EL-GY 6813 / BE-GY 6203 / G16.4426
Medical Imaging

Computed Tomography
(Part 1)

Jonathan Mamou and Yao Wang
Tandon School of Engineering
New York University, Brooklyn, NY 11201

Based on Prince and Links, Medical Imaging Signals and Systems and Lecture Notes by Prince. Figures are from the book.
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 angle spacing is sufficiently small and angular coverage is complete, 2D slices can be reconstructed successfully
1st Generation CT: Parallel Projections
2nd Generation
3G: Fan Beam

Much faster than 2G
Fast
Cannot use collimator at detector, hence affected by scattering
5G: Electron Beam CT (EBCT)

Stationary source and detector. Used for fast (cine) whole heart imaging. Source of x-ray moves around by steering an electron beam around X-ray tube anode.
6G: Helical (or spiral) CT

Entire abdomen or chest can be completed in 30 sec.
7G: Multislice

From http://www.kau.edu.sa/Files/0008512/Files/19500_2nd_presentation_final.pdf
Reduced scan time and increased Z-resolution (thin slices)
Most modern MSCT systems generates 64 slices per rotation, can image whole body (1.5 m) in 30 sec.
<table>
<thead>
<tr>
<th>Generation</th>
<th>Source</th>
<th>Source Collimation</th>
<th>Detector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>Single X-ray Tube</td>
<td>Pencil Beam</td>
<td>Single</td>
</tr>
<tr>
<td>2nd</td>
<td>Single X-ray Tube</td>
<td>Fan Beam (not enough to cover FOV)</td>
<td>Multiple</td>
</tr>
<tr>
<td>3rd</td>
<td>Single X-ray Tube</td>
<td>Fan Beam (enough to cover FOV)</td>
<td>Many</td>
</tr>
<tr>
<td>4th</td>
<td>Single X-ray Tube</td>
<td>Fan Beam covers FOV</td>
<td>Stationary Ring of Detectors</td>
</tr>
<tr>
<td>5th</td>
<td>Many tungsten anodes in single large tube</td>
<td>Fan Beam</td>
<td>Stationary Ring of Detectors</td>
</tr>
<tr>
<td>6th</td>
<td>3G/4G</td>
<td>3G/4G</td>
<td>3G/4G</td>
</tr>
<tr>
<td>7th</td>
<td>Single X-ray Tube</td>
<td>Cone Beam</td>
<td>Multiple array of detectors</td>
</tr>
</tbody>
</table>
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
Dual Energy CT

• Recall that linear attenuation coefficient depends on the X-ray energy
• Dual energy CT enables simultaneous acquisition to two images of the organ, reflecting their attenuation profiles corresponding to different energies
• The scanner has two x-ray tubes using two different voltage levels to generate the x-rays.
X-ray Detectors

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

Convert detected photons to lights

Convert light to electric current
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:

$$g_d = - \ln \frac{I_d}{I_0}$$

$$= \int_0^d \mu(s; \bar{E'}) ds$$

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

$I_0$ must be calibrated for each detector, measured in the absence of attenuating objects
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 \(\sim3,000\)

Requires at least 12 bits for representation (e.g., \(h=-1000\) for air)

Godfrey Hounsfield, together with Alan Cormack, invented X-ray CT!
CT numbers of Different Tissues at 70KeV

[Smith & Webb]

<table>
<thead>
<tr>
<th>Tissue</th>
<th>CT number (Hounsfield units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bone</td>
<td>1000–3000</td>
</tr>
<tr>
<td>Muscle</td>
<td>10–40</td>
</tr>
<tr>
<td>Water</td>
<td>0</td>
</tr>
<tr>
<td>Lipid</td>
<td>−50 to −100</td>
</tr>
<tr>
<td>Air</td>
<td>−1000</td>
</tr>
<tr>
<td>Brain (white matter)</td>
<td>20 to 30</td>
</tr>
<tr>
<td>Brain (grey matter)</td>
<td>35 to 45</td>
</tr>
<tr>
<td>Blood</td>
<td>40</td>
</tr>
</tbody>
</table>
Parameterization of a Line

Each projection line $L$ is defined by $(l, \theta)$

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

Option 1 (parameterized by $s$):

$$
\begin{align*}
x(s) &= l \cos \theta - s \sin \theta \\
y(s) &= l \sin \theta + s \cos \theta
\end{align*}
$$

Option 2:

$$
x \cos \theta + y \sin \theta = 1
$$
Line Integral: parametric form

- What is integral of $f(x, y)$ on $L(\ell, \theta)$?
- Step 1: Parameterize $L(\ell, \theta)$:
  \[
  x(s) = \ell \cos \theta - s \sin \theta \\
  y(s) = \ell \sin \theta + s \cos \theta 
  \]
- 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 $L(\ell, \theta)$

$$g(\ell, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \ell) \, dx \, dy$$
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:
  \[ f(x, y) = \mu(x, y; \bar{E}) \]
  \[ g(\ell, \theta) = - \ln \frac{I_d}{I_0} \]
What is $g(l, \theta)$?

- Fix $l$ and $\theta$: line integral of $f(x, y)$
- Fix $\theta$: projection of $f(x, y)$ at angle $\theta$
- Function of $\theta$ and $l$:
  
  $$g(l, \theta) \text{ is the Radon transform of } f(x, y)$$

$$g(l, \theta) = \mathcal{R}\{f(x, y)\}$$
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 $\ell$ and $\theta$
- CT scanners acquire a sinogram

Bottom row: $\theta = 0$ (vertical proj), middle $\theta = \pi/2$ (horizontal proj), top $\theta = \pi - \Delta \theta$
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 \((l, \theta)\) the projected value \(g(l, \theta)\), repeat for all \(l\) for the given \(\theta\)
$b_\theta(x, y)$ is a **laminar image**

\[ b_\theta(x, y) = g(x \cos \theta + y \sin \theta, \theta) \]
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.
Two Ways of Performing Backprojection

- Option 1: assigning value of $g(l, \theta)$ to all points on the line $(l, \theta)$
  - $g(l, \theta)$ is only measured at certain $l$: $l_n = n \Delta l$
  - If $l$ is coarsely sampled ($\Delta l$ 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: For each $\theta$, go through all sampling points $(x, y)$ in an image, find its corresponding “$l = x \cos \theta + y \sin \theta$”, take the $g$ value for $(l, \theta)$
  - $g(l, \theta)$ is only measured at certain $l$: $l_n = n \Delta l$
  - must interpolate $g(l, \theta)$ for any $l$ from given $g(l_n, \theta)$

- 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:

\[ f_b(x, y) = \int_0^{\pi} b_\theta(x, y) d\theta \]

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

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

• \( f_b(x, y) \) 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 $g(s_m, \phi_n)$, $m=1, ..., M$ for a given $\phi_n$.

Back-projection in MATLAB:

```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
```

g(:,phi) stores the projection data at angle phi corresponding to excentricities stored in s_n.

Implementation: Projection

- Creating projection data using computers will yield similar problems. Possible \( l \) and \( \theta \) are both quantized. If you first specify \((l, \theta)\), then find \((x,y)\) that are on this line. It is not easy. Instead, for given \( \theta \), 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)
  - \( f(x,y) \) stores the original image data
  - \( G(l,\phi) \) stores projection data, \( q_l \) is the desired quantization stepsize for \( l \).

\[
\begin{align*}
N &= \text{ceil}(\sqrt{I^2+J^2})+1; \quad \%(\text{assume image size } I \times J), \quad N_0 \text{ is maximum lateral distance} \\
N_0 &= \text{floor}((N-1)/2); \\
q_l &= 1; \\
G &= \text{zeros}(N,180); \\
\text{for phi=0:179} \\
& \quad \text{for } (x=-J/2:J/2-1; \ y=-I/2:I/2-1) \\
& \quad \quad l = x \cos(\phi \pi/180) + y \sin(\phi \pi/180); \\
& \quad \quad l = \text{round}(l/q_l)+N_0+1; \\
& \quad \quad \text{if } (l>1) \&\& (l<N) \\
& \quad \quad \quad G(l,\phi+1) = G(l,\phi+1) + f(x+J/2+1,y+I/2+1); \\
& \quad \quad \text{End} \\
& \quad \text{end} \\
& \text{end}
\end{align*}
\]
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:
  \[ g(\ell, \theta) = \mathcal{R} \{ f(x, y) \} \]

- Fourier transforms:
  \[ G(\rho, \theta) = \mathcal{F}_{1D} \{ g(\ell, \theta) \} \]
  \[ F(u, v) = \mathcal{F}_{2D} \{ f(x, y) \} \]

- Projection-slice theorem:
  \[ G(\rho, \theta) = F(\rho \cos \theta, \rho \sin \theta) \]

The 1D Fourier Transform of a projection at angle \( \theta \) is a line in the 2D Fourier transform of the image at the same angle passing through the origin.

If \( (\ell, \theta) \) are sampled sufficiently dense, then from \( g(\ell, \theta) \) we essentially know \( F(u, v) \) (on the polar coordinate), and by inverse transform we can obtain \( f(x, y) \)!
Illustration of the Projection Slice Theorem

2D Fourier Transform

\[ f(x,y) \]

1D Fourier Transform

\[ F(u,v) \]
Proof

- Go through on the board
- Using the set form of the line integral
- See Prince&Links, P. 203

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

\[ G(\rho, \theta) = \int_{-\infty}^{\infty} g(\ell, \theta) \exp\{-j2\pi\rho\ell\} \, d\ell \]
The Fourier Method

- The projection slice theorem leads to the following conceptually simple reconstruction method
  - Take 1D FT of each projection $g(l, \theta)$ to obtain $G(\rho, \theta)$ for all $\theta$
  - 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 of $G(\rho, \theta)$ in the polar coordinate grid 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:
  \[ f(x, y) = \int_{0-2\pi}^{0->\infty} \int_{0->\infty} F(\rho \cos \theta, \rho \sin \theta)e^{j2\pi \rho (\cos \theta + y \sin \theta)} \rho d\rho d\theta \]

- Proof of filtered backprojection algorithm

\[ f(x, y) = \int_{0}^{\pi} \left[ \int_{-\infty}^{\infty} |\rho|G(\rho, \theta)e^{j2\pi \rho \ell} d\rho \right]_{\ell=x \cos \theta + y \sin \theta} d\theta \]

Where the fact that \( g(l, \theta) = g(-l, \theta+\pi) \) was used
Filtered Backprojection Algorithm

- Algorithm:
  - For each $\theta$
    - Take 1D FT of $g(l, \theta)$ for each $\theta$ -> $G(\rho, \theta)$
    - 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 $\theta$
Function of the Ramp Filter

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

- The Filtered backprojection method requires taking two 1D 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:
  - $\text{FT}(|\rho|)$ is $c(l)$
  - $\text{FT}(|\rho|G(\rho, \theta))$ is $c(l) * g(l, \theta)$
- For each $\rho$:
  - 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:

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

where

\[ c(\ell) = \mathcal{F}^{-1}\{\lambda\} \]

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 $\rightarrow$ 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

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

- Simplest window function is

$$W(\varrho) = \text{rect}\left(\frac{\varrho}{2\varrho_0}\right)$$
The Ram-Lak Filter (from [Kak&Slaney])

\[ H(w) = |w| b_w(w) \]

\[ b_w(w) = \begin{cases} 
1 & |w| < W \\
0 & \text{otherwise.} 
\end{cases} \]

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

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

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

\[ h(n\tau) = \begin{cases} 
1/4\tau^2, & n = 0 \\
0, & n \text{ even} \\
-\frac{1}{n^2\pi^2\tau^2}, & n \text{ odd.} 
\end{cases} \]
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
Practical Implementation

- Projections $g(l, \theta)$ are only measured at finite intervals
  - $l = n\tau$;
  - $\tau$ chosen based on maximum frequency in $G(\rho, \theta)$, $W$
    - $1/\tau \geq 2W$ or $\tau \leq 1/2W$ (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
    $$c(n) = \begin{cases} 
    1/4\tau^2; & n = 0 \\
    -1/(n\pi\tau)^2; & n = odd \\
    0; & n = even
    \end{cases}$$
Matlab Implementation

MATLAB (image toolbox) has several built-in functions:

- **phantom**: create phantom images of size NxN
  
  \[
  I = \text{PHANTOM(DEF,N)} \quad \text{DEF} = \text{‘Shepp-Logan’, ‘Modified Shepp-Logan’}
  \]
  
  Can also construct your own phantom, or use an arbitrary image, use `imread()`

- **radon**: generate projection data from a phantom
  
  - Can specify sampling of $\theta$
  
  \[
  R = \text{RADON}(I, \text{THETA})
  \]
  
  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

  \[
  [I, H] = \text{IRADON}(R, \text{THETA, INTERPOLATION, FILTER, FREQUENCY_SCALING, OUTPUT_SIZE})
  \]

  Interpolation filters: used for backprojection
  
  - 'nearest' - nearest neighbor interpolation,
  
  - 'linear' - linear interpolation (default),
  
  - 'spline' - spline interpolation
  

  - Use ‘help radon’ etc. to learn the specifics

- Other useful command:
  
  - `imshow`, `imagesc`, `colormap`
Top: 180 projections  
Middle: 90 projections  
Bottom: 45 projections

MATLAB code:

```matlab
>> I=imread('lena.jpg');
>> r180=radon(I,0:179);  
% 0:2:179;
>> p180=iradon(r180,0:179,'spline','cosine'
>> subplot(1,3,1),imagesc(I);
>> subplot(1,3,2),imagesc(r180);
>> subplot(1,3,3),imagesc(p180);
>> truesize
>> colormap(gray)
```
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
- 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.
• Good description of different generations of CT machines
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:
  1. P6.5
  2. P6.14 (Note that you can leave the solution of part (a) in the form of the convolution of two functions without deriving the actual convolution. For part (b), you can describe the solution using “intuition.”)
  3. P6.17
  4. The added problem in the following page
Added problem

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).

Note that for back-projection, you should normalize the backprojection in each direction based on the total length of projection along that direction in the square area.
Computer Assignment
Due: 10 am on Friday October 13

Will send assignment via NYU classes today.