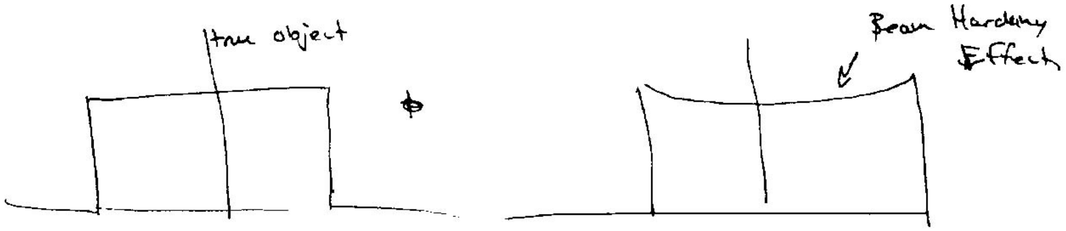


Beam Hardening

Beam hardening manifests itself as an apparent decrease in μ as the beam passes through the object. Thus, for short paths through the object (e.g., near the edge) the apparent μ is higher than for long paths through the object (e.g., near the center):



One effect of this is a "cupping" of the images of μ, which results in an apparent reduction in image intensity at the center of the object:


true object
Φ
Beam Hardening Effect