Practical considerations - back slice /projection reconstruction

If we want to build a practical x-ray scanning system such as the one described last time, There are a variety of considerations we must accommodate. The math tells us the back-slice technique works, but how well and under what conditions.

Back-projection methods

Layergram. Simply acquiring each slice and accumulating the measurements in the u-v plane leads to the layergram, which can be defined by the formula

$$\ell(x,y) = N^{-1} \sum_{i=1}^{N} f_{\theta_i}(x\cos\theta_i + y\sin\theta_i)$$

where $f_{\theta_i}()$ is the projection acquired at angle θ_i and we assume that the θ_i span $0 < \theta < \pi$ with equal spacing. This is not the same as the original but is related. Here, each scan is back-projected and the projections summed.

Because of the density of overlapping back projections near the origin, the distribution here is over-represented. Consider the response of the layergram to a unit mass at the origin. Then each slice is simply a δ-function at the origin. Back projecting each of these yields a line through the center. Thus the layergram becomes

$$\ell(x,y) = N^{-1} \sum_{i=1}^{N} \delta\left[y\cos\frac{\pi i}{N} - x\sin\frac{\pi i}{N}\right]$$

The resulting layergram can be pictured as follows:

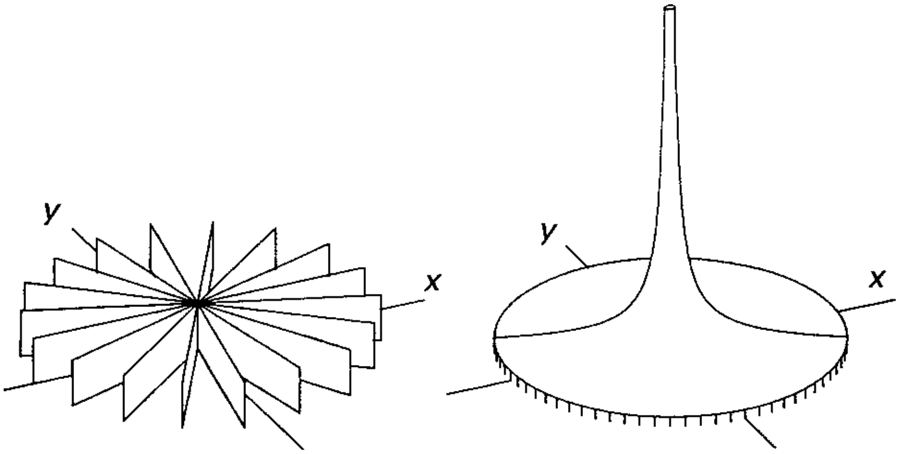


Figure 15-4 The point-mass layergram $l_0(x, y, z)$ formed from 18 equispaced scans (left) and the inverse dependence on radius to which it is likened (right).

From this we can see that the impulse response of the layergram is r^{-1}, hence the relation of the layergram to the true distribution is

$$\ell(x,y) = r^{-1} ** f(x,y)$$

We can compensate for this impulse response by examining the Fourier transform of the layergram.

$$L(u,v) = F(u,v)\left\{N^{-1}\sum_{i=1}^{N}\delta\left(u\cos\frac{\pi i}{N} + v\sin\frac{\pi i}{N}\right)\right\}$$

In other words, the transform of the layergram is the true spectrum, oversampled near the center and undersampled near the edges. The sampling pattern is a spoke pattern defined by the scans. This results in a sampling density proportional to q^{-1}, hence multiplication by q "restores" the proper weighting.

Of course, multiplication by q does not converge, and generates very high weights for large q. But, if we can limit the range of q we can minimize this problem. Limiting q simply means not solving for spatial frequencies beyond some cutoff value.

The transform is thus multiplied by

$$q \operatorname{rect}\left(\frac{q}{2M}\right)$$

where M is the maximum frequency desired. This factor can be written

$$\operatorname{rect}\left(\frac{q}{2M}\right) - \Lambda\left(\frac{q}{M}\right)$$

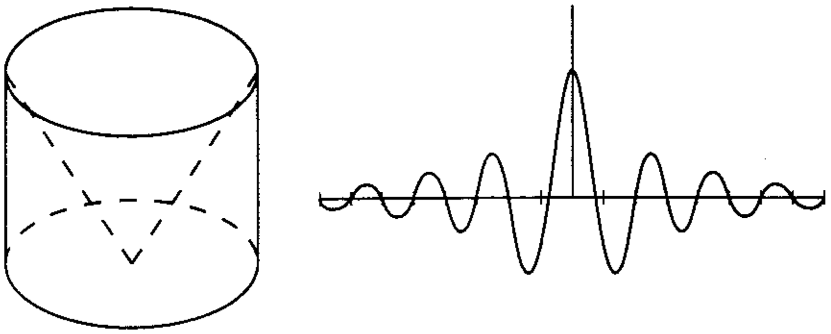


Figure 15-5 The conical crater rect$(q/2M) - \Lambda(q/M)$ and a radial section through its two-dimensional Fourier transform.

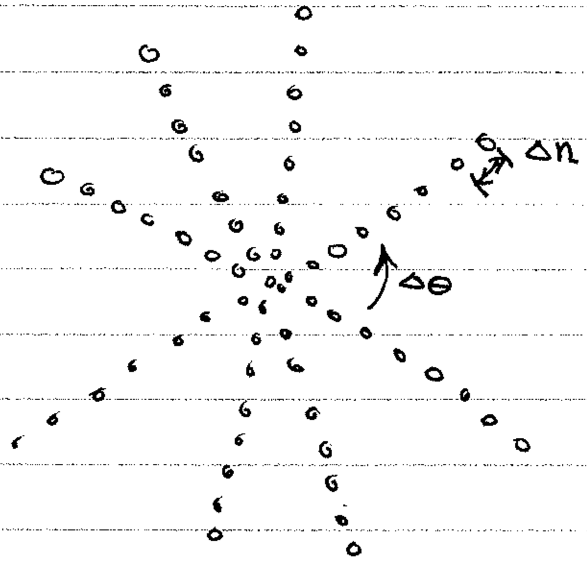
In the time domain this is seen to be convolution with

$$4M^2 \operatorname{jinc} 2Mr - M^2 \mathcal{H}(\Lambda(Mr))$$

<u>How many scans?</u>

How many scans are needed to reconstruct a function? We know there is a limit to ΔR for sampling adequately, so what is the limit to the step $\Delta\Theta$ needed. This gives us the number of needed scans $\frac{\pi}{\Delta\Theta}$.

Here is the sampling pattern:



For an object of diameter D, Bracewell showed that the number $N = \pi D M$, where M is the cutoff frequency we have chosen. This follows from D^{-1} being the critical sampling interval in the u-v plane. The ~~diameter of a~~ circumference of a circle of radius M is $2\pi M$, hence sampling at D^{-1} yields

$$N = \frac{\pi M}{D^{-1}} = \pi M D$$

This is critical sampling, we choose in fact an N slightly larger.

Algorithm comparison

So we have briefly looked at two algorithms, one Fourier based and one back projection based. The relative merits of each will ~~be~~ depend on the specific case to be evaluated. Historically, in the medical systems, back projection was used first because it was faster than the 2-D Fourier reconstruction.

Practical aspects for comparison also concern whether the full 180 degrees of scan are available, or other causes for unequal spacing.

Refraction, or the bending of rays through propagation, also distorts the result. Again a case by case evaluation is needed.

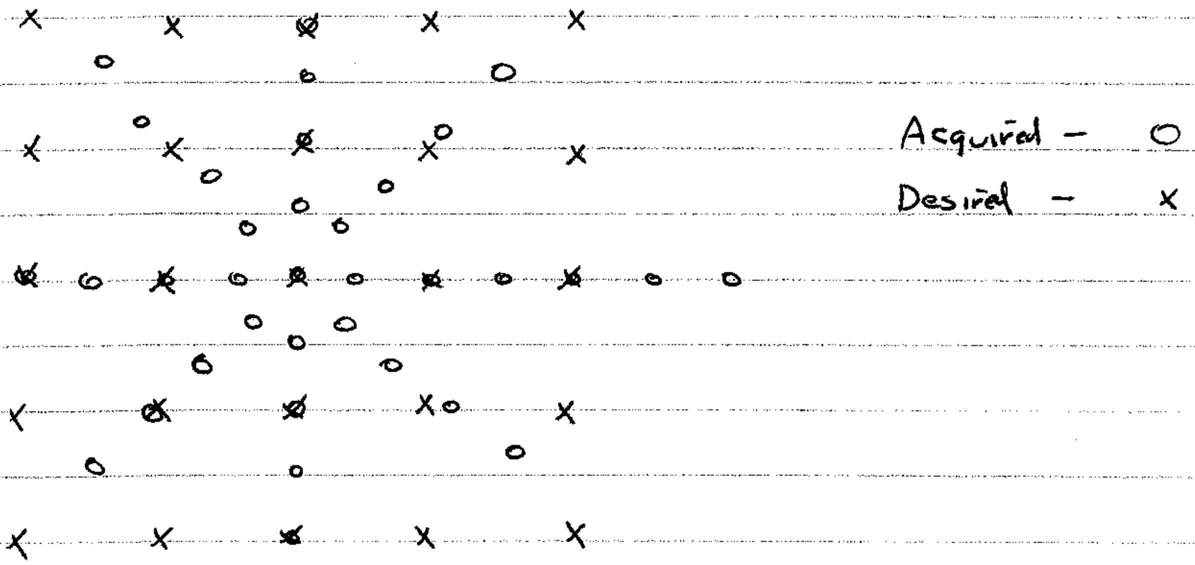
Interpolation

If we are to reconstruct a function efficiently, we would like to take advantage of fast transform algorithms such as the FFT.

Unfortunately, scan data are acquired in polar coordinates, while FFT's are applicable only to data sampled on square, Cartesian grids. Thus the measurements must be interpolated.

For the FFT reconstruction, we will have more points than we need near the center, and usually the critical amount at the edges. But if the interpolation uses only the nearest points, it can be reasonably efficient and result in a grid for F(u,v) that is not overly weighted in the center.

Consider the following diagram of available and desired sample points:



For the points along the axes, no interpolation is needed. But for the others, perhaps nearest neighbor approaches suffice. If more accuracy is desired, twisted plane or even higher order methods may be called for.

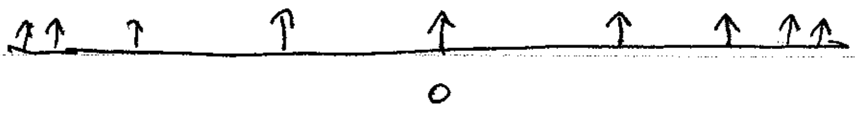
But, since we are not summing all of the data, we do not have the problem of weighting "down" the signal near the origin. This, plus the availability of FFTs, make the transform methods desirable. The main drawbacks are:

- We have to evaluate a 2-D Fourier transform
- We "waste" samples near the origin

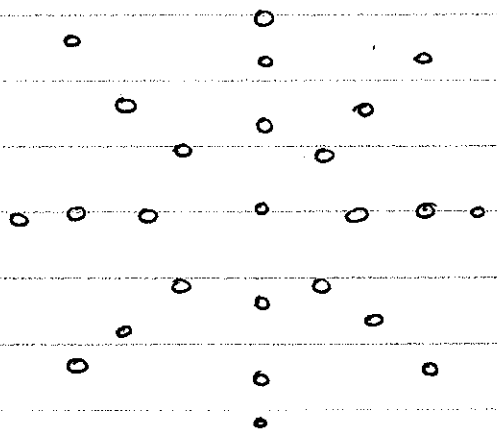
How might we mitigate these?

First, we want fast, efficient interpolations so that the FFT can be used. DFTS are far too slow!

We can sample our projections at a density inversely proportional to distance from the origin:



Then, in 2-D:



We now do not have the overabundance in the center, and our memory is more efficiently used. I don't know if anyone uses this approach, but perhaps the advantage is less important now that memory is cheaper. But for high resolution ~~ore~~ or large area reconstructions it may still be an issue.