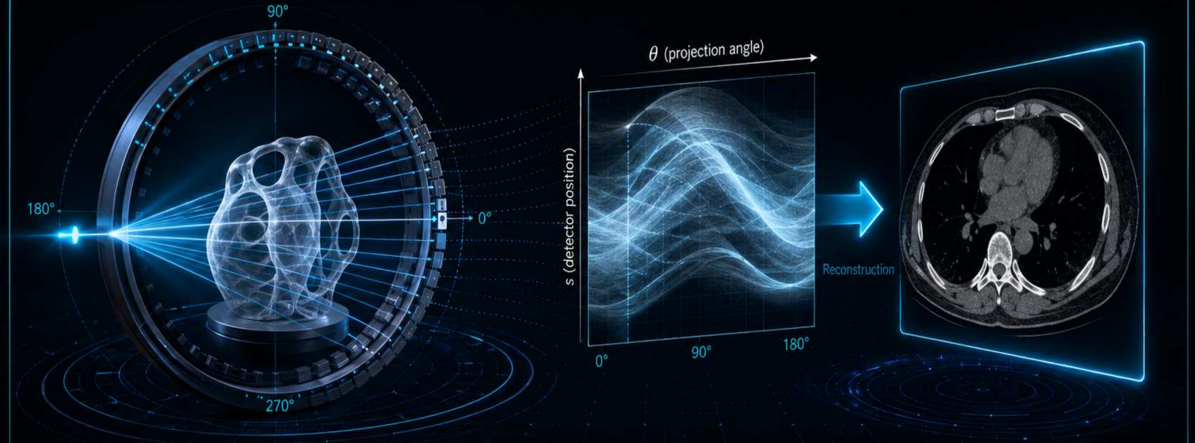
Chapter: Tomographic Imaging

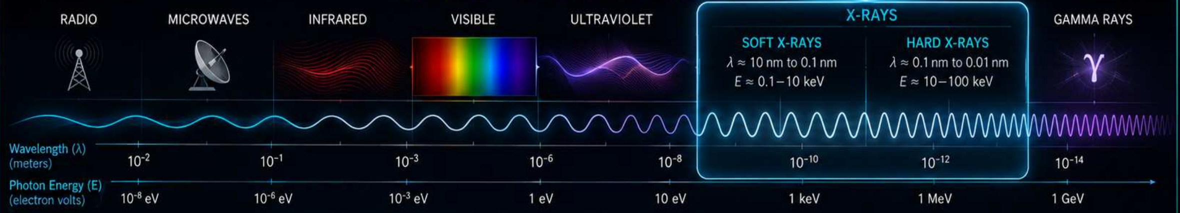
From projection measurements to cross-sectional reconstruction



X-Ray Wavelengths: Soft and Hard Regimes

Where x-rays sit in the electromagnetic spectrum, and why wavelength determines penetration and application

THE ELECTROMAGNETIC SPECTRUM



Shorter wavelength → higher photon energy → stronger penetration.

SOFT X-RAYS

- Better for high contrast in thin specimens and microscopy
- Surface / near-surface analysis
- Mammography, soft-tissue contrast
- X-ray microscopy
- EUV / laboratory science

HARD X-RAYS

- Better penetration for radiography and CT
- Bones / chest imaging
- Baggage / security scanning
- Industrial nondestructive testing
- Crystallography
- Thick-object imaging

Soft x-rays favor *contrast* and *shallow penetration*; hard x-rays favor *penetration, tomography, and inspection.*

Soft X-Rays: Contrast for Thin Specimens

Lower-energy x-rays are strongly absorbed and are best for microscopy, surface science, and thin biological samples.

Soft x-rays:
$\lambda \approx 10$ nm to 0.1 nm
$E \approx 0.1\text{–}10$ keV

ENERGY–WAVELENGTH RELATION

$$E = \frac{hc}{\lambda}$$

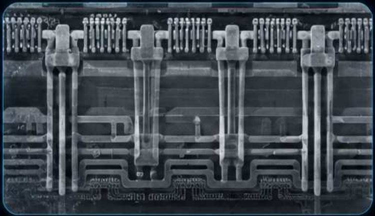
$h = 6.626 \times 10^{-34}$ J s
$c = 2.998 \times 10^{8}$ m/s
λ = wavelength (m)
E = photon energy (J)

- Longer wavelength than hard x-rays
- Lower photon energy
- Shallower penetration
- Stronger absorption contrast in thin samples

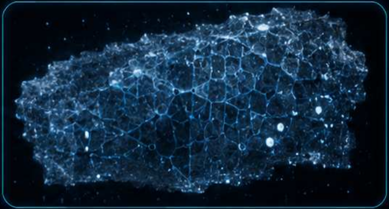
TYPICAL APPLICATIONS

- X-ray microscopy
- Soft x-ray tomography of cells
- Surface / near-surface analysis
- Materials science
- EUV / laboratory science

SOFT X-RAY MICROSCOPY
Biological cell (phase / absorption contrast)



MICROELECTRONIC / NANOSTRUCTURE IMAGING
Thin device cross-section (absorption contrast)



ELEMENTAL / SURFACE CONTRAST
Thin specimen (chemical contrast)

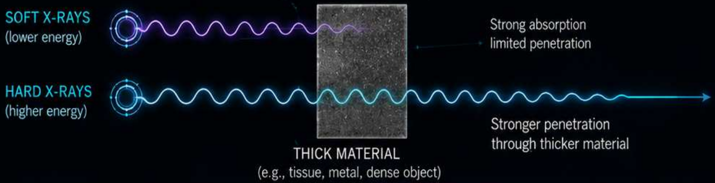
Soft x-rays deliver *excellent contrast in thin specimens, but their limited penetration makes them unsuitable for thick objects.*

Hard X-Rays: Penetration for Radiography and CT

Higher-energy x-rays travel through thicker objects and enable medical imaging, security screening, and industrial inspection.

Hard x-rays:
$\lambda \approx 0.1$ nm to 0.01 nm
$E \approx 10–100$ keV

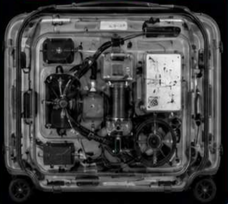
PENETRATION CONCEPT



RADIOGRAPHY (X-RAY)

COMPUTED TOMOGRAPHY (CT)

SECURITY / INDUSTRIAL X-RAY



APPLICATIONS

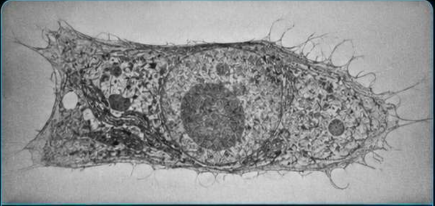
- Radiography (projection imaging)
- Fluoroscopy (real-time imaging)
- Computed Tomography (CT)
- Baggage / Security Scanning
- Industrial Nondestructive Testing (NDT)
- Crystallography / Diffraction

Shorter wavelength ⇒ higher energy ⇒ better penetration

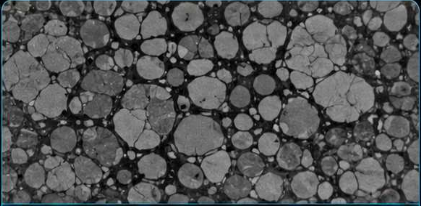
Hard x-rays are the workhorse of tomography and inspection because they *penetrate dense or thick objects* while *preserving useful structural contrast.*

X-Ray Example Imagery Across Applications

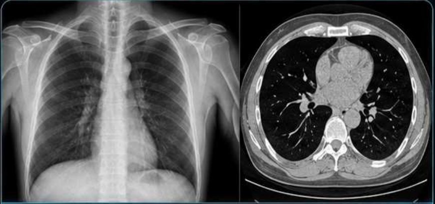
Representative images reveal how wavelength and photon energy shape what x-rays can see.



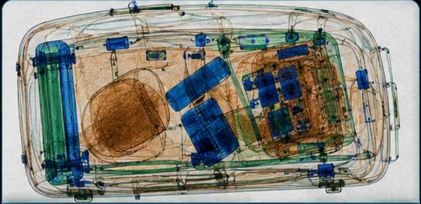
1 Soft X-ray Microscopy — Thin Cell Specimen — SOFT X-RAY



2 Soft X-ray Imaging — Materials / Microstructure — SOFT X-RAY



3 Hard X-ray Medical — Radiograph / CT — HARD X-RAY



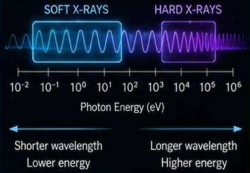
4 Hard X-ray Industrial — Baggage Inspection — HARD X-RAY

AT A GLANCE

Soft x-rays
→ thin samples, high absorption contrast

Hard x-rays
→ thick samples, deep penetration

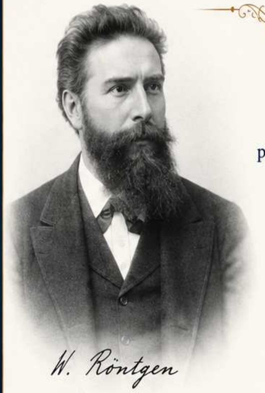
THE X-RAY SPECTRUM



Choosing the x-ray regime is a trade-off between contrast and penetration, and that choice determines the right application.

HISTORICAL STORIES

EARLY AND UNEXPECTED USES OF X-RAYS



Discovery of X-rays, 1895

From curious demonstrations to practical applications, X-rays quickly moved from laboratory marvel to everyday impact.

1 THE RÖNTGEN RING STORY

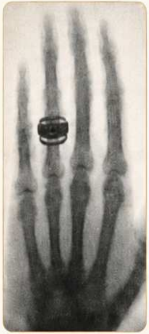
The first X-ray image ever taken—of love and curiosity.

On December 22, 1895, Wilhelm Conrad Röntgen made the first X-ray image in history. He used his wife's hand and a ring she wore as the subject.

What Happened?

Röntgen noticed a bony structure and a dark shadow where the metal ring blocked the rays. He was amazed that he could "see" inside the body without surgery.

Bertha Röntgen (his wife)
Wore the ring that made history.



The first X-ray image (1895)
Bertha Röntgen's hand with ring

Why It Matters

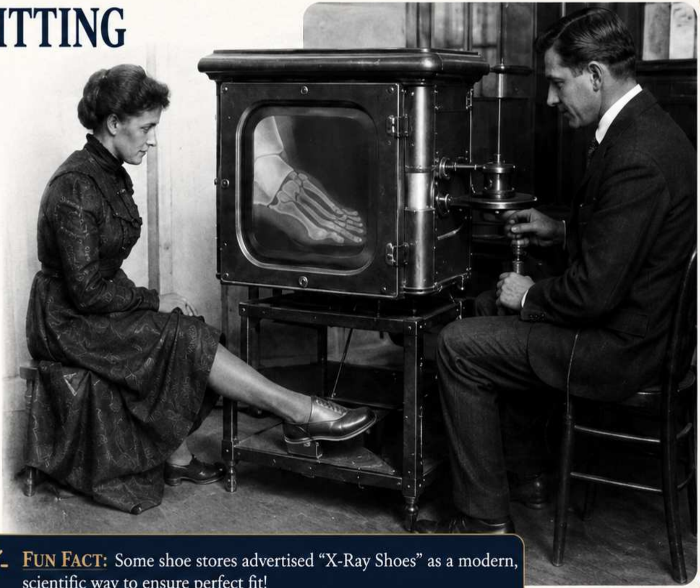
This simple experiment proved the extraordinary power of X-rays and opened the door to medicine, science, and countless technological advances.

> *"Great discoveries often begin with simple questions and bold curiosity."*

X-Rays for Shoe Fitting

A window inside the shoe.

In the early 1900s, shoe stores in the United States began using X-ray fluoroscopes to see how well shoes fit customers—without removing their shoes!



How it Worked

- The customer placed their foot in a special X-ray shoe-fitting device.
- The salesperson could see the bones and shoe alignment in real time on a fluorescent screen.
- Adjustments were made for comfort and proper fit.

Fun Fact: Some shoe stores advertised "X-Ray Shoes" as a modern, scientific way to ensure perfect fit!

Legacy of Early Innovation

Curiosity led to compassion.

These early stories show how X-rays moved beyond the laboratory and imagination into real-world applications—transforming healthcare, industry, and everyday life. What began as a glimpse into the unseen became a foundation for technologies we rely on today.

Röntgen's Ring
(1895)

Shoe Fitting
(1900s)

Medical Imaging
(Today)

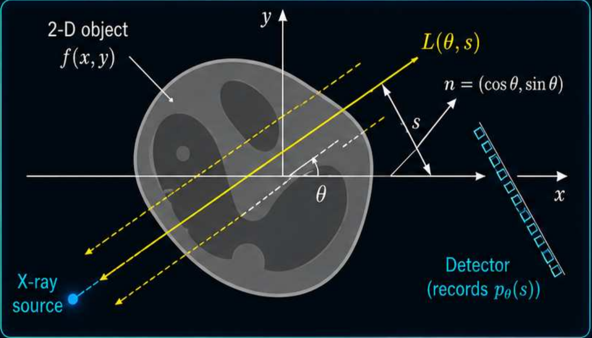
Security Screening
(Today)

The Takeaway

From a wife's ring to the inside of a shoe, X-rays revealed the unseen. These surprising early uses paved the way for life-saving medical imaging and countless modern technologies.

The Radon Transform: Geometry of Tomographic Projections

Parallel-beam CT measures line integrals of the object from many viewing angles.

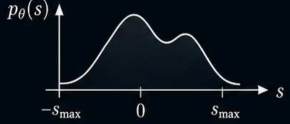


$$p_\theta(s) = \mathcal{R}\{f\}(\theta, s) = \int_{L(\theta,s)} f(x,y)\, d\ell$$

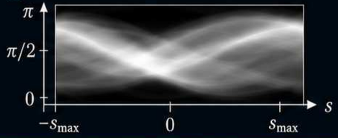
- $f(x,y)$ = attenuation map (object function) in image space
- θ = projection angle (angle of the beam direction)
- s = signed perpendicular distance from origin to the line (positive in the direction of n)
- $L(\theta, s)$ = the line of integration (defined by θ *and* s)
- $d\ell$ = differential length element along $L(\theta, s)$
- $p_\theta(s)$ = 1-D projection measured by the detector at angle θ

1. Object $f(x,y)$

2. One projection $p_\theta(s)$ at angle θ



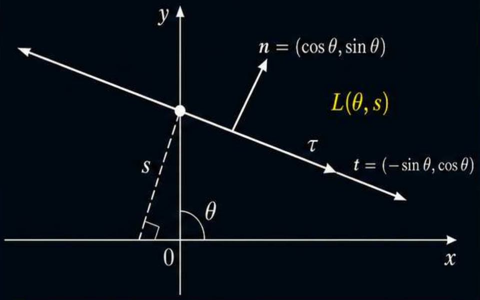
3. Collection of projections over many angles



Takeaway: *A tomographic projection is a line integral through the object.*

Parameterizing a Line and Writing the Radon Transform

A line is specified by its angle θ and signed distance s, then integrated using a running coordinate τ.



Coordinate parameterization of the line $L(\theta, s)$

$$x = s\cos\theta - \tau\sin\theta$$

$$y = s\sin\theta + \tau\cos\theta$$

1) Geometric (line integral) form

$$p_\theta(s) = \mathcal{R}f(\theta, s) = \int_{-\infty}^{\infty} f(s\cos\theta - \tau\sin\theta,\, s\sin\theta + \tau\cos\theta)\, d\tau$$

Integrates the object $f(x, y)$ directly along the line $L(\theta, s)$ by sweeping the running coordinate $\tau \in (-\infty, \infty)$.

2) Delta-function (constraint) form

$$\mathcal{R}f(\theta, s) = \iint_{\mathbb{R}^2} f(x, y)\, \delta(s - x\cos\theta - y\sin\theta)\, dx\, dy$$

Uses a Dirac delta to enforce the line constraint $x\cos\theta + y\sin\theta = s$ and integrates over the entire image plane.

Definitions

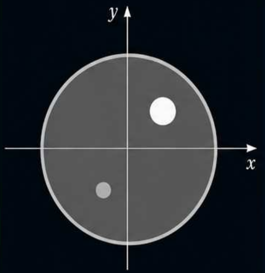
- $\theta \in [0, \pi)$: angle of the line's normal vector $\boldsymbol{n} = (\cos\theta, \sin\theta)$.
- $s \in \mathbb{R}$: signed perpendicular distance from the origin to the line $L(\theta, s)$.
- $\tau \in (-\infty, \infty)$: running coordinate along the line in the direction $\boldsymbol{t} = (-\sin\theta, \cos\theta)$.
- $\delta(\cdot)$: Dirac delta function.
- $p_\theta(s) = \mathcal{R}f(\theta, s)$: Radon transform value (projection) at angle θ and distance s.

Takeaway: *The Radon transform can be expressed either geometrically or with a delta-function constraint.*

From Individual Projections to a Sinogram

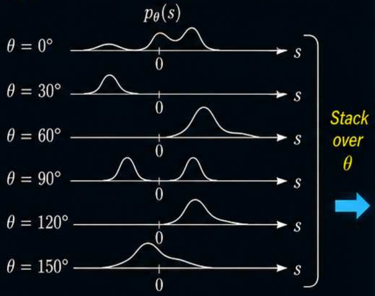
Each angle θ produces a 1-D projection; stacking them over θ creates the measurement image $p(\theta, s)$.

① Object $f(x, y)$

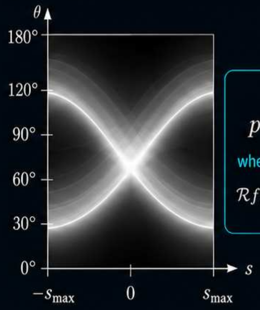


Unknown 2-D attenuation distribution.

② 1-D projections $p_\theta(s)$ at sample angles



③ Sinogram $p(\theta, s)$



Sinogram definition

$$p(\theta, s) = \mathcal{R}f(\theta, s)$$

where

$$\mathcal{R}f(\theta, s) = \int_{L(\theta, s)} f(x, y)\, d\ell$$

What the sinogram encodes

- Each row (fixed θ) is one 1-D projection $p_\theta(s)$.
- Horizontal axis s is the detector position (signed distance from origin).
- Brightness indicates cumulative attenuation along the corresponding line.

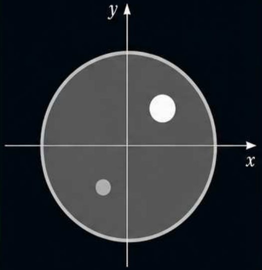
Intensity meaning

Takeaway: *A sinogram is the structured collection of all 1-D projections.*

From Individual Projections to a Sinogram

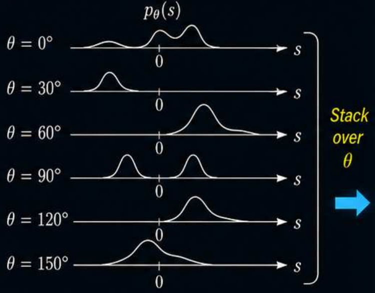
Each angle θ produces a 1-D projection; stacking them over θ creates the measurement image $p(\theta, s)$.

1 Object $f(x, y)$

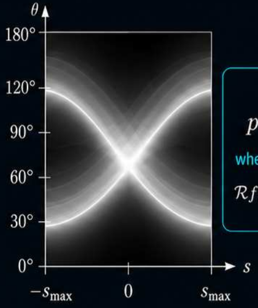


Unknown 2-D attenuation distribution.

2 1-D projections $p_\theta(s)$ at sample angles



3 Sinogram $p(\theta, s)$



Sinogram definition

$$p(\theta, s) = \mathcal{R}f(\theta, s)$$

where

$$\mathcal{R}f(\theta, s) = \int_{L(\theta, s)} f(x, y)\, d\ell$$

What the sinogram encodes

- Each row (fixed θ) is one 1-D projection $p_\theta(s)$.
- Horizontal axis s is the detector position (signed distance from origin).
- Brightness indicates cumulative attenuation along the corresponding line.

Intensity meaning

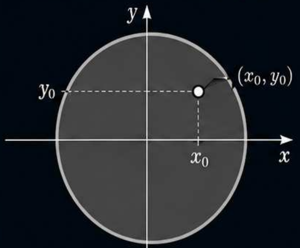
Takeaway: *A sinogram is the structured collection of all 1-D projections.*

How Image Features Appear in a Sinogram

Points, edges, and objects leave recognizable signatures in Radon space.

① Point object in the image

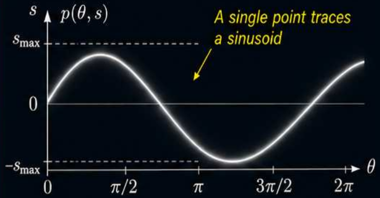
$$f(x,y) = \delta(x - x_0, y - y_0)$$



A single point at (x_0, y_0).

② Sinusoidal trace in the sinogram

$$\mathcal{R}f(\theta, s) = \delta(s - x_0 \cos\theta - y_0 \sin\theta)$$

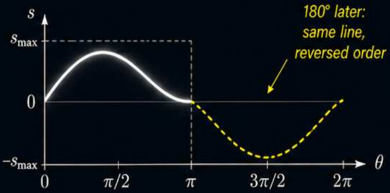


The point (x_0, y_0) traces

$$s(\theta) = x_0 \cos\theta + y_0 \sin\theta.$$

③ π-periodic symmetry

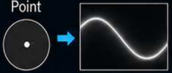
$$p(\theta + \pi, s) = p(\theta, -s)$$



Projections 180° apart correspond to the same set of lines in the object, but the detector order is reversed.

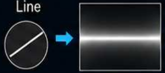
④ Signatures of common features

Point



Thin sinusoid (δ-like)

Line



Horizontal band

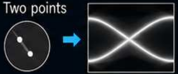
Edge

Sine-like wave (changes sign)

Disk (filled)

Thick arc (wide band)

Two points



Two sinusoids (crossing)

Takeaway: *Interpreting a sinogram means recognizing how geometric features map into curves and patterns.*

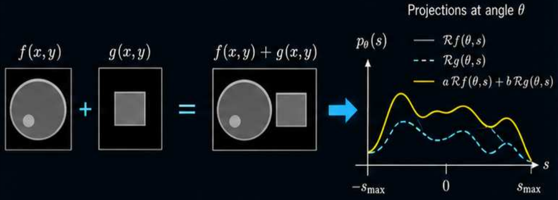
Important Properties of the Radon Transform I

Linearity and translation show how image-space operations map into projection space.

① *Linearity*

$$\mathcal{R}\{af + bg\}(\theta, s) = a\,\mathcal{R}f(\theta, s) + b\,\mathcal{R}g(\theta, s)$$

- a, b are scalars.
- $f(x,y)$, $g(x,y)$ are images (objects).
- Because integration is linear, projections add and scale exactly.

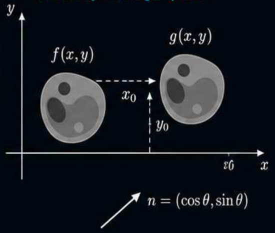


② *Translation*

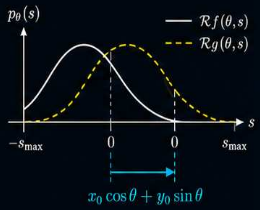
If $g(x,y) = f(x - x_0, y - y_0)$, then

$$\mathcal{R}g(\theta, s) = \mathcal{R}f(\theta, s - x_0\cos\theta - y_0\sin\theta)$$

Geometry in image space



Effect in projection space



The amount of shift in s depends on the *component* of the image translation along the detector *normal* $n = (\cos\theta, \sin\theta)$.

Takeaway: *Linearity preserves sums; translations in the image become shifts in detector coordinate s.*

Important Properties of the Radon Transform II

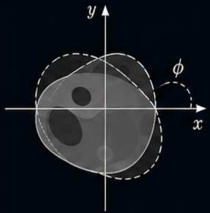
Rotation and simple canonical examples build intuition for later reconstruction methods.

① *Rotation property*

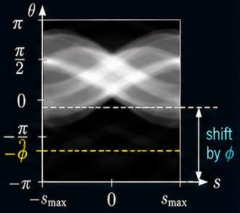
If $g(x,y) = f\left(\mathcal{R}_{-\phi}\begin{bmatrix} x \\ y \end{bmatrix}\right)$, then

$$\boxed{\mathcal{R}_g(\theta, s) = \mathcal{R}_f(\theta - \phi, s)}$$

Image space
(object rotated by ϕ)



Sinogram
(shifted along θ by ϕ)

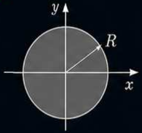


$\mathcal{R}_{-\phi}$ = rotation by angle $-\phi$ (clockwise) in the image plane

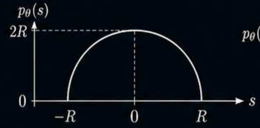
$$\mathcal{R}_{-\phi} = \begin{bmatrix} \cos(-\phi) & -\sin(-\phi) \\ \sin(-\phi) & \cos(-\phi) \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

② *Canonical examples*

① Uniform disk of radius R



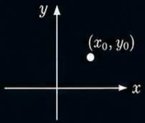
Projection (independent of θ):



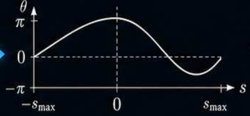
$$p_\theta(s) = \begin{cases} 2\sqrt{R^2 - s^2}, & |s| \le R \\ 0, & \text{otherwise} \end{cases}$$

Why it matters: Sets the scale of chord lengths; appears in many analytic formulas and is used to normalize and sanity-check tomographic data.

② Point object at (x_0, y_0)



Sinusoidal trace:



$$s(\theta) = x_0\cos\theta + y_0\sin\theta$$

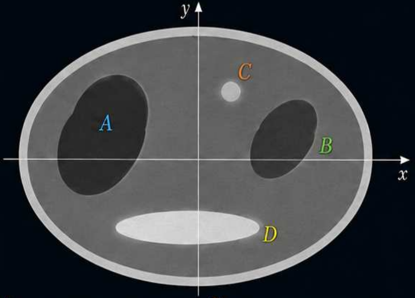
Why it matters: Fundamental building block—any object can be viewed as a superposition of shifted point responses.

Takeaway: *Rotations shift the sinogram in angle; simple objects create projection profiles that are easy to recognize analytically.*

A High-Resolution Object and Its Radon Transform

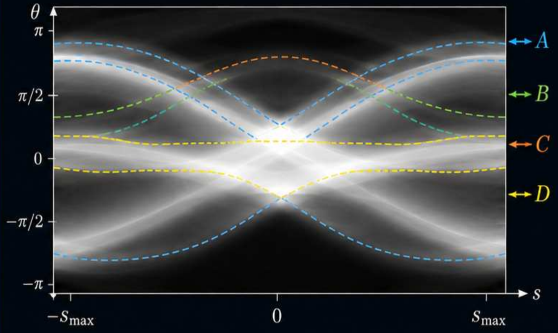
Larger image features map into structured, interpretable traces in the sinogram.

① High-resolution object $f(x,y)$

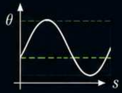


- A Left dark oval (elongated)
- B Right dark oval (elongated)
- C Small bright circular spot
- D Bright horizontal ellipse (elongated)

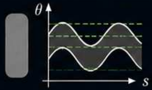
② Sinogram $p(\theta, s)$



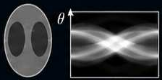
③ How to read the sinogram



- **Points** produce thin sinusoidal traces.



- **Extended regions** produce broader bands.



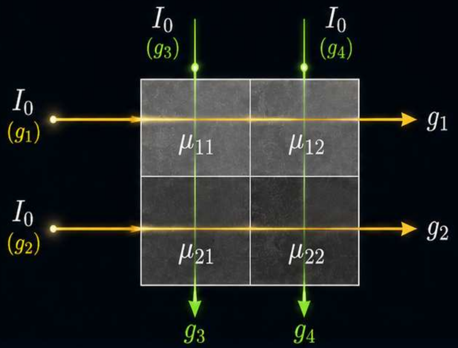
- **Symmetry** in the object produces mirrored or repeated structures.

Takeaway: *The sinogram is not abstract: its curves are direct signatures of the geometry inside the object.*

A 2×2 Beer–Lambert Tomography Example

Two horizontal rays and two vertical rays produce Beer–Lambert measurements from a tiny attenuation image.

Tiny 2×2 attenuation image (unknown coefficients μ)



$$x = \begin{bmatrix} \mu_{11} \\ \mu_{12} \\ \mu_{21} \\ \mu_{22} \end{bmatrix}^T$$

Beer–Lambert measurements (exponential form)

$$g_1 = I_0 \exp\left(-(\mu_{11} + \mu_{12})\right)$$

$$g_2 = I_0 \exp\left(-(\mu_{21} + \mu_{22})\right)$$

$$g_3 = I_0 \exp\left(-(\mu_{11} + \mu_{21})\right)$$

$$g_4 = I_0 \exp\left(-(\mu_{12} + \mu_{22})\right)$$

Intuition Each ray **adds** the attenuation coefficients it crosses; Beer–Lambert converts that path sum into an **exponential intensity** measurement.

Logarithmic linearization (turn exponentials into sums)

$$\boxed{y_i = -\ln\left(\frac{g_i}{I_0}\right)}$$

Therefore,

$$\begin{cases} y_1 = \mu_{11} + \mu_{12}, \\ y_2 = \mu_{21} + \mu_{22}, \\ y_3 = \mu_{11} + \mu_{21}, \\ y_4 = \mu_{12} + \mu_{22}. \end{cases}$$

Takeaway: after the negative log, the nonlinear x-ray measurements become linear projection equations. >>>

Inverse Formulation for the 2×2 Attenuation Image

After the negative log, horizontal and vertical Beer–Lambert data become a small linear system.

Unknowns and measurements

$$x = \begin{bmatrix} \mu_{11} \\ \mu_{12} \\ \mu_{21} \\ \mu_{22} \end{bmatrix}^T$$

Unknown attenuation coefficients (vectorized).

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$\boxed{y_i = -\ln\left(\frac{g_i}{I_0}\right)}$$

Measured (log) data from horizontal (y_1, y_2) and vertical (y_3, y_4) rays.

Linear system after the negative log

$$y = Ax$$

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$y_1 = \mu_{11} + \mu_{12}$$

$$y_2 = \mu_{21} + \mu_{22}$$

$$y_3 = \mu_{11} + \mu_{21}$$

$$y_4 = \mu_{12} + \mu_{22}$$

Key observation: one equation is dependent

- $y_1 + y_2 = y_3 + y_4$ (row sums = column sums)
- $\Rightarrow$ One equation is dependent.
- $\mathrm{rank}(A) = 3 \Rightarrow$ The system is underdetermined.
- Therefore, horizontal + vertical projections alone do not uniquely determine all four unknowns.

One-parameter family of solutions

Let $t = \mu_{11}$. Then all solutions are

$$\left\{ \begin{aligned} \mu_{11} &= t \\ \mu_{12} &= y_1 - t \\ \mu_{21} &= y_3 - t \\ \mu_{22} &= y_2 - y_3 + t \end{aligned} \right.$$

Different 2×2 images (different t) can produce the same four measurements.

Example (same y)

$t = 0$

0	y_1
y_3	$y_2 - y_3$

or

$t = 1$

1	$y_1 - 1$
$y_3 - 1$	$y_2 - y_3 + 1$

Same row sums (y_1, y_2) and column sums (y_3, y_4).

Takeaway: *the inverse problem becomes linear after the log, but with only horizontal and vertical rays the 2×2 system is underdetermined.*

Recovering the 2×2 Coefficients

Add one more independent projection, then solve for the attenuation coefficients explicitly.

1) Five independent projections

From the four axis-aligned rays:

$$y_1 = \mu_{11} + \mu_{12}$$
$$y_2 = \mu_{21} + \mu_{22}$$
$$y_3 = \mu_{11} + \mu_{21}$$
$$y_4 = \mu_{12} + \mu_{22}$$

Add one diagonal ray (top-left → bottom-right):

$$g_5 = I_0 \exp\left(-(\mu_{11} + \mu_{22})\right)$$

After negative log:

$$\boxed{y_5 = \mu_{11} + \mu_{22}}$$

This fifth, independent measurement makes the system informative enough to recover the coefficients uniquely in this toy example.

2) Solve for the coefficients

Add and subtract the equations:

$$y_3 - y_2 = (\mu_{11} + \mu_{21}) - (\mu_{21} + \mu_{22}) = \mu_{11} - \mu_{22}$$
$$y_5 = \mu_{11} + \mu_{22}$$
$$\Rightarrow \quad 2\mu_{11} = y_3 - y_2 + y_5$$
$$\boxed{\mu_{11} = \frac{y_3 + y_5 - y_2}{2}}$$
$$\Rightarrow \quad \mu_{12} = y_1 - \mu_{11}$$
$$\Rightarrow \quad \mu_{21} = y_3 - \mu_{11}$$
$$\Rightarrow \quad \mu_{22} = y_5 - \mu_{11}$$

Consistency check: The remaining equation

$$y_4 = \mu_{12} + \mu_{22}$$

should hold (up to numerical error).

3) Numerical example

Measured (negative-log) projections:

$$y_1 = 0.5 \qquad y_2 = 0.5$$
$$y_3 = 0.7 \qquad y_4 = 0.3$$
$$y_5 = 0.4$$

Recovered coefficients:

$$\mu_{11} = 0.3 \qquad \mu_{12} = 0.2$$
$$\mu_{21} = 0.4 \qquad \mu_{22} = 0.1$$

Reconstructed 2×2 attenuation image:

0.3	0.2
0.4	0.1

(rows = vertical, columns = horizontal)

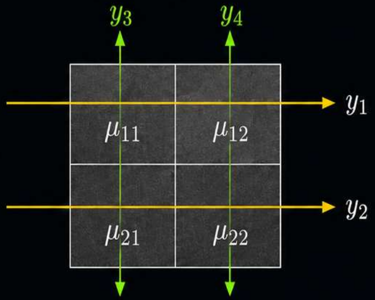
In practical CT, many more projections over many angles are collected so that large images can be reconstructed robustly.

Takeaway: Beer–Lambert gives nonlinear intensities, the negative log gives linear projections. and enough independent rays let us recover the attenuation coefficients.

Backprojection on a 2×2 Tomography Example

Start with four negative-log projections and a tiny attenuation image.

1) The tiny attenuation image and its four ray sums



$$\boxed{y_i = -\ln\left(\frac{g_i}{I_0}\right)}$$

Each y_i is a negative-log projection (measured sum) along one ray.

2) Unknowns, measurements, and line-sum equations

Unknown vector	Measurement vector	Line-sum equations
$x = \begin{bmatrix} \mu_{11} \\ \mu_{12} \\ \mu_{21} \\ \mu_{22} \end{bmatrix} \in \mathbb{R}^4$	$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \in \mathbb{R}^4$	$y_1 = \mu_{11} + \mu_{12}$ (top horizontal) $y_2 = \mu_{21} + \mu_{22}$ (bottom horizontal) $y_3 = \mu_{11} + \mu_{21}$ (left vertical) $y_4 = \mu_{12} + \mu_{22}$ (right vertical)

Takeaway: *After the negative log,* **Beer–Lambert measurements become line sums.**

The Backprojection Operator

Backprojection smears each projection value back over all pixels touched by that ray.

1) Forward linear system

Forward model:

$$y = Ax$$

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

2) Backprojection operator

Define backprojection:

$$b = A^T y$$

$$b = \begin{bmatrix} y_1 + y_3 \\ y_1 + y_4 \\ y_2 + y_3 \\ y_2 + y_4 \end{bmatrix}$$

Normalized backprojection estimate:

$$x_{\mathrm{BP}} = \frac{1}{2} A^T y = \frac{1}{2} \begin{bmatrix} y_1 + y_3 \\ y_1 + y_4 \\ y_2 + y_3 \\ y_2 + y_4 \end{bmatrix}$$

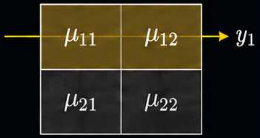
Each pixel is hit by two rays in this toy geometry (one horizontal and one vertical). This is why we use the factor 1/2 in the normalized backprojection.

Takeaway: *Backprojection is the adjoint of the forward model: smear and add.*

Visual Intuition: How Each Ray Is Smeared Back

Each measurement is painted uniformly over the pixels on its path.

1) Top horizontal ray (y_1)



y_1 contributes $\frac{y_1}{2}$ to μ_{11} and $\frac{y_1}{2}$ to μ_{12}.

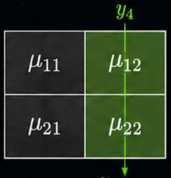
2) Bottom horizontal ray (y_2)

y_2 contributes $\frac{y_2}{2}$ to μ_{21} and $\frac{y_2}{2}$ to μ_{22}.

3) Left vertical ray (y_3)

y_3 contributes $\frac{y_2}{2}$ to μ_{11} and $\frac{y_3}{2}$ to μ_{21}.

4) Right vertical ray (y_4)



y_4 contributes $\frac{y_4}{2}$ to μ_{12} and $\frac{y_4}{2}$ to μ_{22}.

5) Sum of all contributions

$\frac{y_1 + y_3}{2}$	$\frac{y_1 + y_4}{2}$
$\frac{y_2 + y_3}{2}$	$\frac{y_2 + y_4}{2}$

$\Rightarrow$

- easy to compute
- uses only geometry
- but spreads information across multiple pixels

Takeaway: *Backprojection is intuitive, but it **blurs** because each ray sum is **shared**.*

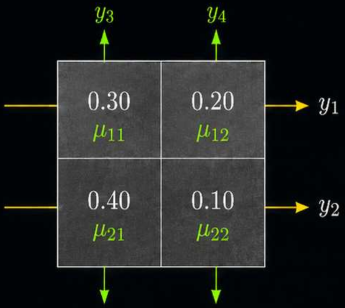
Numerical Backprojection Example

Use a concrete 2×2 attenuation image and compute the four projections.

1) True 2×2 attenuation image and rays

True attenuation image μ :

$$\mu = \begin{bmatrix} 0.30 & 0.20 \\ 0.40 & 0.10 \end{bmatrix}$$



2) Negative–log projections (measured sums)

Each projection is the sum of μ along the ray.

y_1 (top horizontal): $0.30 + 0.20 = 0.50$

y_2 (bottom horizontal): $0.40 + 0.10 = 0.50$

y_3 (left vertical): $0.30 + 0.40 = 0.70$

y_4 (right vertical): $0.20 + 0.10 = 0.30$

Raw Beer–Lambert intensities (with $I_0 = 1000$):

$$g_1 = 1000\, e^{-0.50} \approx 606.5$$
$$g_2 = 1000\, e^{-0.50} \approx 606.5$$
$$g_3 = 1000\, e^{-0.70} \approx 496.6$$
$$g_4 = 1000\, e^{-0.30} \approx 740.8$$

3) Backproject each ray (distribute uniformly)

Spread each measurement equally across the pixels that the ray crosses.

y_1 (top horizontal):

0.25	0.25
0	0

y_2 (bottom horizontal):

0	0
0.25	0.25

y_3 (left vertical):

0.35	0
0.35	0

y_4 (right vertical):

0	0.15
0	0.15

Takeaway: *Now we can smear the measurements back and compare with the true image.*

The Backprojected Image Is Blurred

Summing the smeared contributions gives a simple estimate, but not the exact attenuation map.

1) Backprojection result (numerical example)

From the previous slide, the backprojected estimate is

$$x_{\mathrm{BP}} = \frac{1}{2} A^T y$$

$$= \frac{1}{2} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}^T \begin{bmatrix} 0.3 \\ 0.2 \\ 0.4 \\ 0.1 \end{bmatrix}$$

$$= \boxed{\begin{matrix} 0.60 & 0.40 \\ 0.60 & 0.40 \end{matrix}}$$

Thus, with only the four horizontal and vertical projections, we obtain

$$x_{\mathrm{BP}} = \begin{bmatrix} 0.60 & 0.40 \\ 0.60 & 0.40 \end{bmatrix}$$

2) Compare to the true image

True image (x)

0.30	0.20
0.40	0.10

vs.

Backprojected image (x_{BP})

0.60	0.40
0.60	0.40

Rows became identical because $y_1 = y_2$ (the two horizontal sums are the same).

Column values are averaged/smeared because only horizontal views contribute along columns.

- Each measured ray collects contributions from multiple pixels.
- With only horizontal and vertical views, pixels cannot be isolated.
- Important angular information is missing, so the inverse is not exact.

Takeaway: *Simple backprojection produces a smeared image rather than the true image.*

Why Backprojection Is Not an Exact Inverse

Backprojection applies A^T, so the result is A^TAx rather than x.

1) Backprojection equals A^TAx

Forward model:

$$y = Ax$$

Backprojection:

$$x_{BP} = \frac{1}{2}A^Ty = \frac{1}{2}A^TAx$$

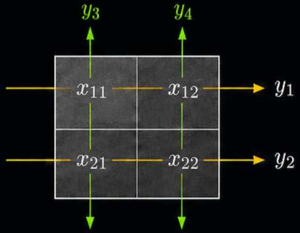
For the 2×2 example,

$$A^TA = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

Not the identity matrix I_4.
Therefore, $x_{BP} \neq x$ in general.

2) What A^TA means

- **Diagonal terms** count self-hits: how many rays pass through each pixel.
- **Off-diagonal terms** represent **cross-talk**: how much two different pixels share rays.



Pixels on the **diagonal** share 2 rays with themselves.
Pixels **off the diagonal** share 1 ray with each other.

3) Numerical example

True image vector:

$$x = \begin{bmatrix} 0.3 \\ 0.2 \\ 0.4 \\ 0.1 \end{bmatrix}$$

Backprojection:

$$x_{BP} = \frac{1}{2}A^TAx = \begin{bmatrix} 0.6 \\ 0.4 \\ 0.6 \\ 0.4 \end{bmatrix}$$

True 2×2 image:

0.3	0.2
0.4	0.1

$\Rightarrow$

Backprojected 2×2 image:

0.6	0.4
0.6	0.4

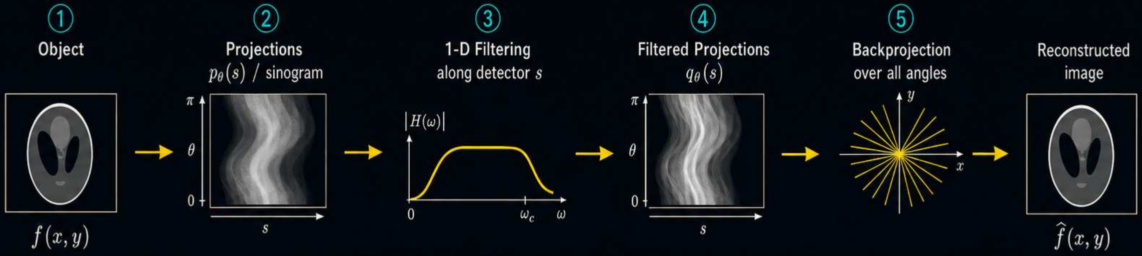
$$A^TA \neq I,$$

*so backprojection is not the inverse; it is the **adjoint**.*

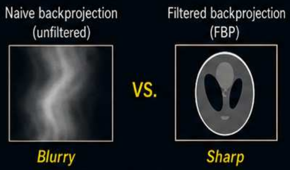
Takeaway: *To undo this blur we need filtering — motivating **filtered backprojection**.*

Filtered Backprojection (FBP): Big Picture

Filter each projection first, then backproject to undo the blur of naive backprojection.



Naive vs. Filtered Backprojection



$$b(x,y) = \int_0^\pi p_\theta(x\cos\theta + y\sin\theta)\, d\theta$$ *(naive backprojection)*

$$\boxed{\hat{f}(x,y) = \int_0^\pi q_\theta(x\cos\theta + y\sin\theta)\, d\theta}$$ *(FBP)*

$$q_\theta = p_\theta * h$$

- $p_\theta(s)$: Radon projection of $f(x, y)$ at angle θ (detector coordinate s).
- $h(s)$: 1-D filter (ramp-like in frequency) applied along s.
- $q_\theta(s)$: Filtered projection obtained by convolving $p_\theta(s)$ with $h(s)$.

Takeaway: *FBP = projection filtering + backprojection; the filter compensates for the low-frequency overemphasis of simple backprojection.*

Why Naive Backprojection Blurs

Backprojection spreads each projection over a full line, which overweights low spatial frequencies.

1 Naive backprojection (reminder)

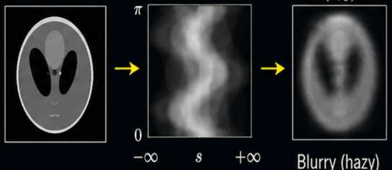
Backproject all projections and sum.

$$b(x,y) = \int_0^{\pi} p_\theta(x\cos\theta + y\sin\theta)\, d\theta$$

Object

Sinogram $p_\theta(s)$

Naive backprojection $b(x,y)$



Blurry (hazy)

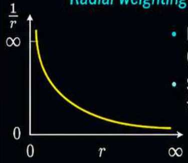
2 Frequency-domain explanation

In the frequency domain, naive backprojection attenuates high frequencies by a factor $1/r$.

$$B(\omega_x,\omega_y) = \frac{F(\omega_x,\omega_y)}{\sqrt{\omega_x^2+\omega_y^2}}$$

Let $r = \sqrt{\omega_x^2+\omega_y^2}$, so $B(\omega_x,\omega_y) = \frac{F(\omega_x,\omega_y)}{r}$

Radial weighting $1/r$



- Large near origin (low frequencies)
- Small at high frequencies

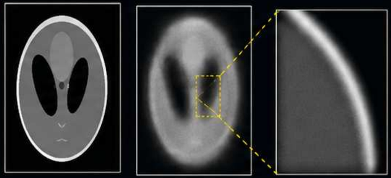
3 Intuition: low frequencies become too strong

Because low frequencies are overemphasized, the reconstruction becomes hazy and edges blur.

True object

Naive backprojection

Blurred edges

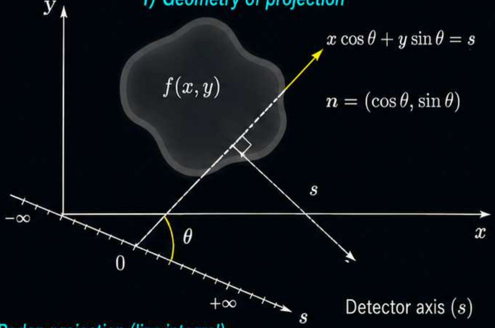


To correct the $1/r$ effect, **multiply by the ramp filter** $H(\omega) = |\omega|$ before backprojection.

Fourier Slice Theorem: Geometry and Definitions

A 1-D Fourier transform of a projection equals a radial slice of the 2-D Fourier transform of the object.

1) Geometry of projection



Radon projection (line integral)

$$p_\theta(s) = \iint_{-\infty}^{\infty} f(x, y)\, \delta(s - x\cos\theta - y\sin\theta)\, dx\, dy$$

$p_\theta(s) =$ projection of the object at angle θ along the line $x\cos\theta + y\sin\theta = s$
$s =$ detector coordinate (signed distance from origin along $\boldsymbol{n}$)

2) Definitions and frequency mapping

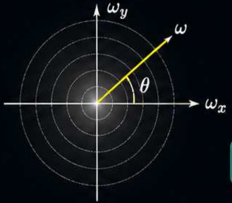
1-D Fourier transform of the projection

$$P_\theta(\omega) = \int_{-\infty}^{\infty} p_\theta(s)\, e^{-j\omega s}\, ds$$

2-D Fourier transform of the object

$$F(\omega_x, \omega_y) = \iint_{-\infty}^{\infty} f(x, y)\, e^{-j(\omega_x x + \omega_y y)}\, dx\, dy$$

Frequency plane and slice (angle θ)



Radial line at angle θ

$\omega =$ 1-D frequency (radial)

$$(\omega_x, \omega_y) = (\omega\cos\theta,\ \omega\sin\theta)$$

Takeaway: *Each projection angle samples one radial line in 2-D frequency space.*

Fourier Slice Theorem: Derivation

Insert the Radon definition into the 1-D Fourier transform and simplify.

1 1-D Fourier transform of the projection

$$P_\theta(\omega) = \int_{-\infty}^{\infty} p_\theta(s)\, e^{-j\omega s}\, ds$$

2 Substitute Radon definition

$$P_\theta(\omega) = \int_{-\infty}^{\infty} \left(\iint_{-\infty}^{\infty} f(x,y)\, \delta(s - x\cos\theta - y\sin\theta)\, dx\, dy \right) e^{-j\omega s}\, ds$$

3 Swap integrals and evaluate the delta integral

$$P_\theta(\omega) = \iint_{-\infty}^{\infty} f(x,y) \left(\int_{-\infty}^{\infty} \delta(s - x\cos\theta - y\sin\theta)\, e^{-j\omega s}\, ds \right) dx\, dy$$

$$= \iint_{-\infty}^{\infty} f(x,y)\, e^{-j\omega(x\cos\theta + y\sin\theta)}\, dx\, dy$$

4 Recognize the 2-D Fourier transform

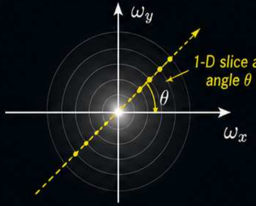
$$P_\theta(\omega) = \iint_{-\infty}^{\infty} f(x,y)\, e^{-j\omega(x\cos\theta + y\sin\theta)}\, dx\, dy$$

$$= F(\omega\cos\theta, \omega\sin\theta)$$

Interpretation

Thus the 1-D spectrum of a projection is a slice through the 2-D object spectrum at the same angle.

Frequency-plane view



Slice coordinates:

$$(\omega_x, \omega_y) = (\omega\cos\theta, \omega\sin\theta)$$

for $-\infty < \omega < \infty$

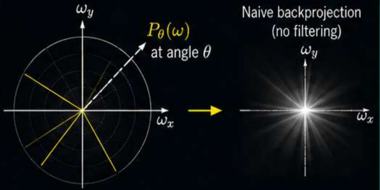
Takeaway: *The theorem is the mathematical bridge from projections to frequency-domain reconstruction.*

From the Fourier Slice Theorem to Filtered Backprojection

Reconstruction is obtained by reweighting radial frequency samples before integrating over angle.

1) Naive backprojection in frequency domain

Polar reconstruction idea:
For each angle θ, we place the 1-D spectrum $P_\theta(\omega)$ along the radial line at angle θ in 2-D frequency.



Integrating unweighted radial samples over all angles overcounts low frequencies. The result is equivalent to multiplying the true spectrum by $1/r$.

$$\boxed{F_{\text{naive}}(r,\phi) = \frac{F(r,\phi)}{r}}$$

where $r = \sqrt{\omega_x^2 + \omega_y^2}$

2) Correction by the ramp filter

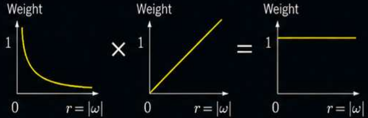
Compensate the $1/r$ bias by applying a ramp filter to each projection in 1-D frequency.

$$\boxed{H(\omega) = |\omega|,}$$

$$\boxed{Q_\theta(\omega) = H(\omega)P_\theta(\omega) = |\omega|P_\theta(\omega)}$$

$$\boxed{q_\theta(s) = \mathcal{F}^{-1}\{Q_\theta(\omega)\}}$$

Radial-frequency weighting comparison



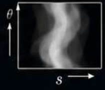
The ramp filter restores **balanced** frequency coverage.

3) Filtered backprojection (FBP)

Reconstruction is obtained by backprojecting the filtered projections over all angles.

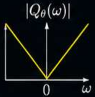
$$\boxed{\hat{f}(x,y) = \int_0^\pi q_\theta\left(x\cos\theta + y\sin\theta\right) d\theta}$$

1



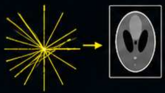
Filter each projection independently.
Apply ramp filter $H(\omega) = |\omega|$ in the 1-D frequency domain.

2



Inverse transform to obtain the filtered projection.
$q_\theta(s) = \mathcal{F}^{-1}\{Q_\theta(\omega)\}$

3



Backproject the filtered data over all angles.
Integrate q_θ along the lines $s = x\cos\theta + y\sin\theta$ for $\theta \in [0,\pi)$.

Takeaway: *Filtered backprojection = filter each projection, then backproject; the ramp filter corrects the $1/r$ bias and restores balanced frequencies.*

Practical FBP Filters: Ramp and Windowed Variants

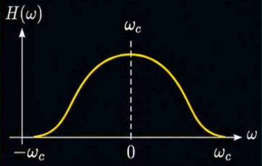
The ideal ramp restores detail, while windows reduce noise at the cost of some blur.

1) Ram-Lak (ideal ramp)

$$H_{RL}(\omega) = \begin{cases} |\omega|, & |\omega| \le \omega_c \\ 0, & |\omega| > \omega_c \end{cases}$$

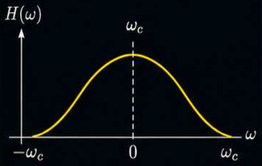
2) Shepp-Logan

$$H_{SL}(\omega) = |\omega| \,\mathrm{sinc}\left(\frac{\omega}{2\omega_c}\right)$$



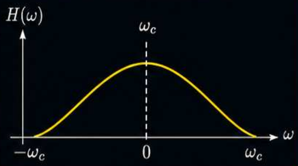
3) Cosine

$$H_{C}(\omega) = |\omega| \cos\left(\frac{\pi\omega}{2\omega_c}\right)$$



4) Hann or Hamming windowed ramp

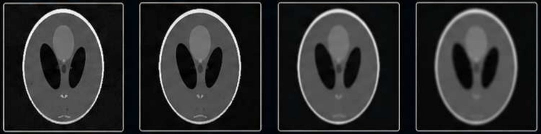
$$H_{H}(\omega) = |\omega|\, w(\omega)$$
$$w(\omega) = \text{Hann or Hamming window}$$



Filter	Sharpness (detail)	Noise sensitivity	Typical use
Ramp (Ram-Lak)	★★★★★ (highest)	★★★★★ (highest)	When noise is low; max detail / resolution
Shepp-Logan	★★★★☆	★★★☆☆	Balanced choice; good general use
Cosine	★★★☆☆	★★☆☆☆	Lower noise; mildly smoother
Hann-Hamming	★★☆☆☆ (lowest)	★☆☆☆☆ (lowest)	Noisy data; smooth reconstructions

Reconstructions of the same phantom (different filters)

Ramp (Ram-Lak) | Shepp-Logan | Cosine | Hann-Hamming



Sharpest edges, more noise | Good balance of detail and noise | Smoother, less noise | Smoothest, least noise (more blur)

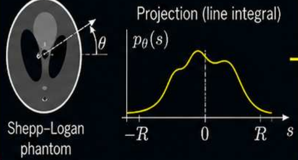
Note: *Windowing suppresses high-frequency noise but also softens edges.*

Step-by-Step Numerical FBP Example

Watch one projection be transformed, filtered, inverted, and then backprojected.

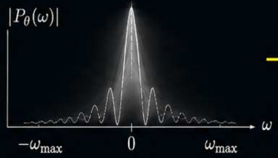
1) Projection $p_\theta(s)$

One angle $\theta = 30°$



2) 1-D Fourier transform $P_\theta(\omega)$

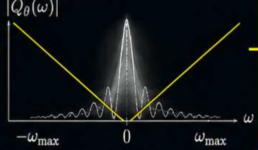
Magnitude spectrum



3) Multiply by ramp filter

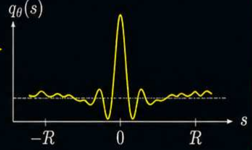
$H(\omega) = |\omega|$

Get $Q_\theta(\omega) = |\omega| P_\theta(\omega)$

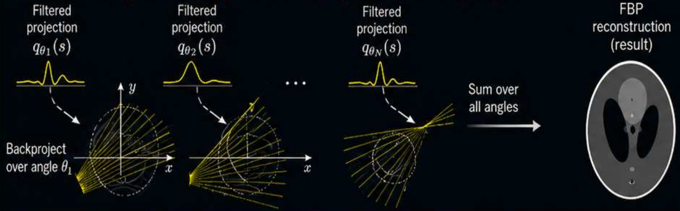


4) Inverse transform to filtered projection $q_\theta(s)$

Back to spatial domain

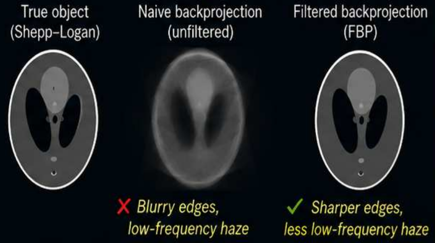


Many filtered projections backprojected over many angles



Filtered projections smear energy along each angle; summing over $0° \leq \theta < 180°$ reconstructs the object.

Final comparison



Takeaway: *FBP restores high frequencies with the ramp filter, then backprojects to yield a sharp, quantitative reconstruction.*

Discrete FBP Algorithm and Practical Issues

How FBP is implemented from sampled projections, and what can go wrong in practice.

1) Discrete formulas

$$P[k,l] = \sum_{m=0}^{M-1} p[m,\theta_k]\, e^{-j2\pi \frac{lm}{M}}$$ (FFT of projection at angle θ_k)

$$Q[k,l] = H[l]\; P[k,l]$$ (Multiply by ramp / windowed filter)

$$q[m,\theta_k] = \mathrm{IDFT}\{Q[k,l]\}$$ (Back to spatial detector domain)

$$\hat{f}[i,j] = \sum_{k=0}^{K-1} q(s_{ijk},\theta_k)\,\Delta\theta$$ (Backproject and sum)

$$\boxed{s_{ijk} = x_i \cos\theta_k + y_j \sin\theta_k}$$ (Continuous detector coordinate → interpolation is needed)

where:

- $p[m,\theta_k]$: sampled projection at angle θ_k and detector index $m = 0,\dots,M-1$
- $P[k,l]$: DFT over detector index m, frequency index $l = 0,\dots,M-1$
- $H[l]$: ramp filter (optionally windowed) in frequency domain
- $q[m,\theta_k]$: filtered projection (discrete)
- $\hat{f}[i,j]$: reconstructed image at pixel (x_i, y_j)
- $\Delta\theta$: angular spacing between projections

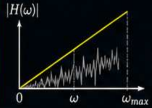
2) Discrete FBP algorithm (pseudocode)

```
Input: projections p[m, k]   # m = 0..M-1, k = 0..K-1
Output: reconstructed image f[i, j]   # i = 0..N_x-1, j = 0..N_y-1

Precompute: frequency axis l = 0..M-1 and filter H[l] (ramp or windowed)
Precompute: angle array theta[k] = k * Delta_theta,   k = 0..K-1
Initialize  f[i, j] = 0 for all i, j

for k = 0 to K-1:
   P_k[l] = FFT{ p[m, k] }            # FFT along detector (over m)
   Q_k[l] = H[l] * P_k[l]             # Apply ramp/windowed filter
   q_k[m] = IFFT{(Q_k[l])}            # Back to spatial domain

   for each image pixel (i, j):
      s = x_i * cos(theta[k]) + y_j * sin(theta[k])   # Detector coordinate
      q_interp = interpolate{ q_k[m], at = s }        # Linear/cubic interp.
      f[i, j] += q_interp * Delta_theta               # Accumulate

f[i, j] *= C                                          # Normalize (e.g., C = pi / (2K))
```

1 Sparse angles cause streaks

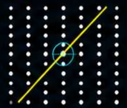
Too few or unevenly spaced angles leave missing wedges in frequency, producing streak artifacts.

2 Ramp filter amplifies noise



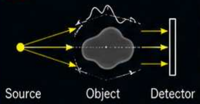
The ramp grows with frequency, boosting high-frequency noise. Windowing trades resolution for noise.

3 Detector sampling & interpolation matter



Finite detector pitch and interpolation errors affect image accuracy and can introduce artifacts, especially at edges.

4 Ideal Radon model ignores physics



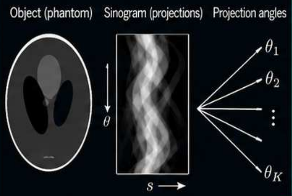
Real data include blur (focal spot, detector), scatter, beam hardening, and nonlinearity—not captured by the ideal model.

Takeaway: *FBP is fast and foundational, but its quality depends on sampling, noise, filtering, and model accuracy.* ››››

Naive Fourier Reconstruction from Multiple Fourier Slices

Place each projection spectrum on a radial line, interpolate onto a Cartesian grid, then invert — a natural but imperfect reconstruction strategy.

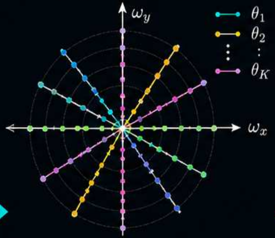
1) Many projection angles create many 1-D spectra



$$P_\theta(\omega) = F(\omega\cos\theta, \omega\sin\theta)$$

- Each projection angle θ_k yields one 1-D Fourier spectrum $P_{\theta_k}(\omega)$ — a radial slice in 2-D.
- More angles give denser coverage of the 2-D frequency plane.
- Low frequencies (near the center) are sampled by all angles; high frequencies by few angles.

2) Populate the 2-D frequency plane with radial slices

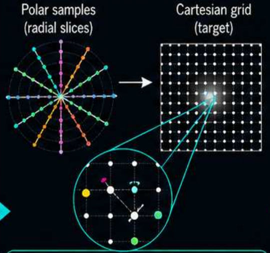


$$\mathcal{S} = \{(\omega_m, \theta_k), P_{\theta_k}(\omega_m)\}$$

$$(\omega_x, \omega_y) = (\omega_m\cos\theta_k, \omega_m\sin\theta_k)$$

Radial (polar) samples do not lie on the regular Cartesian FFT grid.

3) Interpolate from the polar samples to a Cartesian grid



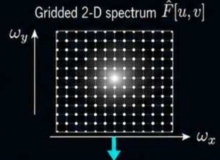
$$\hat{F}[u,v] \approx \mathrm{Interp}\{P_{\theta_k}(\omega_m)\}$$

(u,v): Cartesian frequency-grid indices

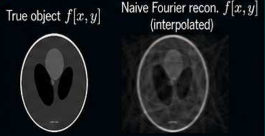
Interp: nearest-neighbor, bilinear, or gridding interpolation

- Interpolation introduces approximation error.
- Center is oversampled; outer frequencies are less uniformly covered.
- Coarse angular sampling leaves gaps.

4) Inverse 2-D DFT on the gridded spectrum



$$\hat{f}[x,y] = \mathcal{F}_{2D}^{-1}\{\hat{F}[u,v]\}$$



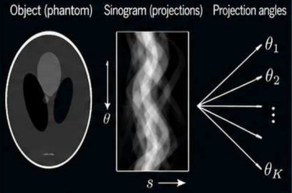
- Simple and intuitive reconstruction pipeline.
- Works from the Fourier Slice Theorem directly.
- Interpolation and nonuniform sampling introduce blur and streak artifacts.

Takeaway: *The naive Fourier method reconstructs by **gridding radial Fourier slices** and applying the inverse 2-D transform, but **interpolation** and **uneven polar sampling** limit reconstruction quality — **motivating filtered backprojection** and more advanced gridding methods.*

Naive Fourier Reconstruction from Multiple Fourier Slices

Place each projection spectrum on a radial line, interpolate onto a Cartesian grid, then invert — a natural but imperfect reconstruction strategy.

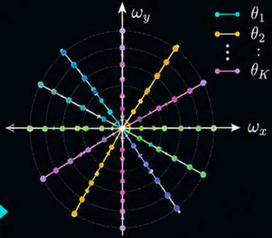
1) Many projection angles create many 1-D spectra



$$P_\theta(\omega) = F(\omega\cos\theta, \omega\sin\theta)$$

- Each projection angle θ_k yields one 1-D Fourier spectrum $P_{\theta_k}(\omega)$ — a radial slice in 2-D.
- More angles give denser coverage of the 2-D frequency plane.
- Low frequencies (near the center) are sampled by all angles; high frequencies by few angles.

2) Populate the 2-D frequency plane with radial slices

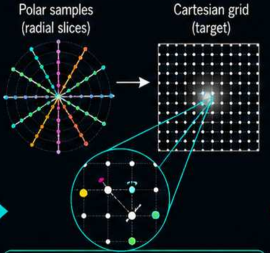


$$\mathcal{S} = \{(\omega_m, \theta_k), P_{\theta_k}(\omega_m)\}$$

$$(\omega_x, \omega_y) = (\omega_m\cos\theta_k, \omega_m\sin\theta_k)$$

Radial (polar) samples do not lie on the regular Cartesian FFT grid.

3) Interpolate from the polar samples to a Cartesian grid



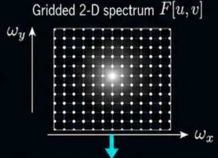
$$\hat{F}[u,v] \approx \mathrm{Interp}\left\{P_{\theta_k}(\omega_m)\right\}$$

(u,v): Cartesian frequency-grid indices

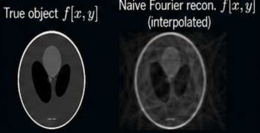
Interp: nearest-neighbor, bilinear, or gridding interpolation

- Interpolation introduces approximation error.
- Center is oversampled; outer frequencies are less uniformly covered.
- Coarse angular sampling leaves gaps.

4) Inverse 2-D DFT on the gridded spectrum



$$\hat{f}[x,y] = \mathcal{F}_{2D}^{-1}\left\{\hat{F}[u,v]\right\}$$



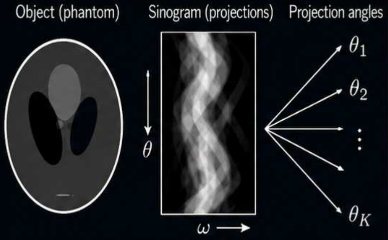
- Simple and intuitive reconstruction pipeline.
- Works from the Fourier Slice Theorem directly.
- Interpolation and nonuniform sampling introduce blur and streak artifacts.

Takeaway: *The naive Fourier method reconstructs by **gridding radial Fourier slices** and applying the inverse 2-D transform, but **interpolation** and **uneven polar sampling** limit reconstruction quality — **motivating filtered backprojection** and more advanced gridding methods.*

Naive Fourier Reconstruction from Multiple Fourier Slices – Part I

Many projection angles create radial Fourier slices that sample the 2-D frequency plane.

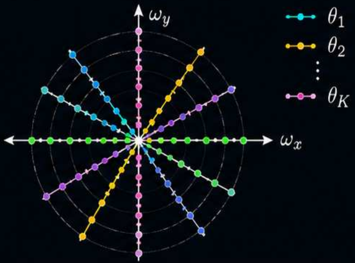
1) From projections to 1-D spectra



$$P_\theta(\omega) = F(\omega\cos\theta, \omega\sin\theta)$$

- Each projection at angle θ_k yields one 1-D Fourier spectrum $P_{\theta_k}(\omega)$ – a radial slice in 2-D.
- More angles give denser coverage of the 2-D frequency plane.

2) Populate the 2-D frequency plane with radial slices



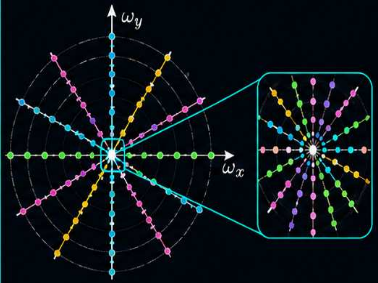
$$S = \{(\omega_m, \theta_k), P_{\theta_k}(\omega_m)\}$$

$$(\omega_x, \omega_y) = (\omega_m\cos\theta_k, \omega_m\sin\theta_k)$$

Radial (polar) samples do not lie on the regular Cartesian FFT grid.

3) Sampling intuition

Low frequencies (near the origin) are densely sampled; outer frequencies are more sparse.



• • • Low $|\omega|$ (dense samples)

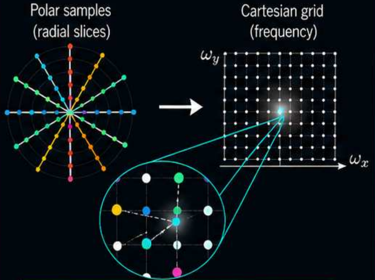
• – • High $|\omega|$ (sparse samples)

Takeaway: *The Fourier Slice Theorem gives many radial samples of the 2-D spectrum, but these samples live on a polar grid rather than the Cartesian grid required by the inverse 2-D DFT.*

Naive Fourier Reconstruction from Multiple Fourier Slices — Part II

Interpolate the polar Fourier samples onto a Cartesian grid, then apply the inverse 2-D DFT.

1) Interpolate from polar samples to a Cartesian grid

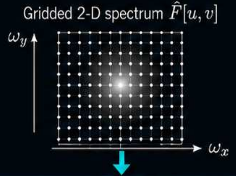


$$\hat{F}[u,v] \approx \mathrm{Interp}\left\{P_{\theta_k}(\omega_m)\right\}$$

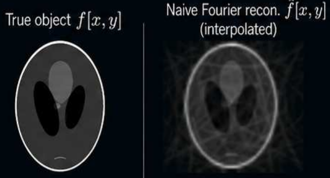
(u,v): Cartesian frequency-grid indices

- Interpolation choices: nearest-neighbor, bilinear, or gridding interpolation.

2) Inverse 2-D DFT on the gridded spectrum



$$\hat{f}[x,y] = \mathcal{F}_{2D}^{-1}\left\{\hat{F}[u,v]\right\}$$



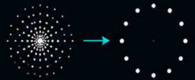
3) Why the naive method is imperfect

Interpolation error
Interpolation approximates values between polar samples.

Nonuniform polar sampling
Many more samples near the center; outer regions are effectively undersampled.

Sparse high-frequency coverage
High frequencies are sampled sparsely, leading to information gaps.

Streak / blur artifacts
Coarse angular sampling and interpolation cause streaks and blur in the image.



Denser near the center; sparser toward the outer radius.

Takeaway: *The naive Fourier method is intuitive, but **interpolation** and **uneven polar sampling** limit reconstruction quality—**motivating filtered backprojection** and more advanced **gridding methods**.*

The Power of CT: Looking Inside Natural Objects

Computed tomography reveals rich internal structure nondestructively.

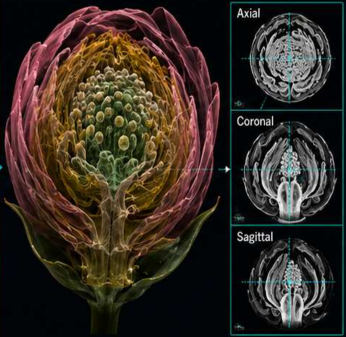
1 Exterior (Visible Light)

Natural beauty on the outside—delicate petals, subtle textures, no hint of the hidden complexity within.

2 CT Cross-Section (Single Slice)

CT slice reveals internal organization: layered petals, reproductive structures, air spaces, and density variations.

3 3-D CT Rendering & Orthogonal Slices



3-D view and orthogonal slices expose the true anatomy: vascular bundles, cavities, and spatial relationships—impossible to see from the outside.

Why CT is powerful

Nondestructive
Preserves the object while revealing what's inside.

High Resolution
Micron-to-submillimeter detail captures fine structures and density variations.

Quantitative Insight
Measure dimensions, densities, and volumes with high accuracy.

Broad Impact
Essential for biology, materials, paleontology, agriculture, and food inspection.

Outside Appearance
Smooth, uniform bud.
Looks solid and simple.

VS

Inside Structure
Complex, highly organized, multi-layered, and functional.

Takeaway: *CT can peer inside complex objects—revealing geometry, density, and hidden structure that conventional photography cannot see.*

First-Generation CT Acquisition: Translate-Rotate Pencil-Beam Geometry

How the original EMI head scanner formed projections by shifting a pencil beam, then rotating and repeating.

ACQUISITION SEQUENCE

1 
At angle θ, translate the pencil beam across the head.

2 

Each position s records one line integral (one projection sample).

3 
Rotate the source-detector pair by a small angle $\Delta\theta \approx 1°$.

4 
Repeat until projections are collected over many angles.

FIRST-GENERATION TRANSLATE–ROTATE ACQUISITION

1) TRANSLATE AT FIXED ANGLE θ

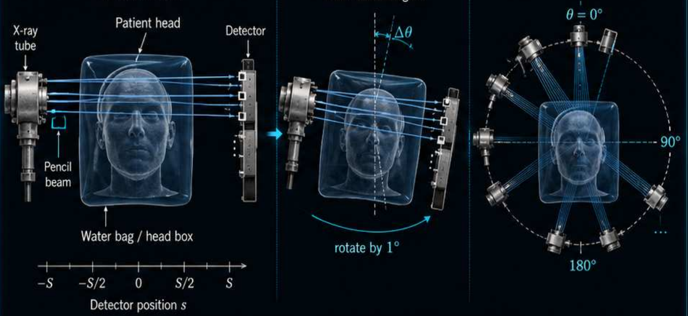
Source–detector pair translates linearly across the head.

2) ROTATE BY A SMALL ANGLE $\Delta\theta \approx 1°$

Then translate again.

3) REPEAT FOR MANY ANGLES

Collect projections over ~180°.



At each angle θ, translation samples many rays (line integrals).
Rotation by small steps builds a complete set of projections.

MATHEMATICAL INTERPRETATION

$$p_\theta(s) = \int\int \mu(x,y)\, \delta(s - x\cos\theta - y\sin\theta)\, dx\, dy$$

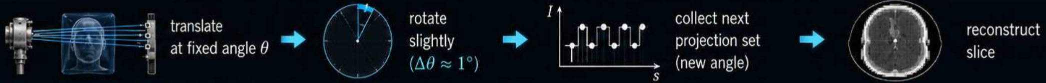
Each translate position measures attenuation along one ray path.


$\mu(x,y)$: attenuation map (linear attenuation coefficient)


s : detector position (translation coordinate)


θ : projection angle (source–detector orientation)


First-generation scanners were slow (minutes per slice) but established the feasibility of cross-sectional X-ray imaging.

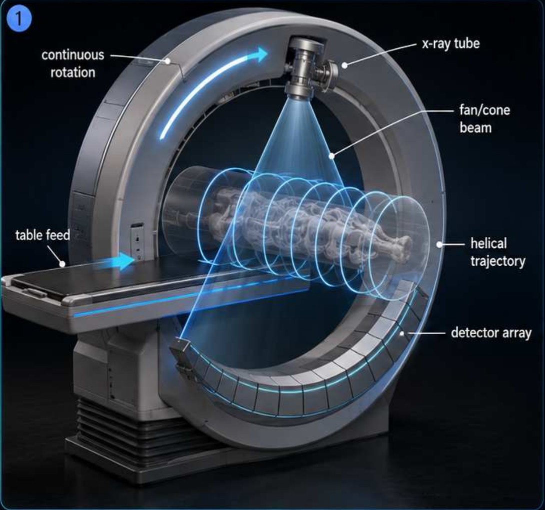


ELEG404/604 *Imaging & Deep Learning*

Spiral / Helical CT: State-of-the-Art Scanning

Continuous gantry rotation plus table motion produces a helical trajectory and fast volumetric imaging.

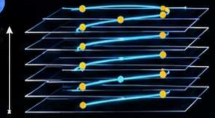
1



2 Why spiral CT is powerful

- Fast whole-volume coverage
- High z-axis sampling
- Fewer motion artifacts
- Supports multiplanar and 3D reconstructions

3 Pitch and sampling



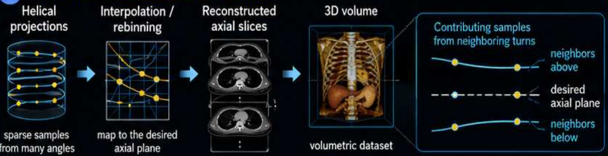
$$\text{pitch} = \frac{\text{table travel per rotation}}{\text{total collimated beam width}}$$

Raw measurements (yellow) do not lie exactly on a single axial plane.

Low pitch = denser sampling

High pitch = faster scan / less dense sampling

4 From helical data to axial slices

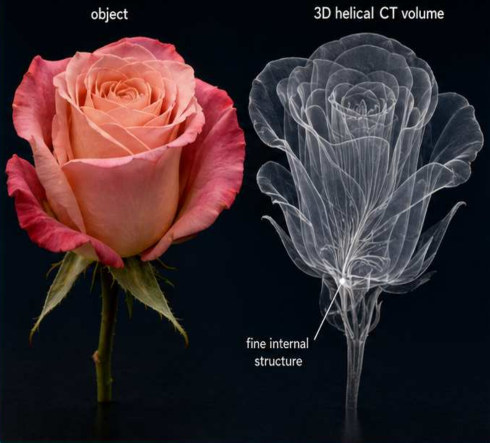


Spiral CT acquires a **3D volume continuously;** slice images are reconstructed by combining nearby helical measurements, enabling **fast, high-resolution volumetric imaging.**

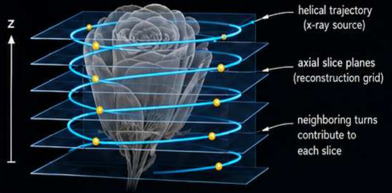
Spiral CT Reconstruction: From Helical Data to High-Resolution Slices

A continuous helical scan can reconstruct detailed axial, coronal, and sagittal views of a complex 3D object.

1 Object: flower (rose blossom)



2 Sparse 3D slice sampling in spiral CT



Measured rays lie on a helix; interpolation reconstructs regular slices.

Helical measurements:

$$g(\theta, s, z)$$

θ: view angle s: detector channel z: table position

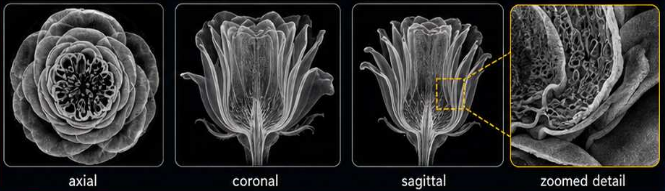
Slice at z_0 uses nearby samples above and below z_0.

$$G(\theta, s; z_0) \approx \mathcal{I}\{g(\theta, s, z) \text{ around } z_0\}$$

$\mathcal{I}$ = interpolation / rebinning

3 Reconstructed slices from helical CT volume

helical acquisition → interpolation / filtered reconstruction → reconstructed slices



Takeaway: Spiral CT converts a **continuously sampled helix** into a **regularly sampled 3D volume**, enabling thin-slice reconstructions with **excellent detail.**

First-Generation CT Acquisition: Translate-Rotate Pencil-Beam Geometry

How the original EMI head scanner formed projections by shifting a pencil beam, then rotating and repeating.

ACQUISITION SEQUENCE

1. At angle θ, translate the pencil beam across the head.
2. Each position s records one line integral (one projection sample).
3. Rotate the source-detector pair by a small angle $\Delta\theta \approx 1°$.
4. Repeat until projections are collected over many angles.

FIRST-GENERATION TRANSLATE–ROTATE ACQUISITION

1) TRANSLATE AT FIXED ANGLE θ

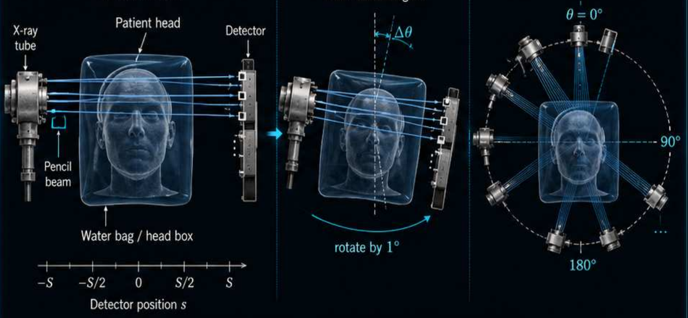
Source–detector pair translates linearly across the head.

2) ROTATE BY A SMALL ANGLE $\Delta\theta \approx 1°$

Then translate again.

3) REPEAT FOR MANY ANGLES

Collect projections over ~180°.



At each angle θ, translation samples many rays (line integrals).
Rotation by small steps builds a complete set of projections.

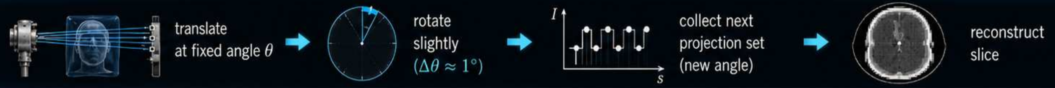
MATHEMATICAL INTERPRETATION

$$p_\theta(s) = \int\int \mu(x,y)\,\delta(s - x\cos\theta - y\sin\theta)\,dx\,dy$$

Each translate position measures attenuation along one ray path.

- $\mu(x,y)$: attenuation map (linear attenuation coefficient)
- s: detector position (translation coordinate)
- θ: projection angle (source–detector orientation)

First-generation scanners were slow (minutes per slice) but established the feasibility of cross-sectional X-ray imaging.



The Invention of CT: A Nobel Prize Story

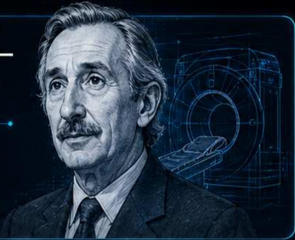
From theoretical tomography to the first clinical scanner and the 1979 Nobel Prize.

Allan M. Cormack —
mathematical theory of reconstruction

- Developed the theoretical basis of transmission tomography
- Key work began in 1956–1957

Godfrey N. Hounsfield —
practical CT scanner inventor

- Built the first successful clinical head CT scanner at EMI
- First patient brain scan in 1971



1956–1957

$\int p_0(s)ds$

Cormack develops the theoretical basis of transmission tomography.

1971

First clinical head CT scan (patient at Atkinson Morley's Hospital, London).

1972

EMI CT scanner introduced publicly.

1979

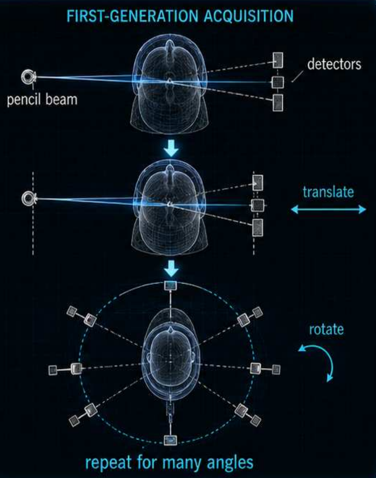
Nobel Prize in Physiology or Medicine awarded jointly to Allan Cormack and Godfrey Hounsfield.

CT emerged from the union of **mathematics** and **engineering** —
theory from Cormack, a working scanner from Hounsfield.

The First CT Scanner: EMI Head Scanner

A first-generation translate–rotate system designed for brain imaging.

Water-filled head box
(water bag)



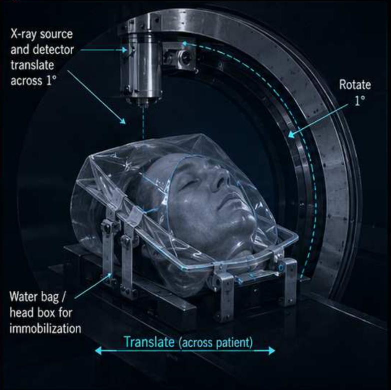
- Head-only scanner
- Translate–rotate geometry
- Pencil beam + detector pair
- About 5 minutes for acquisition
- About 2.5 hours to reconstruct the earliest images

The design was slow by modern standards, but it proved that cross-sectional X-ray imaging was possible.

The First Clinical Brain CT

Atkinson Morley Hospital, London — the first patient brain scans transformed neuroradiology.

1 EARLY ACQUISITION: TRANSLATE–ROTATE SCANNER (1971/72)



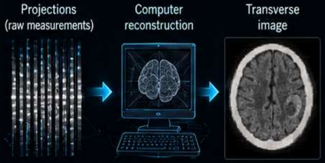
2 EARLY RECONSTRUCTED TRANSVERSE SLICE

Early reconstructed transverse slice (first clinical brain CT)

3 WHY IT MATTERED

- Cross-sectional imaging revealed lesions hidden in conventional radiographs
- The first successful clinical scans demonstrated brain tumors and hemorrhage clearly
- The impact on diagnosis was immediate

FROM PROJECTIONS TO IMAGE



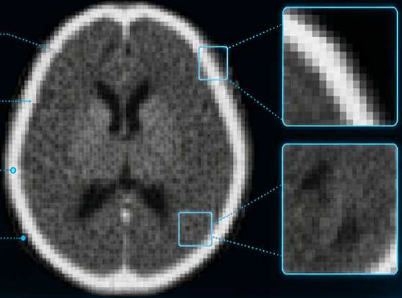
For the first time, physicians could see inside the brain slice by slice.

Then and Now: Early CT vs Modern Brain CT

From coarse, slow head scans to fast, high-resolution volumetric imaging.

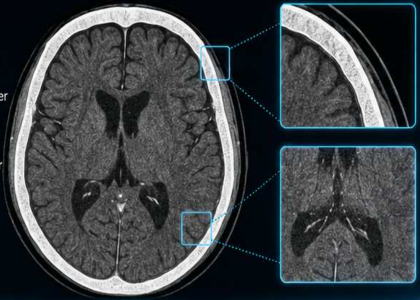
Early CT (1971–1972)

- minutes to scan
- coarse matrix
- limited contrast
- head only



Modern CT

- sub-second gantry rotation
- sub-millimeter slices
- multidetector spiral CT
- high diagnostic detail



	SCANNER GEOMETRY	SPEED	IMAGE QUALITY	COVERAGE	CLINICAL IMPACT
EARLY CT (1971–1972)	Translate–Rotate Single–Detector	Minutes per slice	80×80 matrix 10 mm slice thickness	Head only	Limited applications Research tool
MODERN CT	Rotate–Rotate Multidetector Array	Sub-second rotation Whole brain in < 2 s	512×512 (or higher) 0.5–1 mm slice thickness	Head to toe Volumetric coverage	Broad clinical use Standard of care

The first CT changed medicine — modern CT turned it into a routine, high-performance imaging modality.