EL582/BE620 --- Medical Imaging - I

Computed Tomography (part I)

Yao Wang

Polytechnic University, Brooklyn, NY 11201

Based on J. L. Prince and J. M. Links, Medical Imaging Signals and Systems, and lecture notes by Prince. Figures are from the textbook.

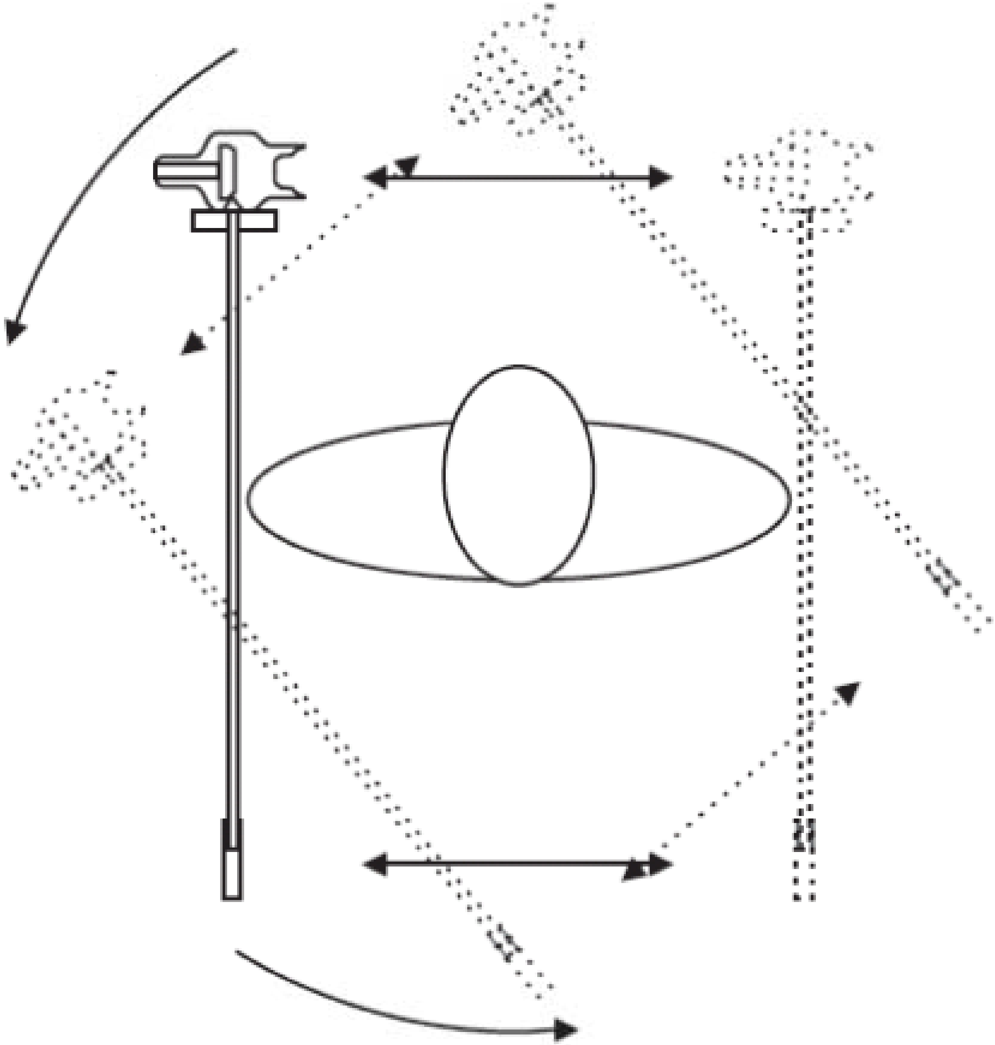
Lecture Outline

- Instrumentation
  - CT Generations
  - X-ray source and collimation
  - CT detectors
- Image Formation
  - Line integrals
  - Parallel Ray Reconstruction
    - Radon transform
    - Back projection
    - Filtered backprojection
    - Convolution backprojection
    - Implementation issues

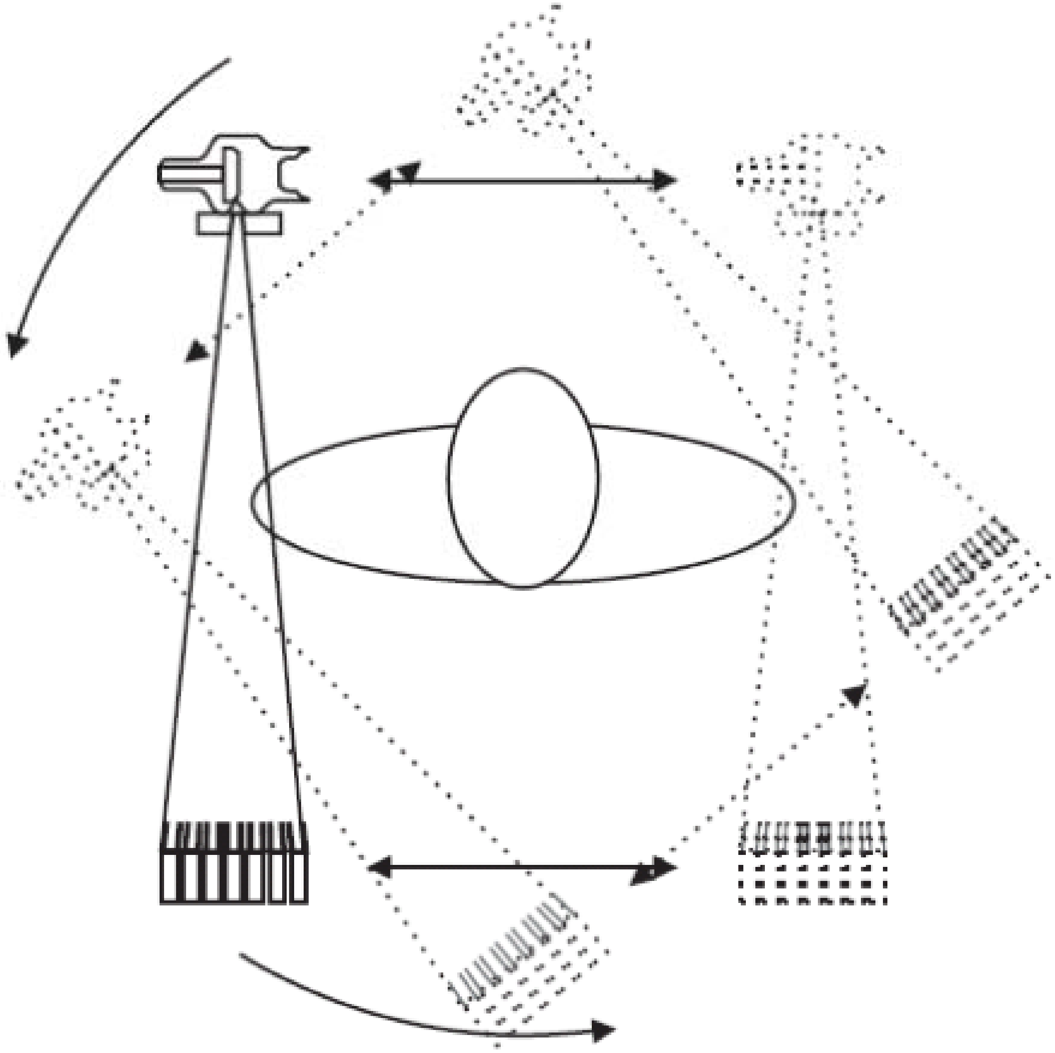
Limitation of Projection Radiography

- Projection radiography
  - Projection of a 2D slice along one direction only
  - Can only see the “shadow” of the 3D body
- CT: generating many 1D projections in different angles
  - When the angle spacing is sufficiently small, can reconstruct the 2D slice very well

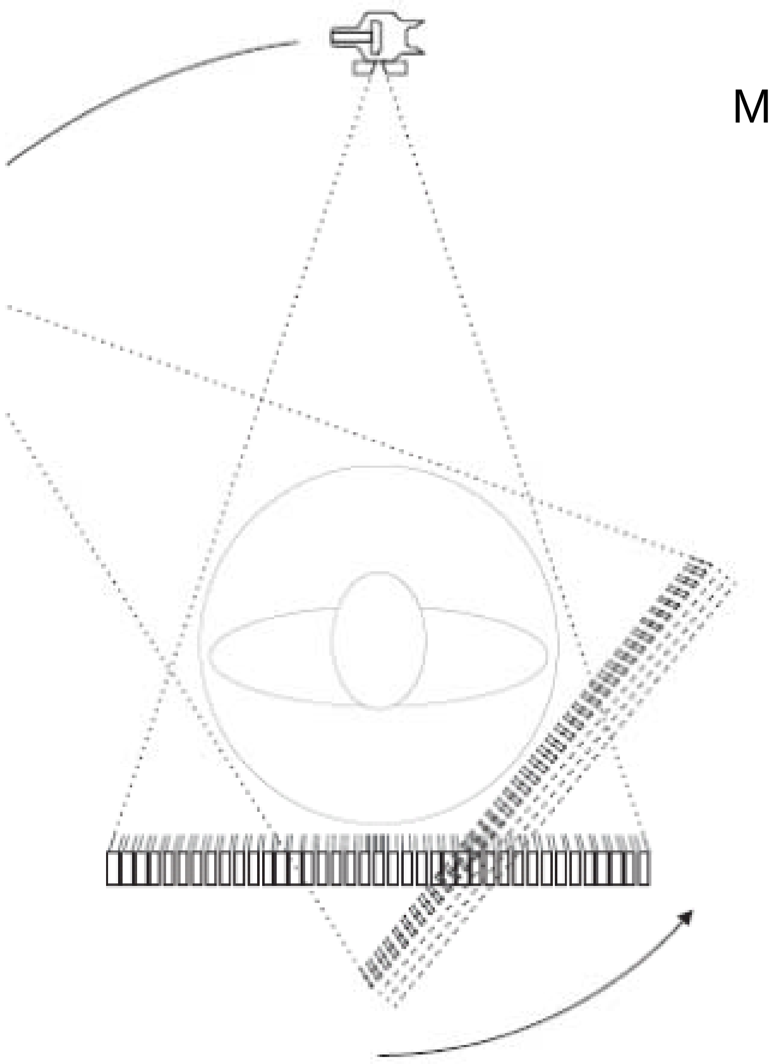
1st Generation CT: Parallel Projections



2^{nd} Generation

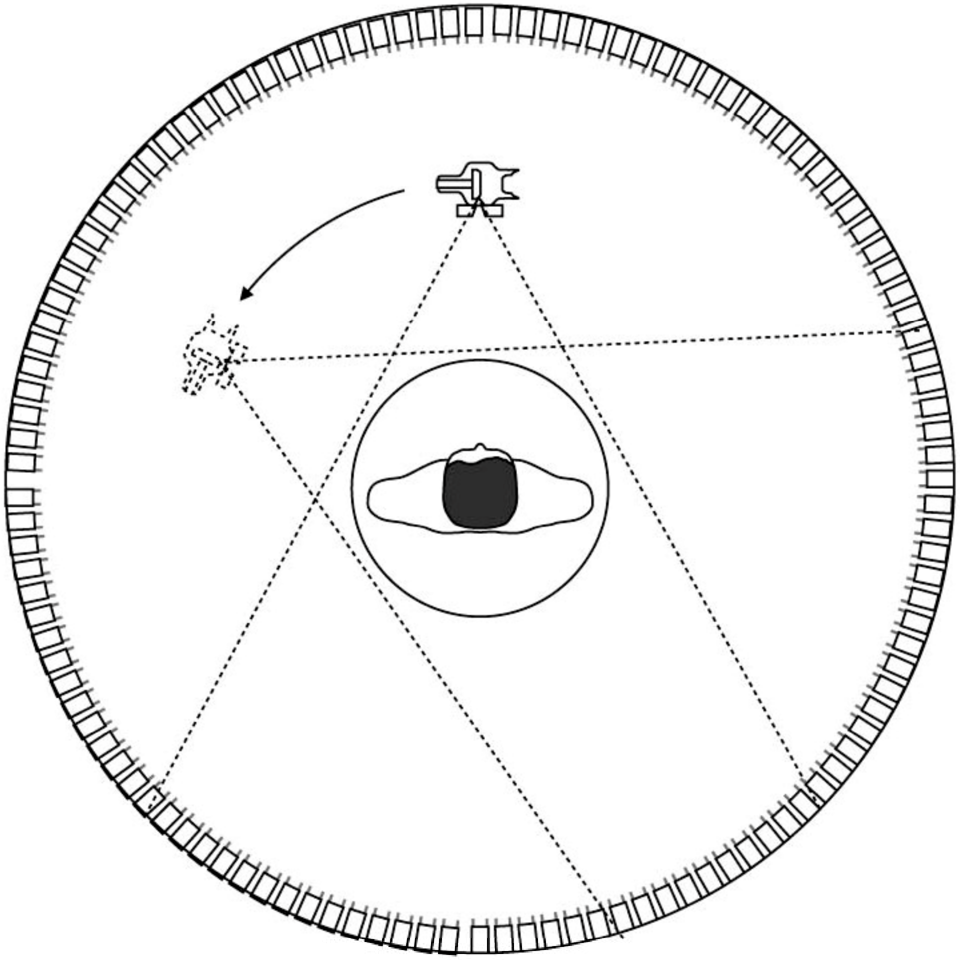


3G: Fan Beam



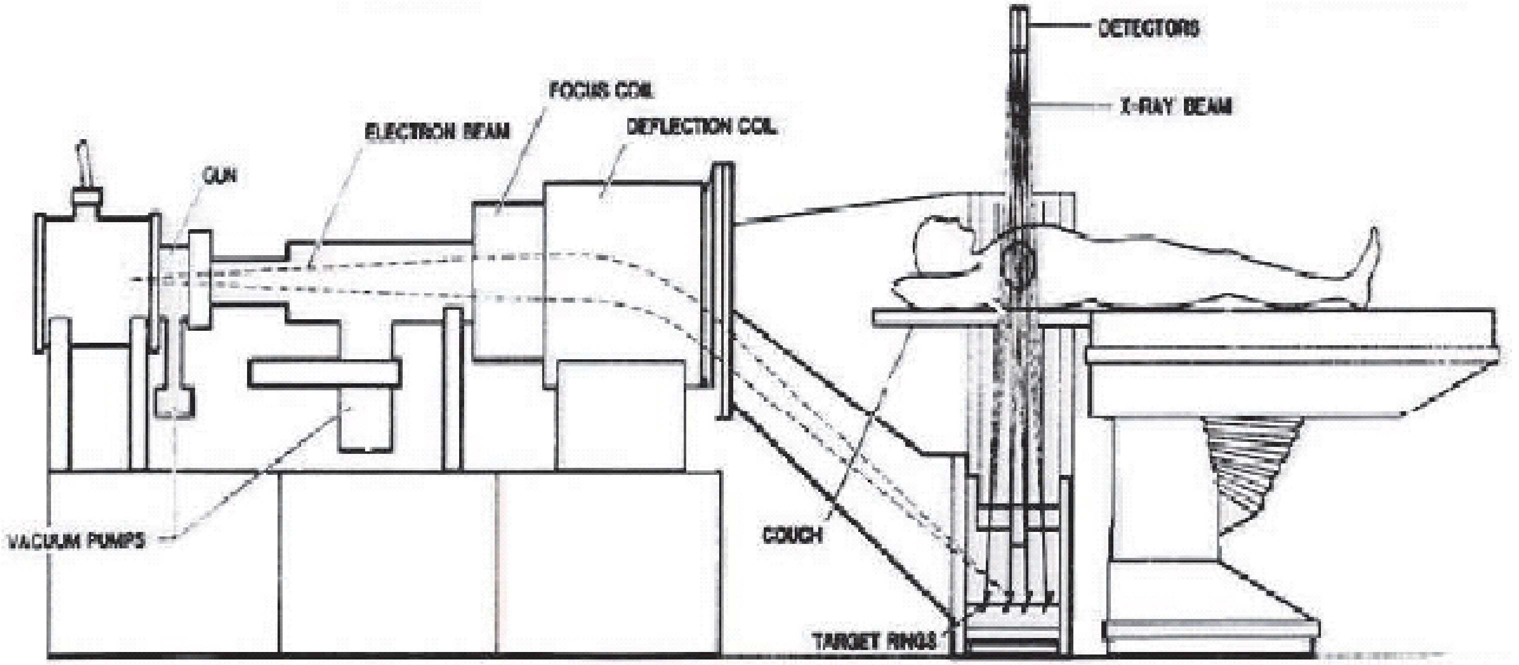
Much faster than 2G

4G

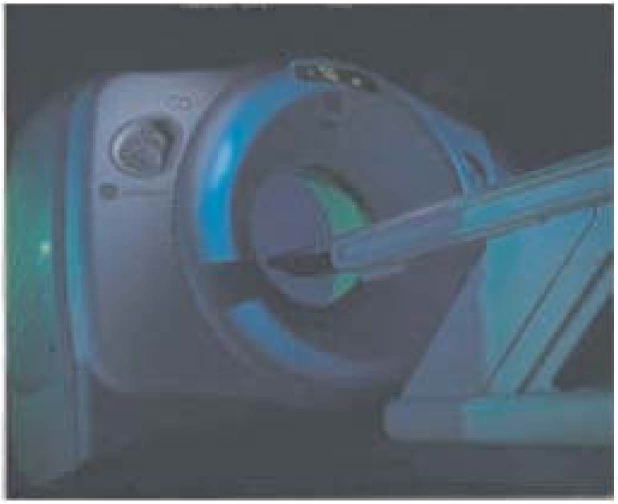


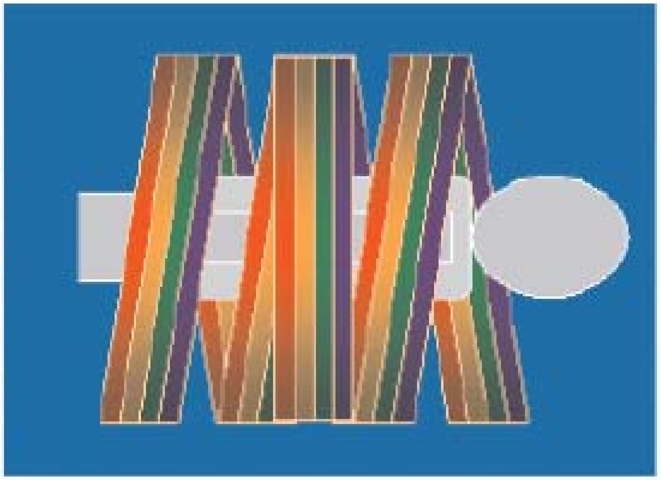
Fast
Cannot use collimator at detector, hence affected by scattering

5G: Electron Beam CT (EBCT)



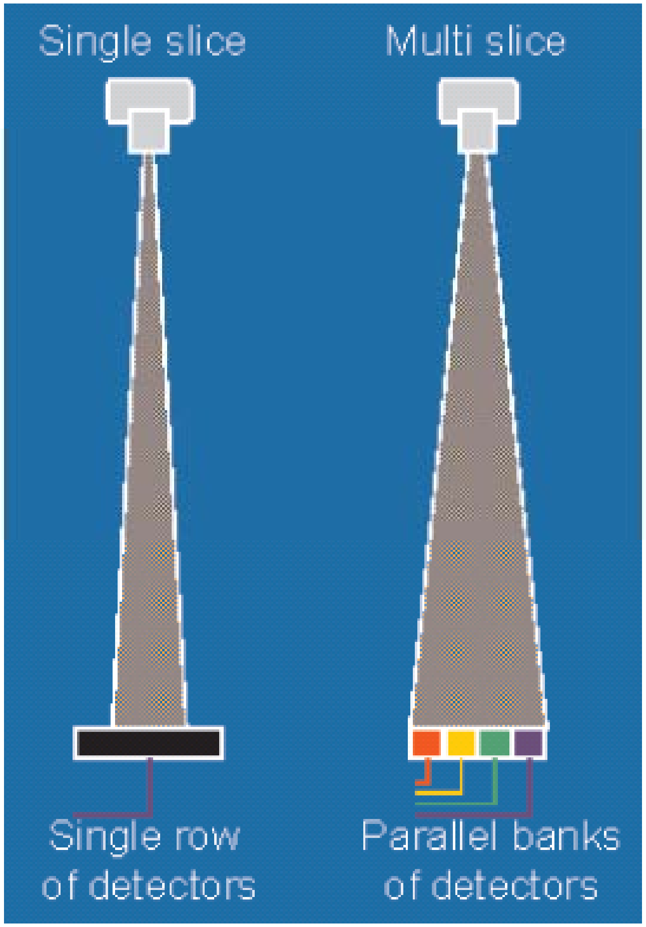
6G: Helical CT



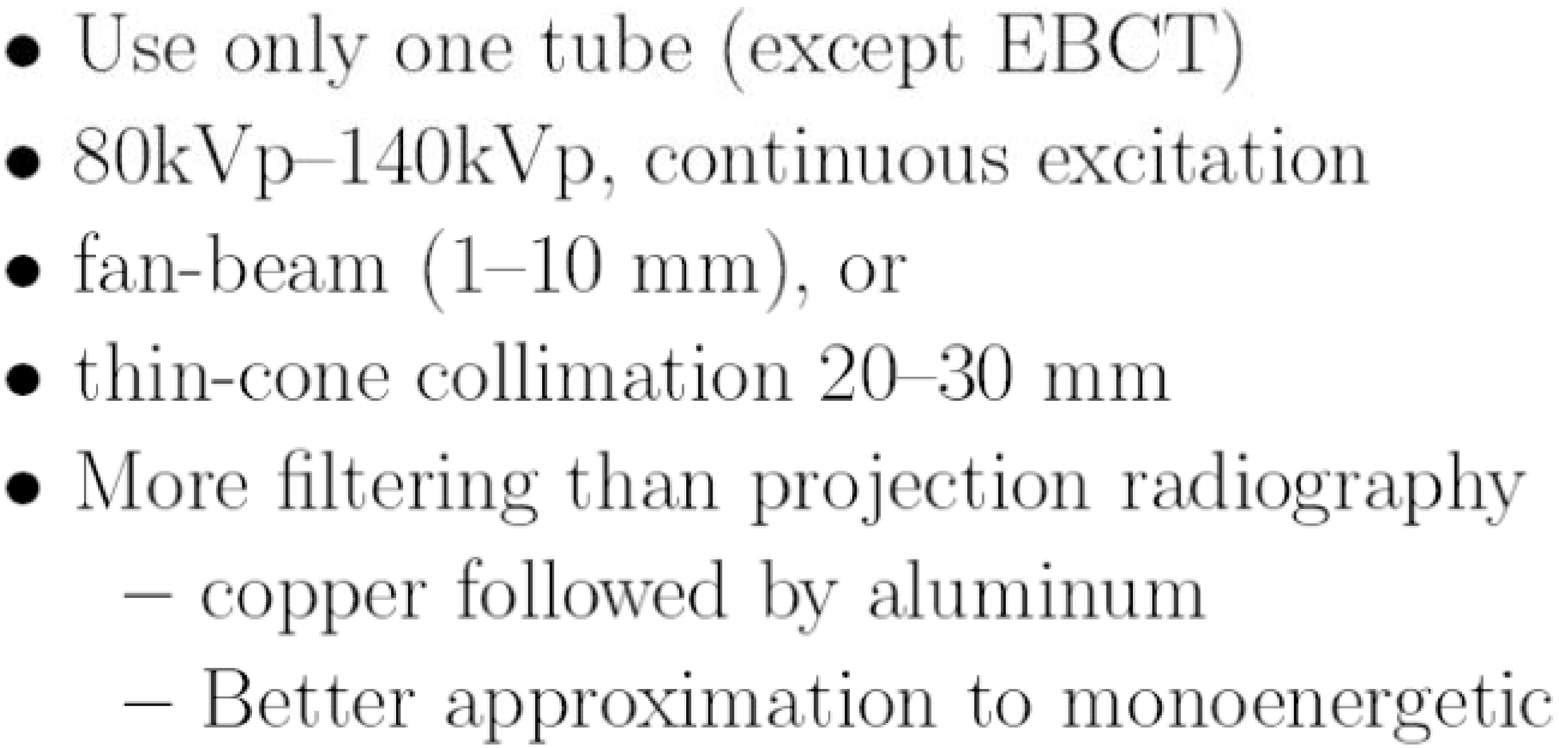


7G: Multislice

- Features:
  - 40 parallel detector rows
  - 35,840 detector elements
  - 32mm detector length
  - 16 0.5mm slices with each second gantry revolution
- CT is becoming “cone beam”



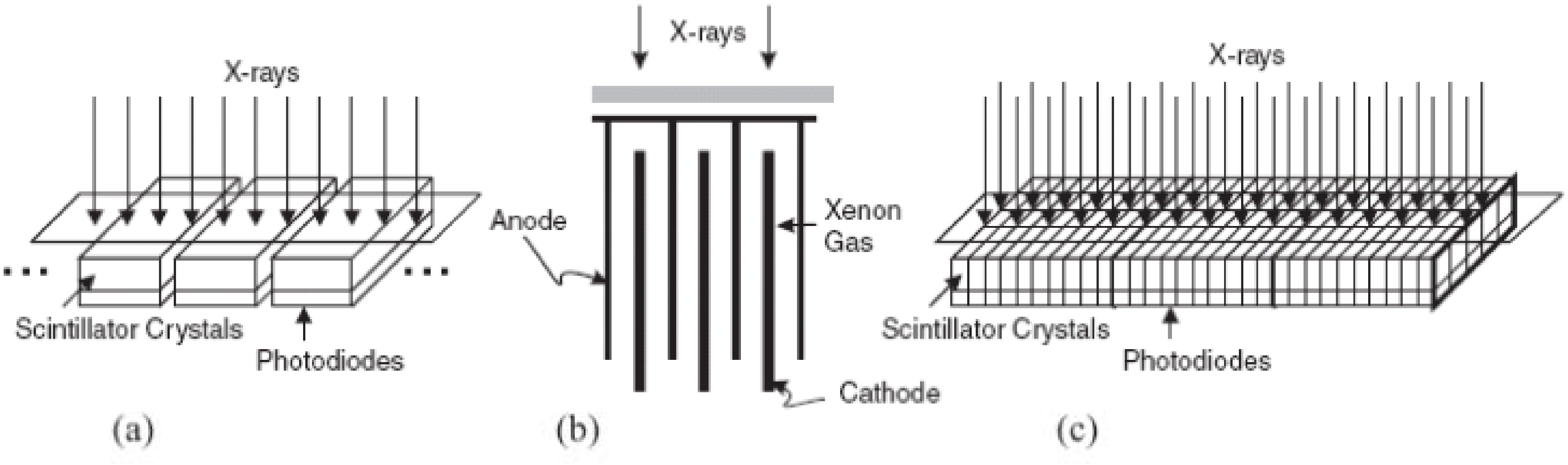
X-ray Source



- Use only one tube (except EBCT)
- 80kVp–140kVp, continuous excitation
- fan-beam (1–10 mm), or
- thin-cone collimation 20–30 mm
- More filtering than projection radiography
  - copper followed by aluminum
  - Better approximation to monoenergetic

X-ray Detectors

- Most are solid-state:
  - scintillation crystal
  - solid state photo-diode



CT Measurement Model

- Monoenergetic model:

$$I_d = I_0 \exp\left\{-\int_0^d \mu(s;\bar{E})ds\right\}$$

- $\bar{E}$ is effective energy

  $\bar{E}$ is that energy which in a given material will produce the same measured intensity from a monoenergetic source as from the actual polyenergetic source.

- Observe I_d
- Rearrange monoenergetic model:

$$\begin{aligned} g_d &= -\ln \frac{I_d}{I_0} \\ &= \int_0^d \mu(s;\bar{E})ds \end{aligned}$$

- g_d is a line integral of the linear attenuation coefficient at the effective energy
- Note: Requires calibration measurement of I_0

CT Number

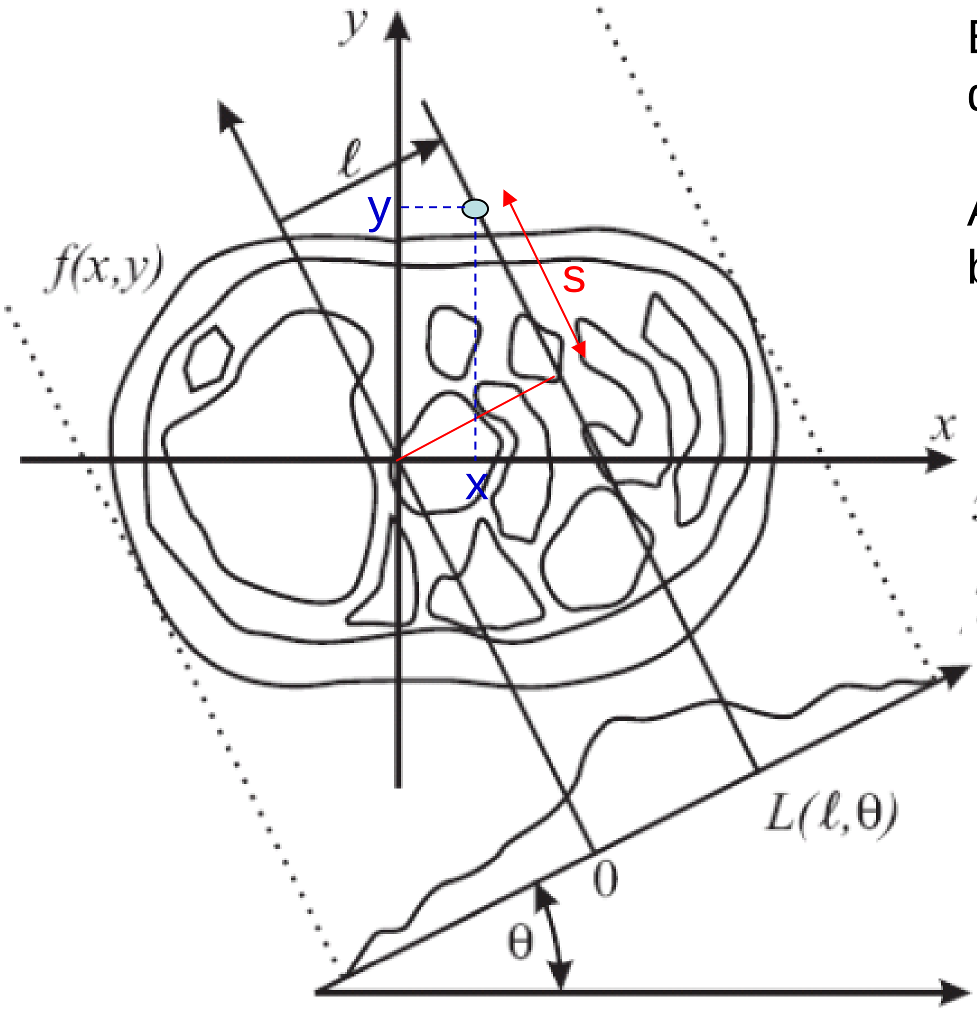
- Consistency across CT scanners desired
- CT number is defined as:

$$h = 1000 \times \frac{\mu - \mu_{\text{water}}}{\mu_{\text{water}}}$$

- h has Hounsfield units (HU)
- Usually rounded or truncated to nearest integer
- Range: $-1{,}000$ to $\sim 3{,}000$

Need 12 bits to represent

Parameterization of a Line



Each projection line is defined by (l,θ)

A point on this line (x,y) can be specified with two options

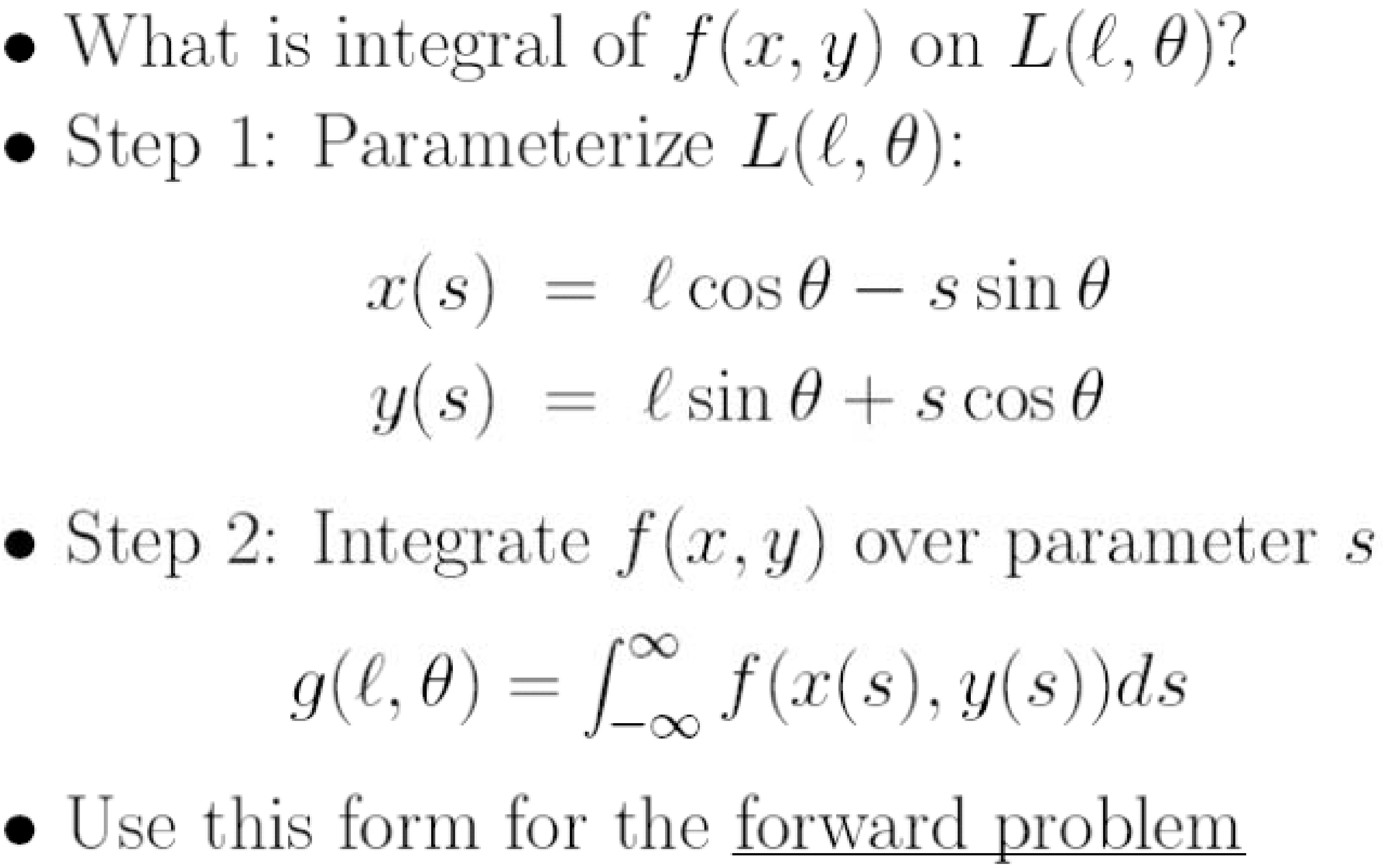
Option 1 (parameterized by s):

$$\begin{aligned} x(s) &= \ell\cos\theta - s\sin\theta \\ y(s) &= \ell\sin\theta + s\cos\theta \end{aligned}$$

Option 2:

$$x\cos\theta + y\sin\theta = l$$

Line Integral: parametric form



- What is integral of $f(x,y)$ on $L(\ell,\theta)$?
- Step 1: Parameterize $L(\ell,\theta)$:

$$\begin{aligned} x(s) &= \ell\cos\theta - s\sin\theta \\ y(s) &= \ell\sin\theta + s\cos\theta \end{aligned}$$

- Step 2: Integrate $f(x,y)$ over parameter s

$$g(\ell,\theta) = \int_{-\infty}^{\infty} f(x(s),y(s))ds$$

- Use this form for the forward problem

Line Integral: set form

- Integrate over whole plane; non-zero only on $L(\ell, \theta)$
- Key is sifting property

$$q(\ell) = \int_{-\infty}^{\infty} q(\ell')\delta(\ell' - \ell)d\ell'$$

- Use line impulse on $L(\ell, \theta)$

$$g(\ell, \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\delta(x\cos\theta + y\sin\theta - \ell)\, dxdy$$

Physical meaning of “f” & “g”

- Recall monoenergetic model:

$$I_d = I_0 \exp\left\{-\int_0^d \mu(s;\bar{E})ds\right\}$$

- Rearrange:

$$-\ln\frac{I_d}{I_0} = \int_0^d \mu(x(s),y(s);\bar{E})ds$$

- Relationship is:

$$\begin{aligned} f(x,y) &= \mu(x,y;\bar{E}) \\ g(\ell,\theta) &= -\ln\frac{I_d}{I_0} \end{aligned}$$

What is g(l,θ)?

- Fix ℓ and θ: line integral of $f(x, y)$
- Fix θ: projection of $f(x, y)$ at angle θ
- Function of θ and ℓ:

$g(\ell, \theta)$ is the Radon transform of $f(x, y)$

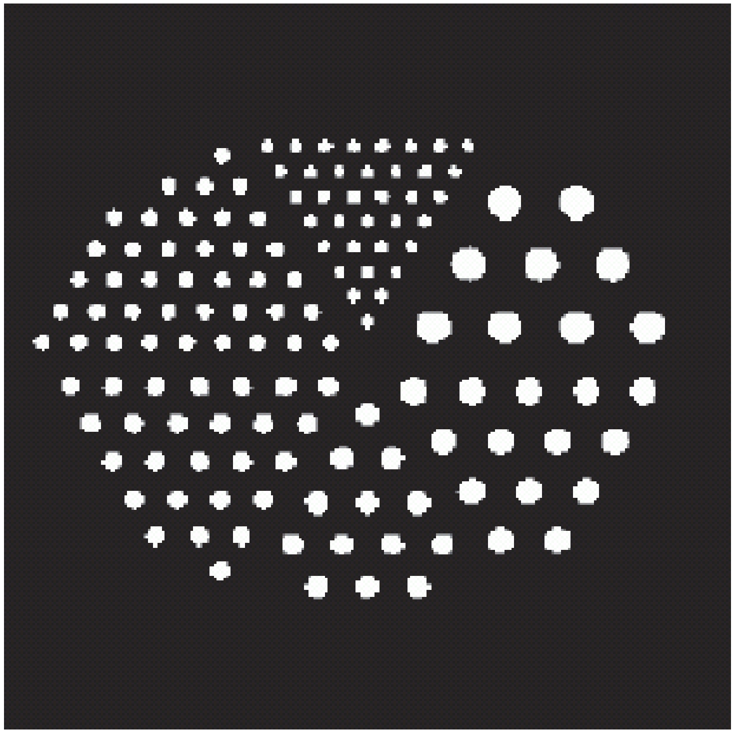
$$g(\ell, \theta) = \mathcal{R}\{f(x, y)\}$$

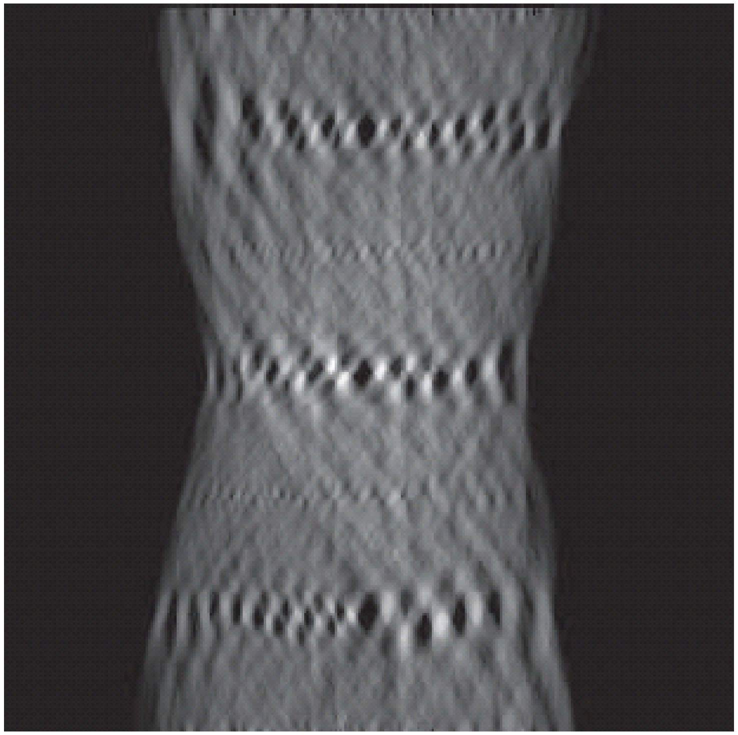
Example

- Example 1: Consider an image slice which contains a single square in the center. What is its projections along 0, 45, 90, 135 degrees?
- Example 2: Instead of a square, we have a rectangle. Repeat.

Sinogram

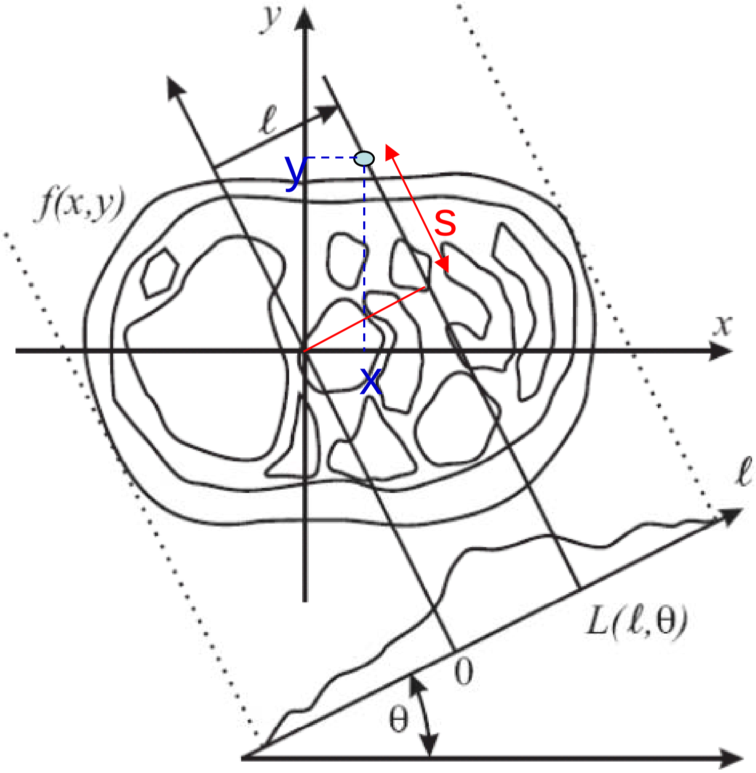
- CT data acquired for collection of ℓ and θ
- CT scanners acquires a sinogram



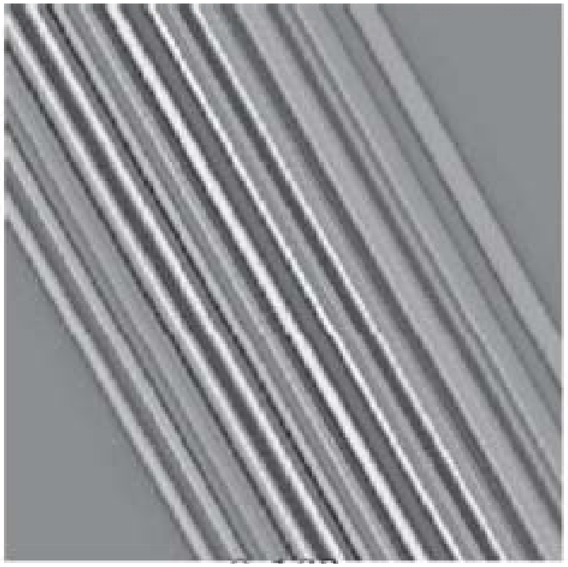


Backprojection

- The simplest method for reconstructing an image from a projection along an angle is by backprojection
  - Assigning every point in the image along the line defined by (l,θ) the projected value g(l, θ), repeat for all l for the given θ



- $b_\theta(x, y)$ is a laminar image



Two Ways of Performing Backprojection

- Option 1: assigning value of g(l, θ) to all points on the line (l, θ)
  - g(l, θ) is only measured at certain l: l_n=n Δl
  - If l is coarsely sampled (Δl is large), many points in an image will not be assigned a value
  - Many points on the line may not be a sample point in a digital image
- Option 2: going through all sampling points (x,y) in an image, find its corresponding “l=x cos θ+y sin θ” for the given θ, take the g value for (l, θ)

$$b_\theta(x, y) = g(x\cos\theta + y\sin\theta, \theta)$$

  - g(l, θ) is only measured at certain l: l_n=n Δl
  - must interpolate g(l, θ) for any l from given g(l_n, θ)
- Option 2 is better, as it makes sure all sample points in an image are assigned a value
- For more accurate results, the backprojected value at each point should be divided by the length of the underlying image in the projection direction (if known)

Backprojection Summation

- “Add up” all the backprojection images:

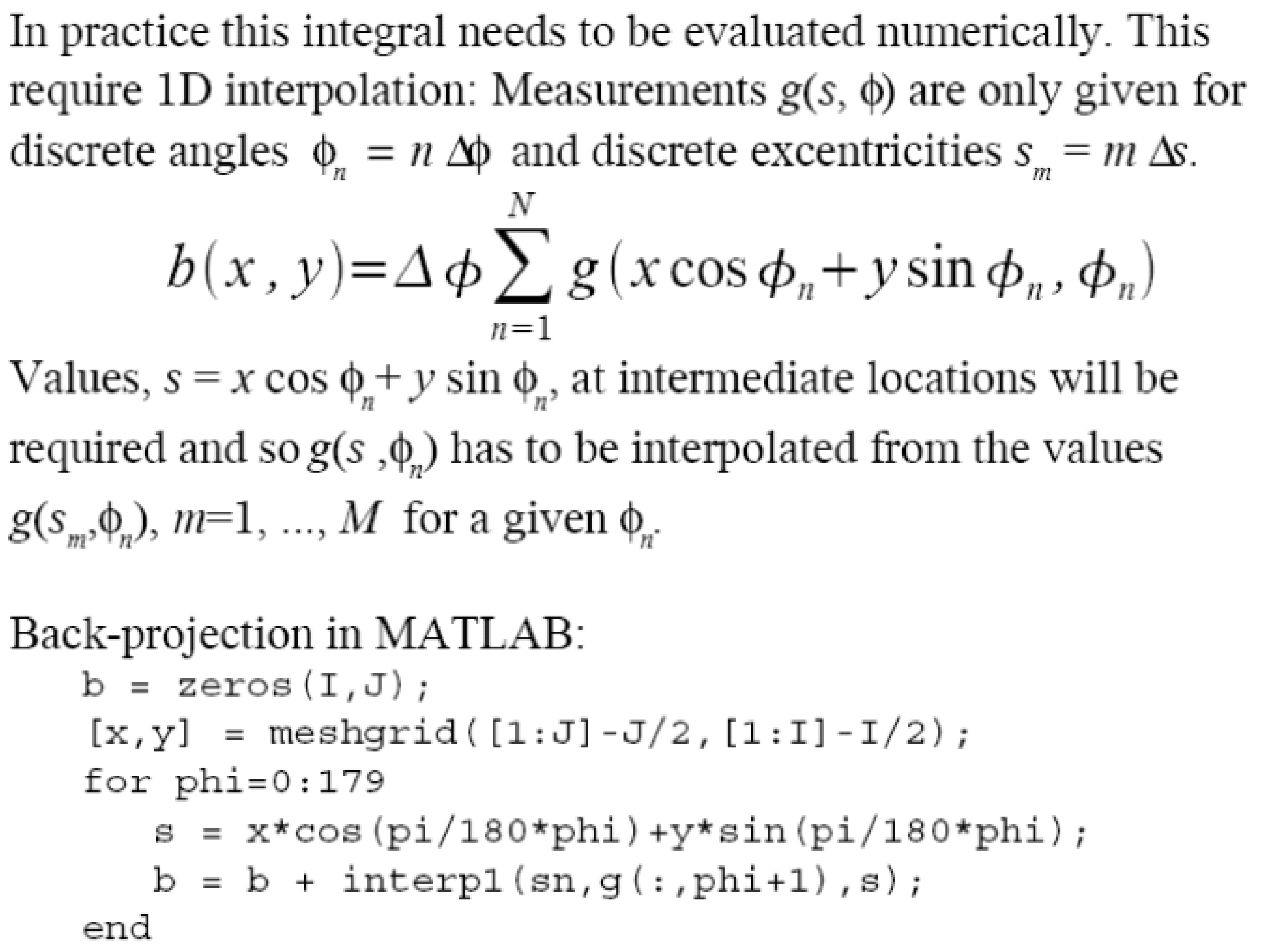
$$
\begin{aligned}
f_b(x,y) &= \int_0^{\pi} b_\theta(x,y)\,d\theta \\
&= \int_0^{\pi} g(x\cos\theta + y\sin\theta, \theta)\,d\theta
\end{aligned}
$$

Replaced by a sum in practice

$$
= \int_0^{\pi} \left[g(\ell,\theta)\right]_{\ell = x\cos\theta + y\sin\theta}\,d\theta
$$

- $f_b(x,y)$ is called a laminogram or backprojection summation image

Implementation Issues



In practice this integral needs to be evaluated numerically. This require 1D interpolation: Measurements $g(s, \phi)$ are only given for discrete angles $\phi_n = n\,\Delta\phi$ and discrete excentricities $s_m = m\,\Delta s$.

$$b(x,y)=\Delta\phi\sum_{n=1}^{N} g(x\cos\phi_n+y\sin\phi_n,\phi_n)$$

Values, $s = x\cos\phi_n + y\sin\phi_n$, at intermediate locations will be required and so $g(s, \phi_n)$ has to be interpolated from the values $g(s_m,\phi_n)$, $m=1, \ldots, M$ for a given ϕ_n.

Back-projection in MATLAB:

```
b = zeros(I,J);
[x,y] = meshgrid([1:J]-J/2,[1:I]-I/2);
for phi=0:179
    s = x*cos(pi/180*phi)+y*sin(pi/180*phi);
    b = b + interp1(sn,g(:,phi+1),s);
end
```

From L. Parra at CUNY, http://bme.ccny.cuny.edu/faculty/parra/teaching/med-imaging/lecture4.pdf

Implementation: Projection

- To create projection data using computers, also has similar problems. Possible l and q are both quantized. If you first specify (l,q), then find (x,y) that are on this line. It is not easy. Instead, for given q, you can go through all (x,y) and determine corresponding l, quantize l to one of those you want to collect data.
- Sample matlab code (for illustration purpose only)

```
[x,y]=meshgrid([0:J-1]-J/2,[0:I-1]-I/2);
N=ceil(sqrt(I*I+J*J))+1;
N0= floor((N-1)/2);
ql=1;
G=zeros(N,180);
for phi=0:179
for (x=-J/2:J/2-1; y=-I/2:I/2-1)
       l=x*cos(phi*pi/180)+y*sin(phipi/180);
        l=round(l/ql)+N0+1;
        If (l>=1) && (l<=N)
            G(l,phi+1)=G(l,phi+1)+f(x+J/2+1,y+I/2+1);
        End
        end
end
```

Example

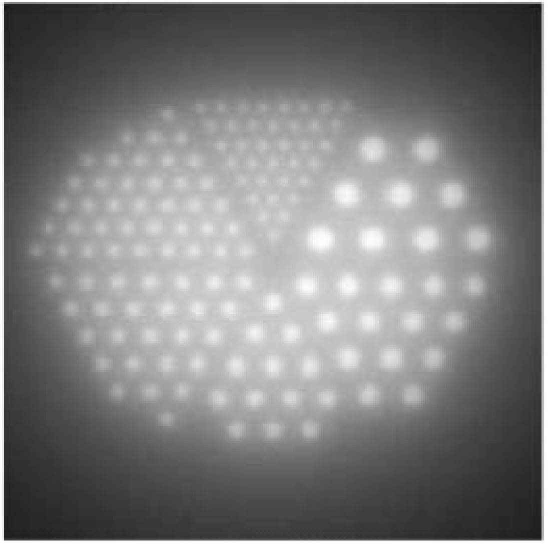
- Continue with the example of the image with a square in the center. Determine the backprojected image from each projection and the reconstruction by summing different number of backprojections

Problems with Backprojection

- "Bright spots" tend to reinforce → Blurring
- Problem:

$$f_b(x, y) \neq f(x, y)$$

- What is wrong?



Projection Slice Theorem

- Radon transform:

$$g(\ell,\theta)=\mathcal{R}\left\{f(x,y)\right\}$$

- Fourier transforms:

$$G(\rho,\theta)=\int_{-\infty}^{\infty} g(l,\theta)\exp\{-j2\pi\rho l\}dl$$

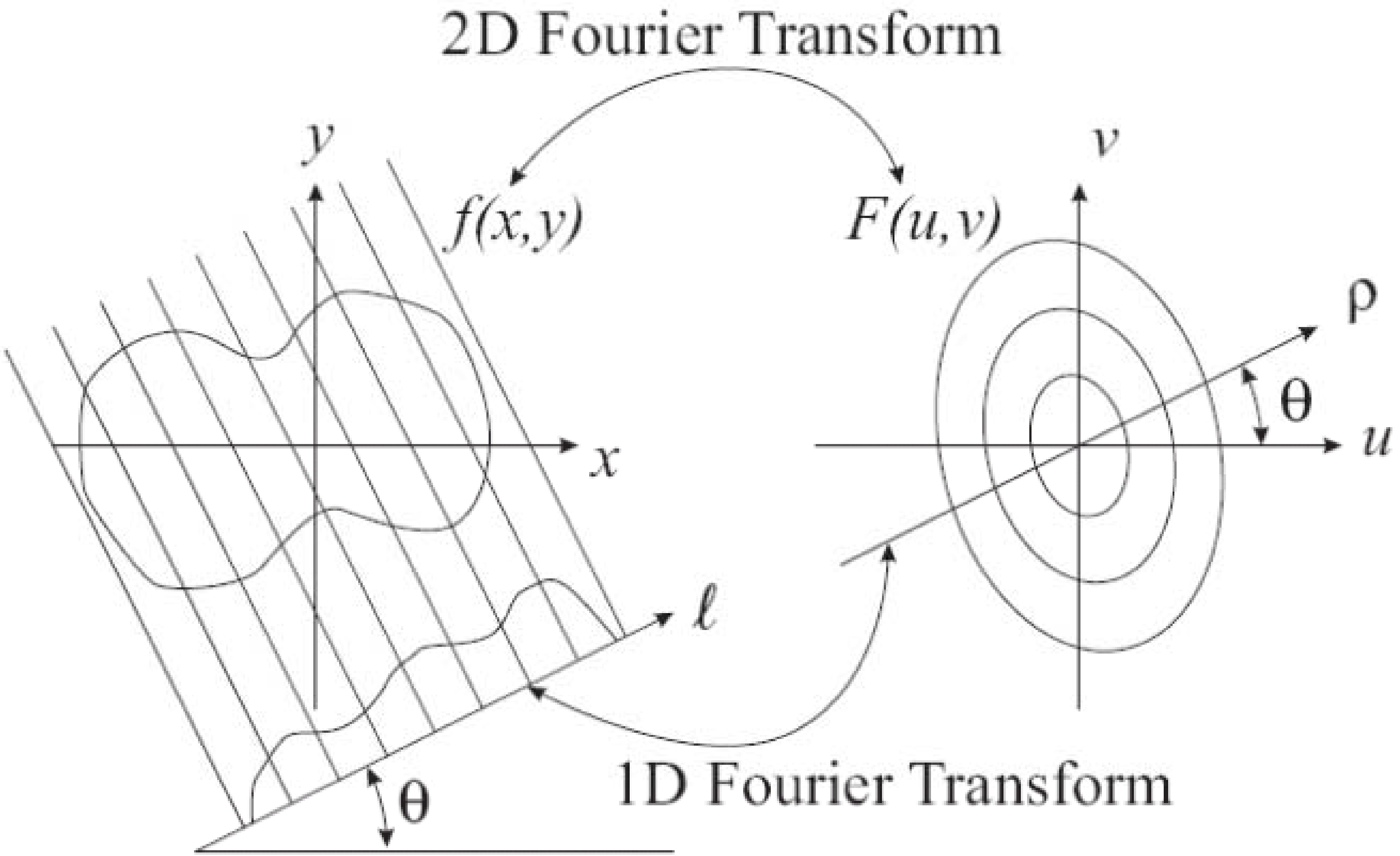
$$\begin{aligned} G(\varrho,\theta) &= \mathcal{F}_{\mathrm{1D}}\left\{g(\ell,\theta)\right\} \\ F(u,v) &= \mathcal{F}_{\mathrm{2D}}\left\{f(x,y)\right\} \end{aligned}$$

- Projection-slice theorem:

$$G(\varrho,\theta)=F(\varrho\cos\theta,\varrho\sin\theta)$$

The Fourier Transform of a projection at angle θ is a line in the Fourier transform of the image at the same angle.
If (l,θ) are sampled sufficiently dense, then from g (l,θ) we essentially know F(u,v) (on the polar coordinate), and by inverse transform can obtain f(x,y)!

Illustration of the Projection Slice Theorem



Proof

- Go through on the board
- Using the set form of the line integral
- See Prince&Links, P. 198

$$g(\ell,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \ell)\,dx dy$$

$$G(\rho,\theta) = \int_{-\infty}^{\infty} g(l,\theta)\exp\{-j2\pi\rho l\}dl$$

The Fourier Method

- The projection slice theorem leads to the following conceptually simple reconstruction method
  - Take 1D FT of each projection to obtain $G(\rho,\theta)$ for all θ
  - Convert $G(\rho,\theta)$ to Cartesian grid F(u,v)
  - Take inverse 2D FT to obtain f(x,y)
- Not used because
  - Difficult to interpolate polar data onto a Cartesian grid
  - Inverse 2D FT is time consuming
- But is important for conceptual understanding
  - Take inverse 2D FT on $G(\rho,\theta)$ on the polar coordinate leads to the widely used Filtered Backprojection algorithm

Filtered Backprojection

- Inverse 2D FT in Cartesian coordinate:

$$f(x,y)=\int\int F(u,v)e^{j2\pi(xu+yv)}dudv$$

- Inverse 2D FT in Polar coordinate:

$$f(x,y)=\int\limits_{0->2\pi}\int\limits_{0->\infty}\overset{=G(\rho,\theta)}{F(\rho\cos\theta,\rho\sin\theta)}e^{j2\pi\rho(\overset{=l}{x\cos\theta+y\sin\theta})}\rho d\rho d\theta$$

- Proof of filtered backprojection algorithm

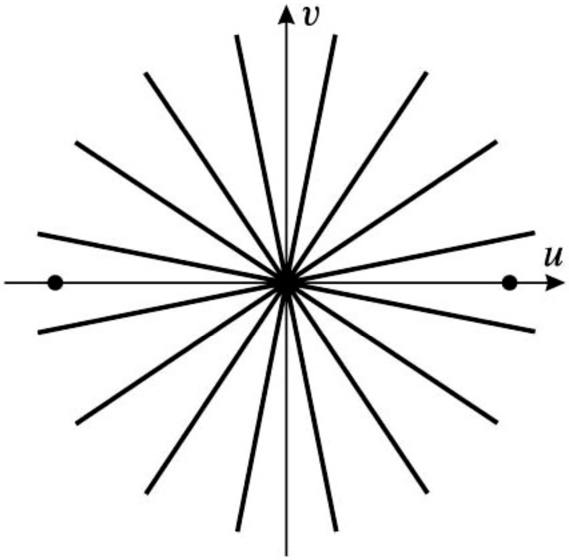
$$f(x,y)=\int_0^{\pi}\left[\overset{\text{Inverse FT}}{\int_{-\infty}^{\infty}|\varrho|G(\varrho,\theta)e^{+j2\pi\varrho\ell}d\varrho}\right]_{\ell=x\cos\theta+y\sin\theta}d\theta$$

Filtered Backprojection Algorithm

- Algorithm:
  - For each θ
    - Take 1D FT of $g(l,\theta)$ for each θ -> $G(\rho,\theta)$
    - Frequency domain filtering: $G(\rho,\theta)$ -> $Q(\rho,\theta)=|\rho|G(\rho,\theta)$
    - Take inverse 1D FT: $Q(\rho,\theta)$ -> $q(l,\theta)$
    - Backprojecting $q(l,\theta)$ to image domain -> $b_\theta(x,y)$
  - Sum of backprojected images for all θ

Function of the Ramp Filter

- Filter response:
  - $c(\rho) = |\rho|$
  - High pass filter
- $G(\rho,\theta)$ is more densely sampled when ρ is small, and vice verse
- The ramp filter compensate for the sparser sampling at higher ρ



Convolution Backprojection

- The Filtered backprojection method requires taking 2 Fourier transforms (forward and inverse) for each projection
- Instead of performing filtering in the FT domain, perform convolution in the spatial domain
- Assuming c(l) is the spatial domain filter
  - $|\rho|$ <-> c(l)
  - $|\rho|G(\rho,\theta)$ <-> $c(l) * g(l,\theta)$
- For each θ:
  - Convolve projection $g(l,\theta)$ with c(l): $q(l,\theta)= g(l,\theta) * c(l)$
  - Backprojecting $q(l,\theta)$ to image domain -> $b_\theta(x,y)$
  - Add $b_\theta(x,y)$ to the backprojection sum
- Much faster if c(l) is short
  - Used in most commercial CT scanners

- Correct reconstruction formula:

  $$f(x,y) = \int_0^{\pi} [c(\ell) * g(\ell,\theta)]_{\ell = x\cos\theta + y\sin\theta}\, d\theta$$

  where

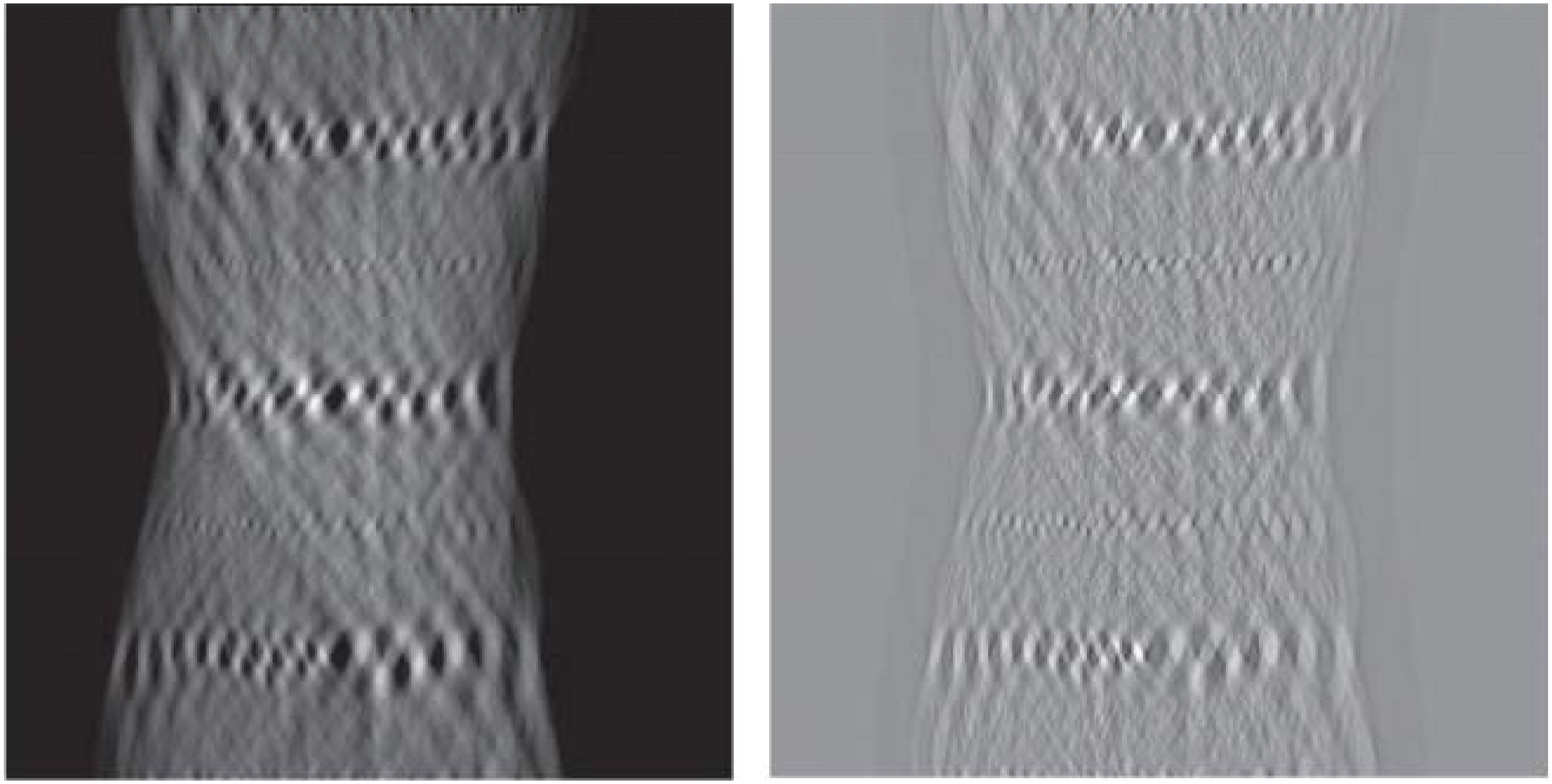
  $$c(\ell) = \mathcal{F}^{-1}\{|\varrho|\}$$

  is called the ramp filter.

- Three steps: ← know/understand these!!
  - 1. convolution
  - 2. backprojection
  - 3. summation

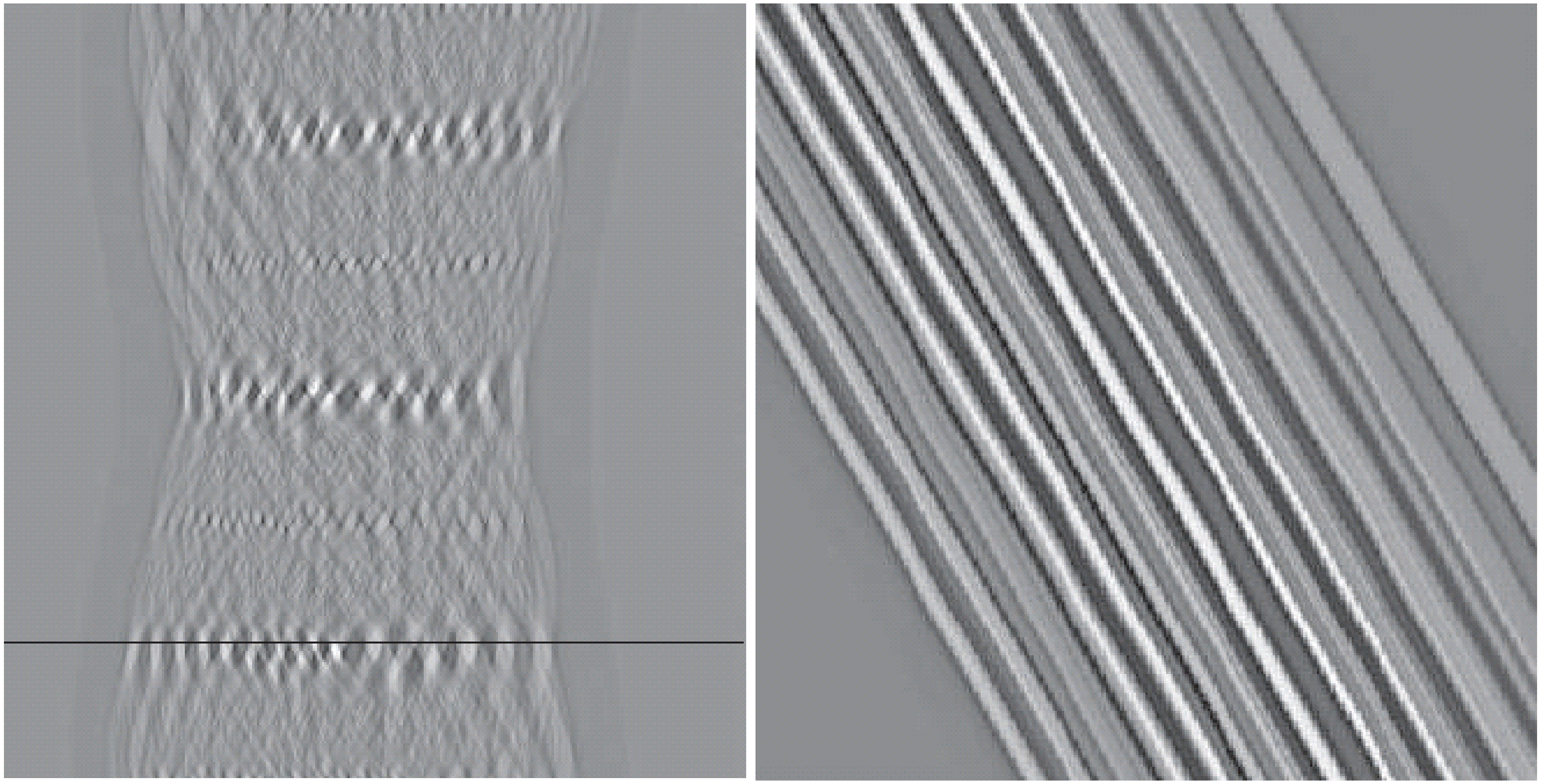
Step 1: Convolution

- Convolve every projection with $c(\ell)$
- the horizontal direction in a sinogram



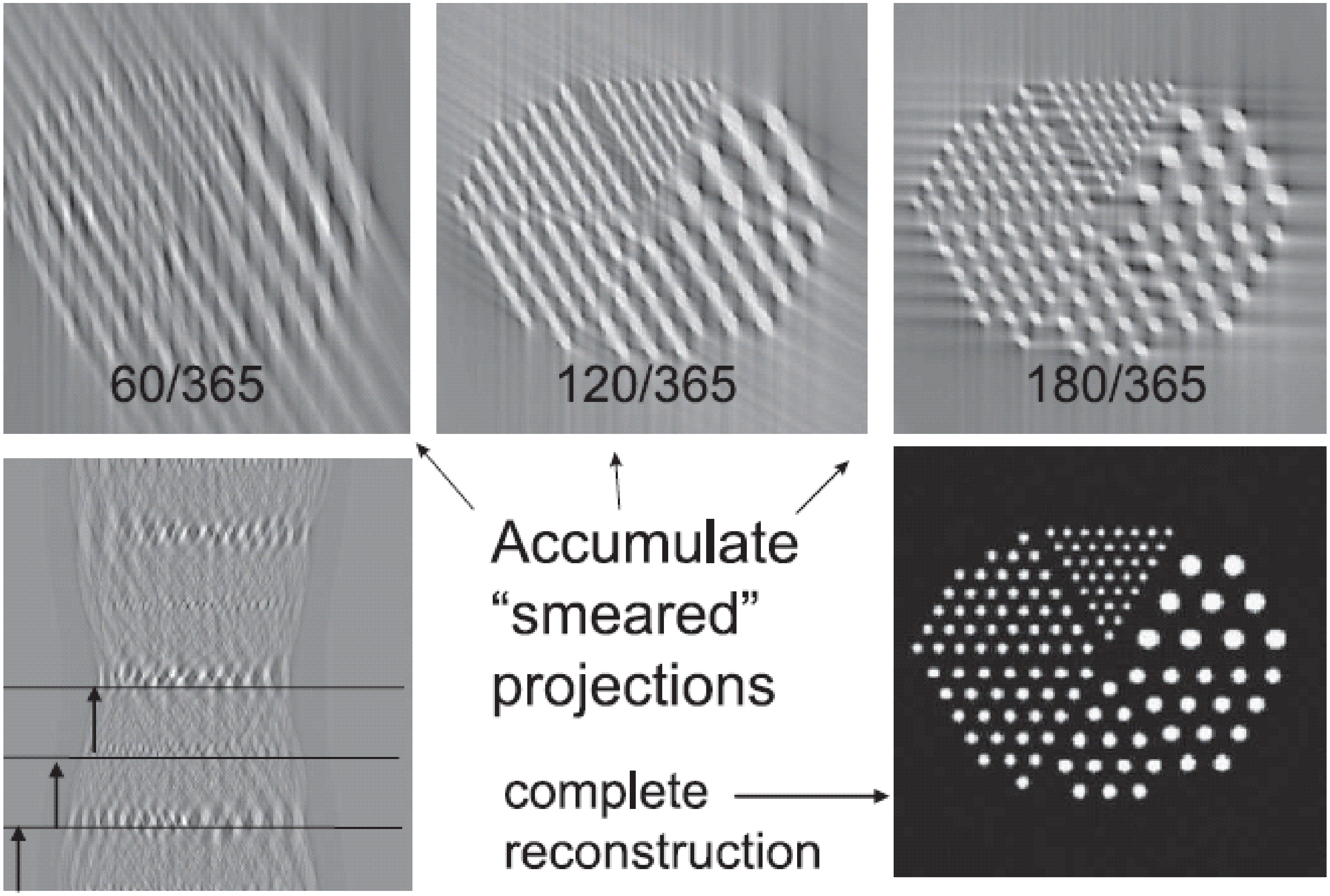
Step 2: Backprojection

- 1D projection → 2D laminar function



Step 3: Summation

- Accumulate sum of backprojection images



Accumulate “smeared” projections

complete reconstruction

Ramp Filter Design

- $|\varrho|$ is not integrable
- $\Rightarrow c(\ell)$ does not exist
- Actual ramp filter is designed as

$$\tilde{c}(\ell) = \mathcal{F}_{\mathrm{1D}}^{-1}\{W(\varrho)|\varrho|\}$$

- Simplest <u>window function</u> is

$$W(\varrho) = \mathrm{rect}\left(\frac{\varrho}{2\varrho_0}\right)$$

Practical Implementation

- Projections $g(l, \theta)$ are only measured at finite intervals
  - $l=n\tau$;
  - τ chosen based on maximum frequency in $G(\rho,\theta)$, W
    - $1/\tau >=2W$ or $\tau <=1/2W$ (Nyquist Sampling Theorem)
    - W can be estimated by the number of cycles/cm in the projection direction in the most detailed area in the slice to be scanned
- For filtered backprojection:
  - Fourier transform $G(\rho,\theta)$ is obtained via FFT using samples $g(n\tau, \theta)$
  - If N sample are available in g, 2N point FFT is taken by zero padding $g(n\tau, \theta)$
- For convolution backprojection
  - The ramp-filter is sampled at $l=n\tau$
  - Sampled Ram-Lak Filter

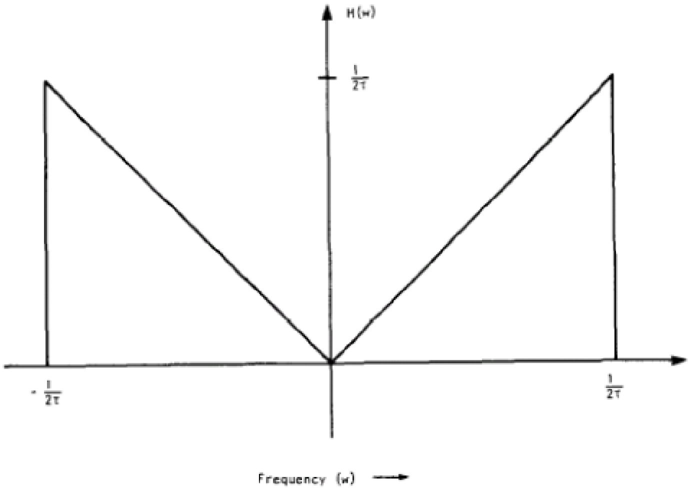
$$c(n)=\begin{cases} 1/4\tau^2; & n=0 \\ -1/(n\pi\tau)^2; & n=odd \\ 0; & n=even \end{cases}$$

The Ram-Lak Filter (from [Kak&Slaney])

$$H(w) = |w| b_w(w)$$

$$b_W(w) = \begin{cases} 1 & |w| < W \\ 0 & \text{otherwise.} \end{cases}$$

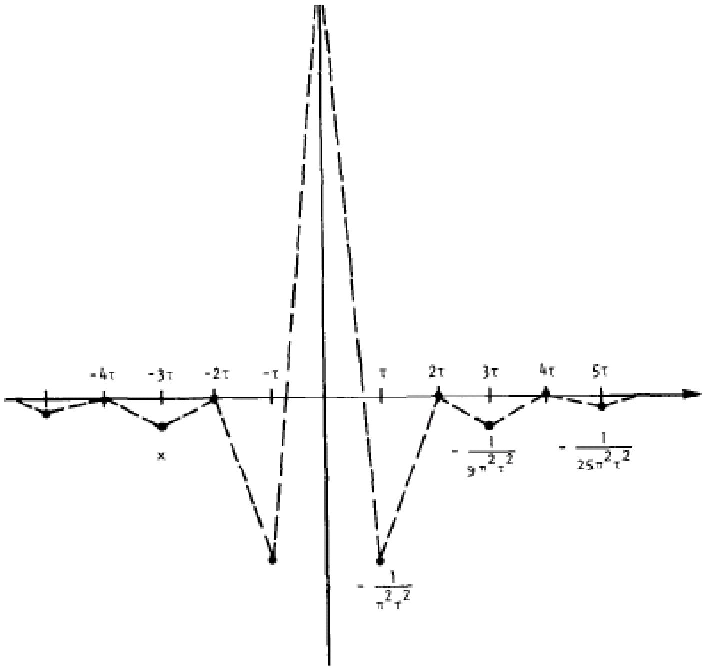
$$W = \frac{1}{2\tau} \text{ cycles/cm.}$$



$$h(t) = \int_{-\infty}^{\infty} H(w) e^{+j2\pi wt}\, dw$$

$$= \frac{1}{2\tau^2} \frac{\sin 2\pi t/2\tau}{2\pi t/2\tau} - \frac{1}{4\tau^2} \left(\frac{\sin \pi t/2\tau}{\pi t/2\tau} \right)^2$$

$$h(n\tau) = \begin{cases} 1/4\tau^2, & n = 0 \\ 0, & n \text{ even} \\ -\dfrac{1}{n^2\pi^2\tau^2}, & n \text{ odd.} \end{cases}$$



Common Filters

- Ram-Lak: using the rectangular window
- Shepp-Logan: using a sinc window
- Cosine: using a cosine window
- Hamming: using a generalized Hamming window
- See Fig. B.5 in A. Webb, Introduction to biomedical imaging

Matlab Implementation

- MATLAB (image toolbox) has several built-in functions:
  - `phantom`: create phantom images of size NxN

    ```
    I = PHANTOM(DEF,N) DEF= 'Shepp-Logan' ,' Modified Shepp-Logan'
    Can also construct your own phantom, or use an arbitrary image
    ```
  - `radon`: generate projection data from a phantom
    - Can specify sampling of θ

    ```
    R = RADON(I,THETA)
    The number of samples per projection angle = sqrt(2) N
    ```
  - `iradon`: reconstruct an image from measured projections
    - Uses the filtered backprojection method
    - Can choose different filters and different interpolation methods for performing backprojection

    ```
    [I,H]=IRADON(R,THETA,INTERPOLATION,FILTER,FREQUENCY_SCALING,OUTPUT_SIZE)
    ```
  - Use 'help radon' etc. to learn the specifics
  - Other useful command:
    - `imshow, imagesc, colormap`

Summary

- Different generations of CT machines:
  - Difference and pros and cons of each
- X-ray source and detector design
  - Require (close-to) monogenic x-ray source
- Relation between detector reading and absorption properties of the imaged slice
  - Line integral of absorption coefficients (Radon transform)
- Reconstruction methods
  - Backprojection summation
  - Fourier method (projection slice theorem)
  - Filtered backprojection
  - Convolution backprojection

Equivalent, but differ in computation

- Impact of number of projection angles on reconstruction image quality
- Matlab implementations

Reference

- Prince and Links, Medical Imaging Signals and Systems, Chap 6.
- Webb, Introduction to biomedical imaging, Appendix B.
- Kak and Slanley, Principles of Computerized Tomographic Imaging, IEEE Press, 1988. Chap. 3
  - Electronic copy available at http://www.slaney.org/pct/pct-toc.html

Homework

- Reading:
  - Prince and Links, Medical Imaging Signals and Systems, Chap 6, Sec.6.1-6.3.3
- Note down all the corrections for Ch. 6 on your copy of the textbook based on the provided errata.
- Problems for Chap 6 of the text book:
  - P6.5
  - Consider a 4x4 image that contains a diagonal line I=[0,0,0,1;0,0,1,0;0,1,0,0;1,0,0,0];
    - a) determine its projections in the directions: 0, 45,90,135 degrees.
    - b) determine the backprojected image from each projection;
    - c) determine the reconstructed images by using projections in the 0 and 90 degrees only.
    - d) determine the reconstructed images by using all projections. Comment on the difference from c).

Computer Assignment

Due: Two weeks from lecture date

1. Learn how do ‘phantom’.’radon’,’iradon’ work; summarize their functionalities. Type ‘demos’ on the command line, then select ‘toolbox -> image processing -> transform -> reconstructing an image from projection data’. Alternatively, you can use ‘help’ for each particular function.
2. Write a MATLAB program that 1) generate a phantom image (you can use a standard phantom provided by MATLAB or construct your own), 2) produce projections in a specified number of angle, 3) reconstruct the phantom using backprojection summation; Your program should allow the user to specify the number of projection angle. Run your program with different number of projections for the same view angle, and the different view angles, and compare the quality. You should NOT use the ‘radon( )’ and ‘iradon()’ function in MATLAB.
3. Repeat 1 but uses filter backprojection method for step 3). In addition to the number of projection angles, you should be able to specify the filter among several filters provided by Matlab and the interpolation filters used for backprojection. Compare the reconstructed image quality obtained with different filters and interpolation methods for the same view angle and number of projections. You can use the “iradon()” function in MATLAB
4. (Optional) Repeat 3 but uses convolution backprojection method. You have to do your own program.