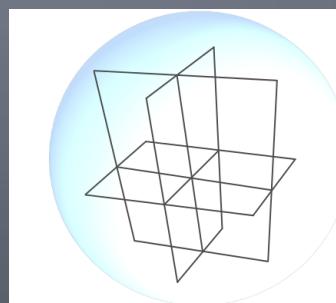


# Tomographic Reconstruction

CS 754  
Ajit Rajwade



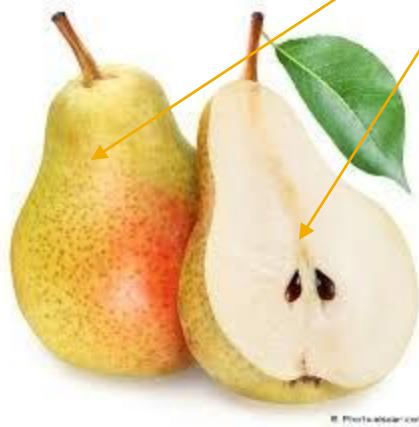
**ViGIL**  
VISION, GRAPHICS  
& IMAGING LABORATORY

# Aim of Tomographic Reconstruction

- It is the task of reconstructing a 2D image (object) from its 1D projections, or a 3D image (object) from its 2D projections.
- What is a projection (also called tomographic projection)?
- It is defined as the Radon transform of the image in a particular direction (see next slide).

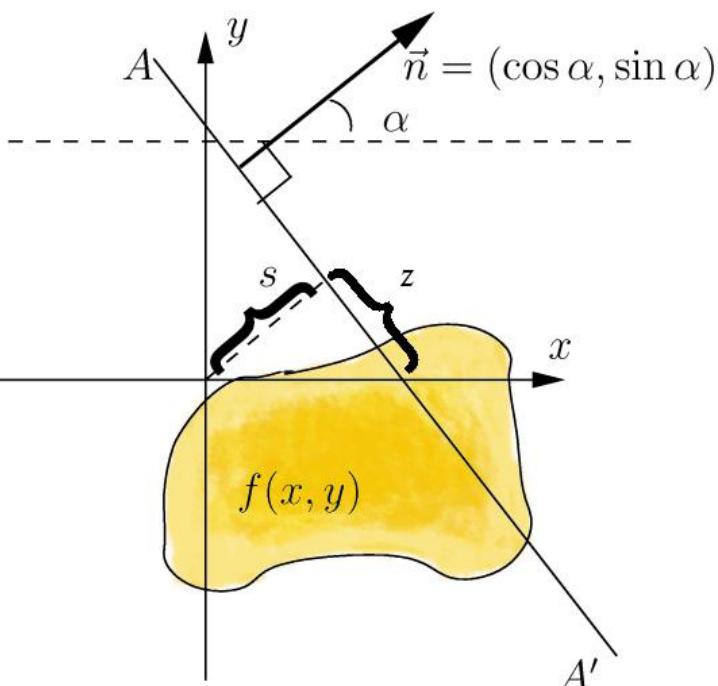
# Aim of Tomographic Reconstruction

- Involves estimation of **inner** structural details of an object (eg: internal organs of a person) as opposed to just the surface (as in photographic imaging).



CT scan of a person's lung

# Radon transform



Imagine a line was drawn through the 2D image in a certain direction  $\alpha$ , and you integrated the intensity values along that line.

Now you repeat this for lines parallel to the original one but at different offsets.

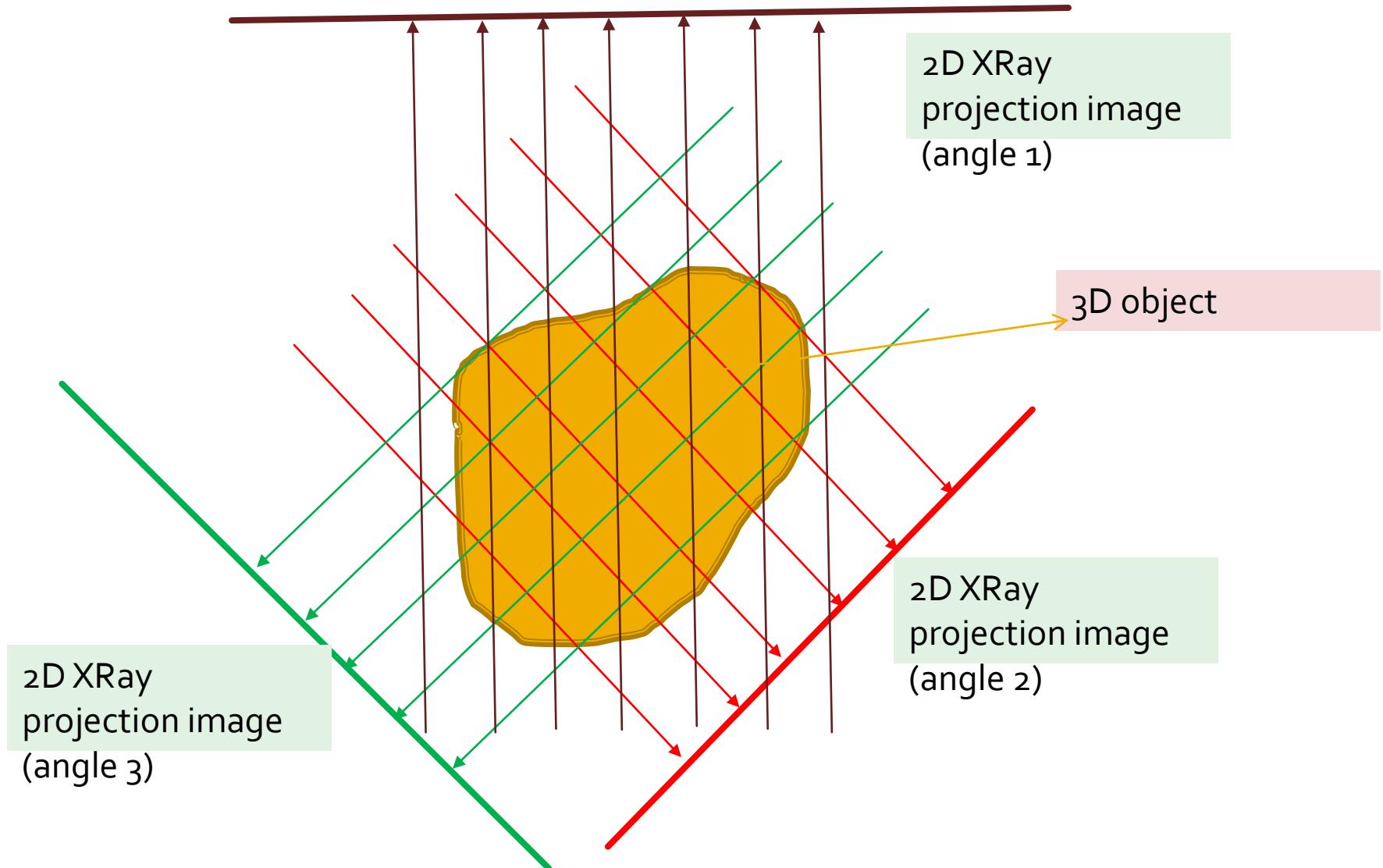
Each such summation produces a bin of the tomographic projection.

The collection of bins form a 1D array which is called the tomographic projection or the Radon transform of the object in the direction  $\alpha$ .

[https://en.wikipedia.org/wiki/Radon\\_transform](https://en.wikipedia.org/wiki/Radon_transform)

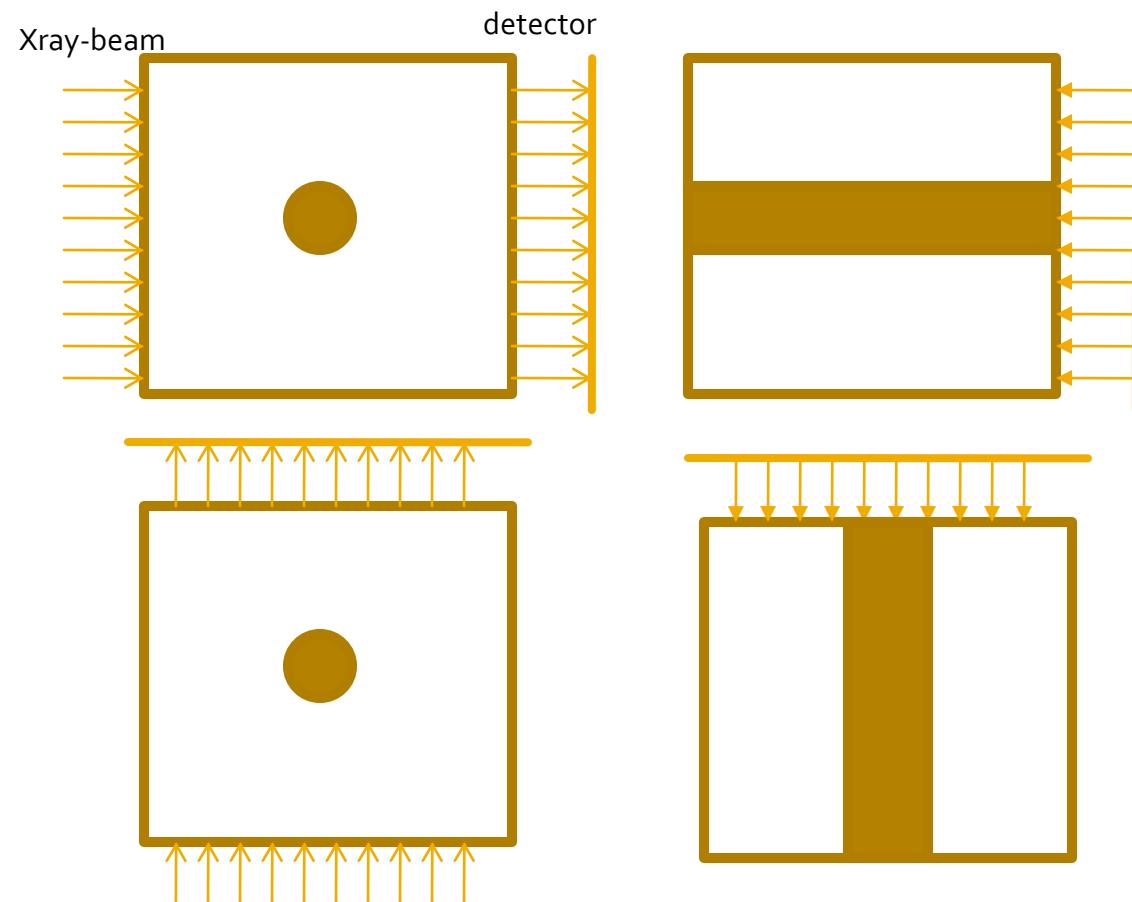
# Reconstruction from the Radon Transform

- The Radon transform computation can be similarly repeated for multiple angles.
- The aim of tomographic reconstruction is to reconstruct the 2D image from a collection of its 1D projections in different angles.



Reconstruction of the 3D object is performed from its many 2D X-ray projection images, each in a different **known** direction. This is what happens typically inside a CT machine. The reconstruction quality is better if there are more projection images.

# Where in practice do you see the Radon transform?

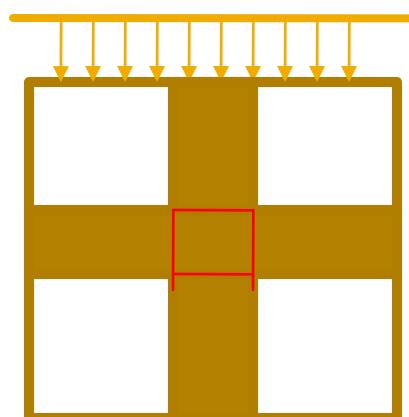
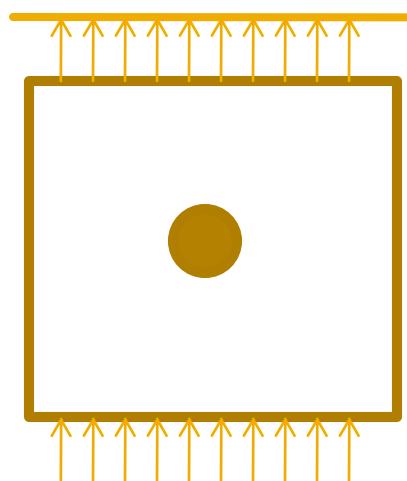
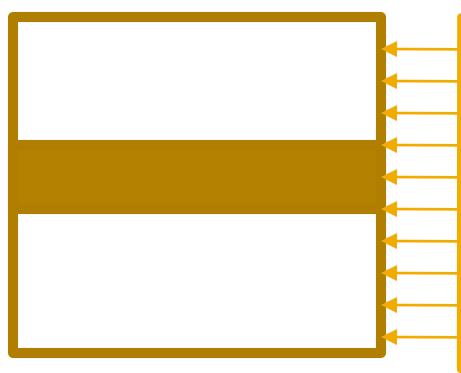
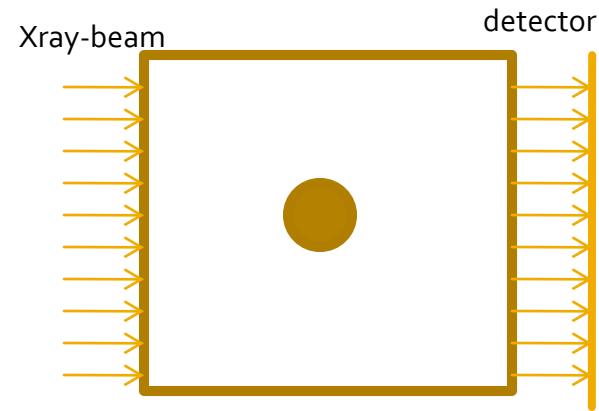


The degree of absorption of X-Rays at each point is measured by an X-Ray absorption detector. This detector produces a 1D signal whose amplitude/intensity is directly proportional to the extent of absorption.

Any point in the signal = sum of the absorptivity values across the path of a single ray in the X-Ray beam that spatially maps onto that point.

The image is a simplification of a set of real biological tissues: example, an organ/tumor surrounded by a background consisting of soft, uniform tissue. The set of tissues is bombarded with an X-Ray beam. The tumor has higher rate of absorption as compared to the surrounding tissue.

# Where in practice do you see the Radon transform?



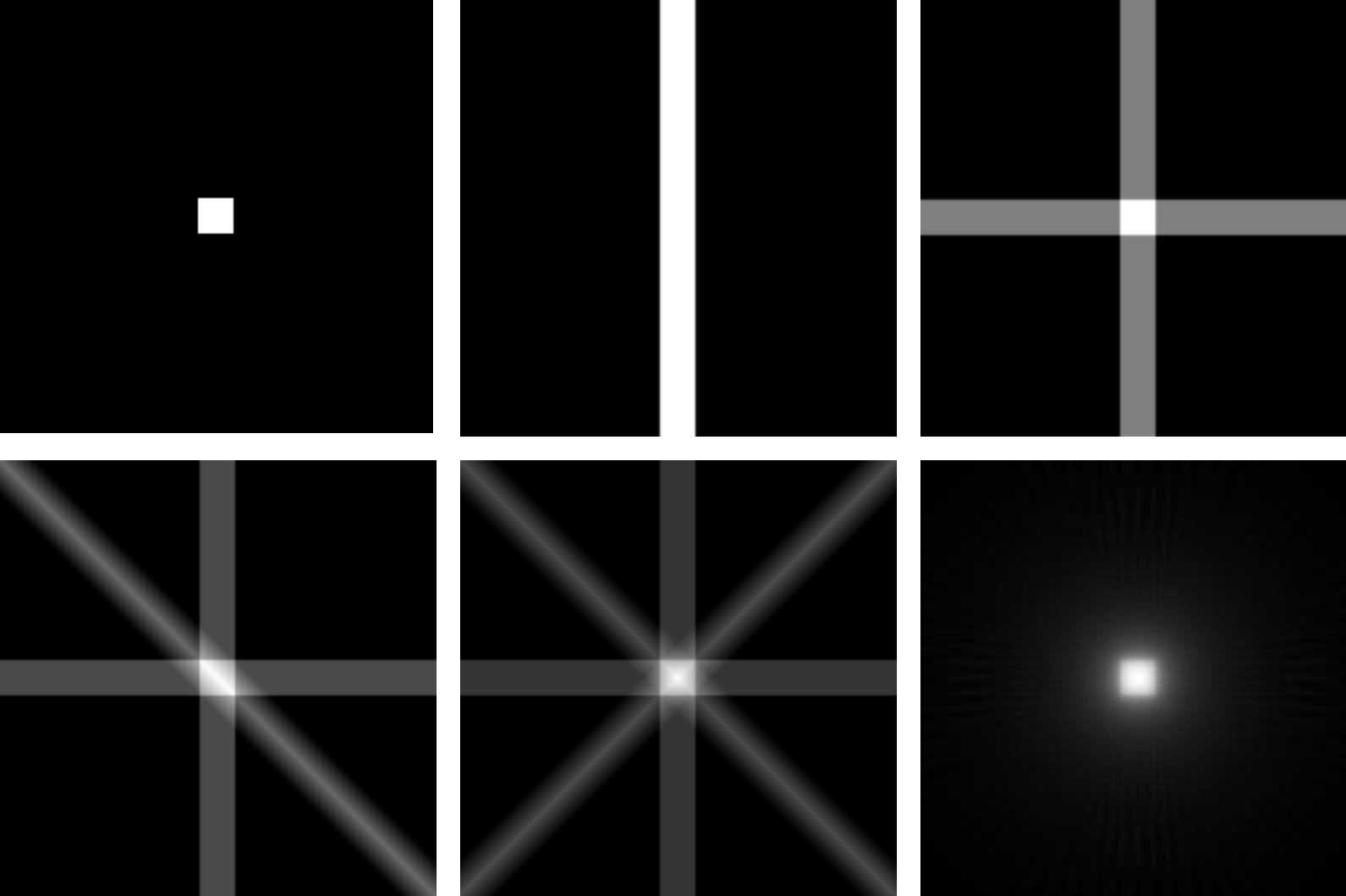
Sum-total of the two  
back-projections

Given the 1D signal (called a **projection signal**), we try to reconstruct the original 2D image by smearing backwards along the direction of projection. This is called as **back-projection**.

The 1D signal that was measured is duplicated along the columns of the image to be estimated (see the directions marked in yellow).

# Introduction

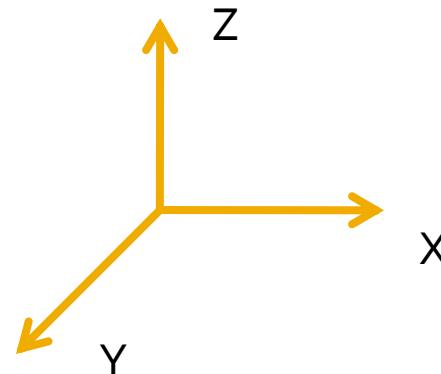
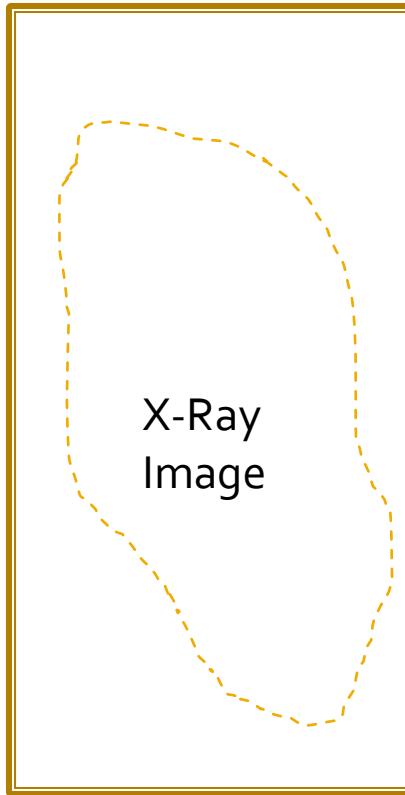
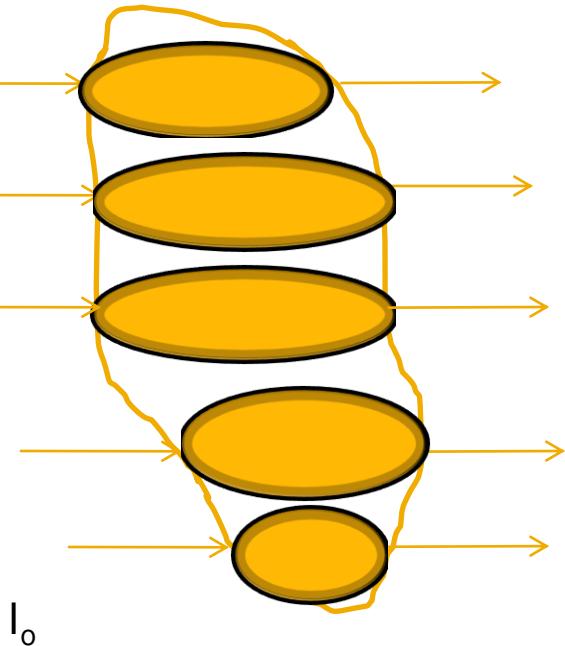
- Given projections in  $K$  different directions, we can hope to reconstruct the original image by performing back-projection along all these directions, and *adding up* the results.
- Back-projection refers to smearing the 1D projection back across the 2D image, i.e. *duplicating* the 1D signal across the image in a direction opposite to the direction of projection.
- The shape of the object will be approximated better and better as  $K$  increases.



Even with many (32) back-projections,  
there is a **blur** artifact in the  
reconstruction. This is called as a “halo  
effect”.

# Reconstructing 3D objects

- A 3D object is illuminated with X-Ray beams to produce a projection which is a 2D-image.
- Changing the direction of the X-ray beam will produce another image. This set of images when back-projected will yield the 3D volume/object.
- However in olden Computed Tomography (CT), each slice of the volume is measured at a time. A slice is a 2D entity obtained by cutting the 3D volume transversely through a plane parallel to the XY plane.
- This allows for the employment of a smaller number of detectors at a time, for the same resolution of the measurement.



$$I = I_0 \exp\left(-\int_L f(x, y, z)dL\right)$$

$$\log\left(\frac{I_0}{I}\right) = \int_L f(x, y, z)dL = g(\rho, \theta)$$

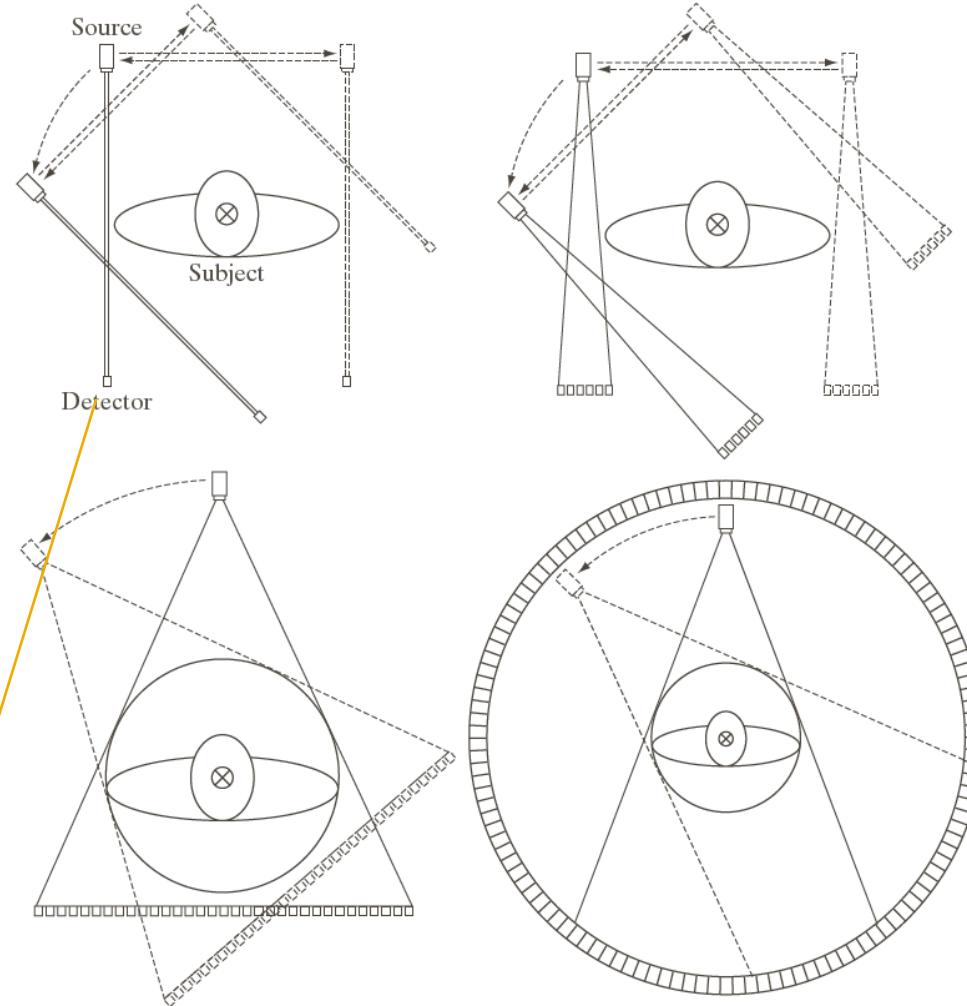
$I_0$  = intensity of the X-ray beam from the source  
 $I$  = intensity of the X-ray beam as measured by the detector, given by Beer's law

# Tomography and medical imaging

- Tomography is useful in medical imaging, because different tissues (eg: tissues in the bone, lungs, etc.) attenuate the X-rays to different extents.
- This allows for good medical diagnosis.

a  
b  
c  
d

**FIGURE 5.35** Four generations of CT scanners. The dotted arrow lines indicate incremental linear motion. The dotted arrow arcs indicate incremental rotation. The cross-mark on the subject's head indicates linear motion perpendicular to the plane of the paper. The double arrows in (a) and (b) indicate that the source/detector unit is translated and then brought back into its original position.

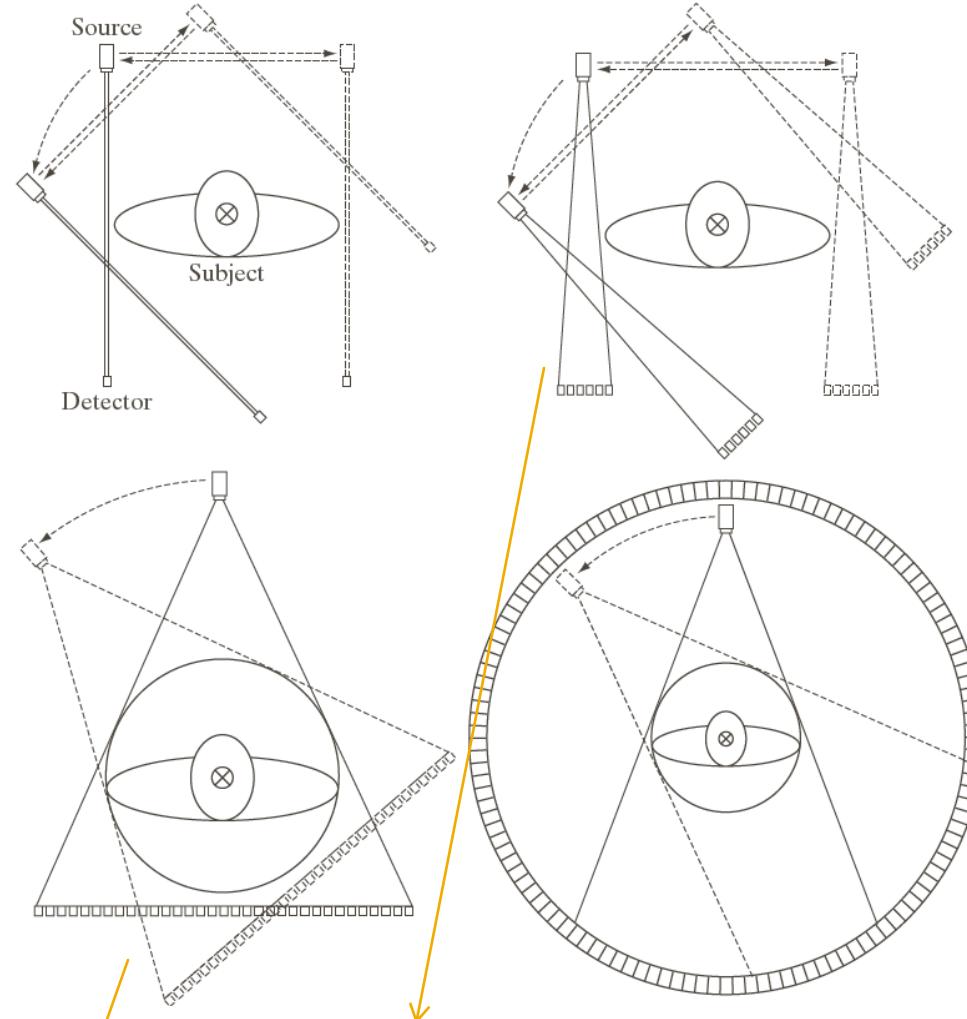


Source of image: Book by Gonzalez, 3<sup>rd</sup> edition

Given a direction  $\Theta$ , the source and detector pair move along that direction in fixed steps (i.e. variation in  $\rho$ ). The distance between source and detector is constant. At each step, the source sends out an X-Ray beam onto the subject, and the projection value is recorded on the detector and stored in a computer. This process is repeated for several values of  $\Theta$ . In the end, we record a single 2D slice. Now, the subject is moved in a direction perpendicular to the plane of the source-detector pair, and another 2D slice is recorded. This is called 1<sup>st</sup> generation CT.

a  
b  
c  
d

**FIGURE 5.35** Four generations of CT scanners. The dotted arrow lines indicate incremental linear motion. The dotted arrow arcs indicate incremental rotation. The cross-mark on the subject's head indicates linear motion perpendicular to the plane of the paper. The double arrows in (a) and (b) indicate that the source/detector unit is translated and then brought back into its original position.



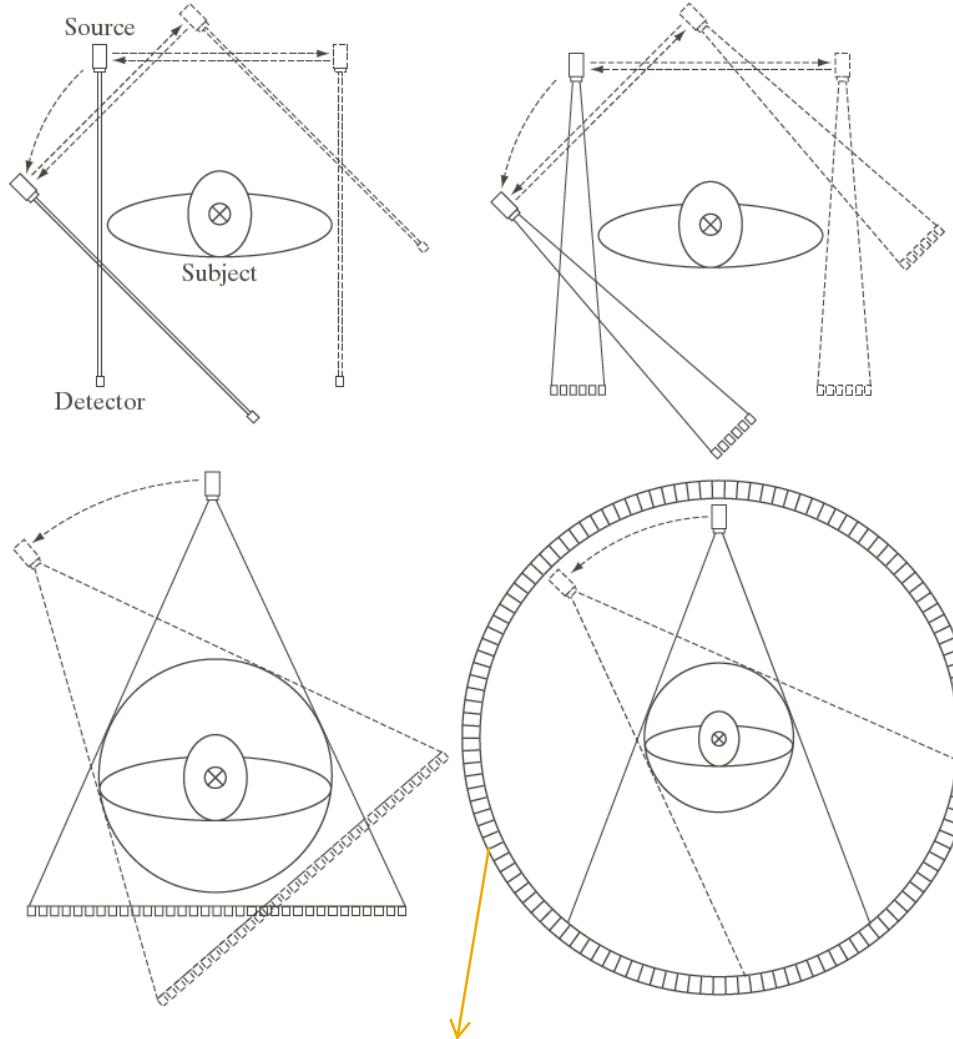
Source of image: Book by Gonzalez, 3<sup>rd</sup> edition

In second-generation CT, the source sends out an X-ray beam in the form of a cone (also called fan). There are more detectors needed, but fewer translations are required to record a single 2D slice (rotations are needed just as before).

In third-generation CT, the source sends out an X-ray beam in the form of a large cone. The number of detectors is long enough to cover the field of view, so no translations are needed to record a single 2D slice (only rotations are needed).

a  
b  
c  
d

**FIGURE 5.35** Four generations of CT scanners. The dotted arrow lines indicate incremental linear motion. The dotted arrow arcs indicate incremental rotation. The cross-mark on the subject's head indicates linear motion perpendicular to the plane of the paper. The double arrows in (a) and (b) indicate that the source/detector unit is translated and then brought back into its original position.



In fourth-generation CT, there is a large number of detectors arranged in the form of a circle. The source alone rotates.

Source of image: Book by Gonzalez, 3<sup>rd</sup> edition

# Newer advancements

- Today's scanners follow cone-beam or helical CT geometry.
- A large stack of slices (a sub-volume) is imaged via such cone beams from different directions.
- The entire sub-volume is reconstructed at a time.
- The object may be moved in a perpendicular fashion to image other sub-volumes.
- The reconstruction of different sub-volumes will be stitched/merged together.

# Aim of Tomography

- To estimate the full 3D structure of an object from its projections.
- The projections are directly measured, the 3D structure is estimated.
- Applications: **medical imaging, industrial** applications such as fault detection in machines, observation of **plant roots, remote sensing** (observation of underground objects or phenomena), seismology, geology.

# Representing Projections

- The complete set of projections for several different values of the parameters  $\rho$  and  $\Theta$  gives:

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

Dirac delta function  
Eqn. of line in “normal form”

- This is called the **Radon Transform** of  $f$ . Its discrete version is:

$$R_\theta(f) = g(\rho, \theta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho)$$

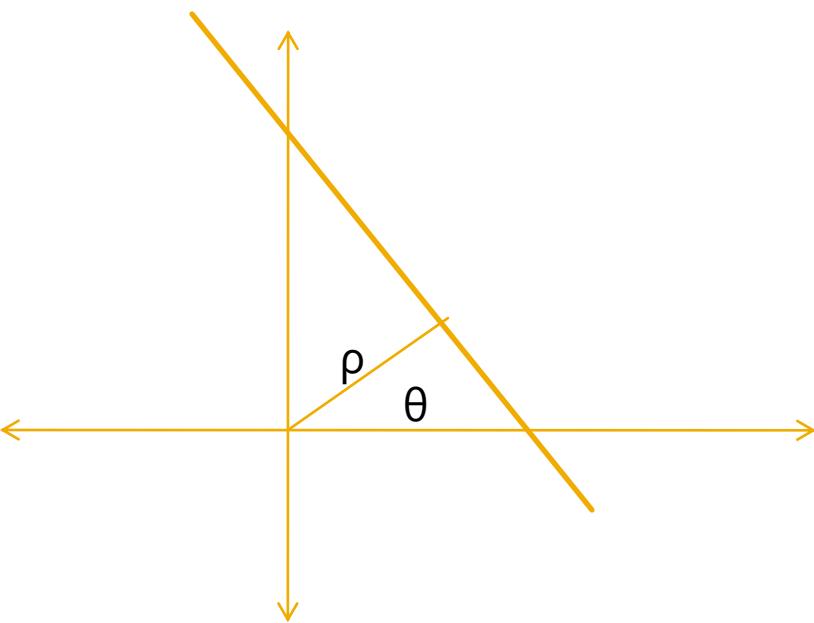
One single projection vector is obtained with a fixed value of  $\Theta$ , but varying  $\rho$ .

Kronecker delta function

# Parameterization of a direction/line

The direction of projection is denoted  $L$ , and  $dL$  is an infinitesimally small element along  $L$ .  $L$  is parameterized as follows (the “normal representation of the line”):

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



# Example:

$$f(x, y) = A, \text{if } x^2 + y^2 \leq r^2, \text{else } 0$$

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

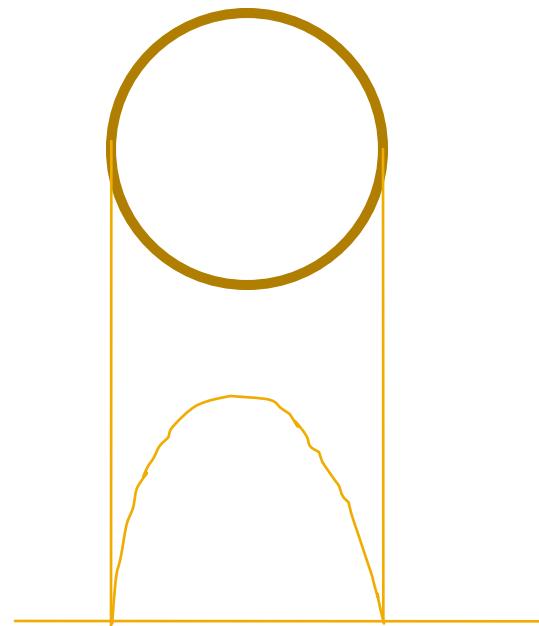
$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta(x - \rho) dx dy = \int_{-\infty}^{+\infty} f(\rho, y) dy$$

$$\begin{aligned} &= \int_{-\sqrt{r^2 - \rho^2}}^{+\sqrt{r^2 - \rho^2}} f(\rho, y) dy \\ &= \int_{-\sqrt{r^2 - \rho^2}}^{+\sqrt{r^2 - \rho^2}} Ady \\ &= 2A\sqrt{r^2 - \rho^2} \end{aligned}$$

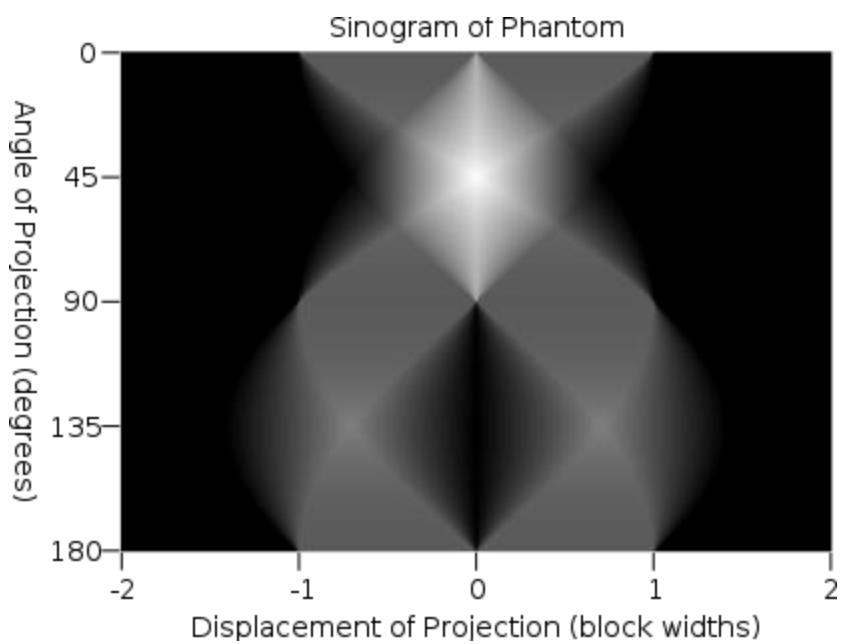
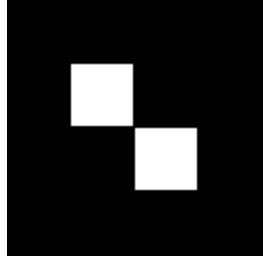
If  $|\rho| \leq r$

If  $|\rho| > r, g(\rho, \theta) = 0$

Since this is a circular object, the projections are identical in all directions. Hence WLOG, we consider theta = 0, due to which the argument to delta gets simplified



Note: there is no closed form for the Radon transform for most functions  $f(x, y)$ .



Sinogram (a radon transform plotted as an image in a  $(\rho, \Theta)$  grid.)

[https://en.wikipedia.org/wiki/Radon\\_transform](https://en.wikipedia.org/wiki/Radon_transform)

# Back-projection

Radon transform:  
obtained by  
sampling several  
different angles

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



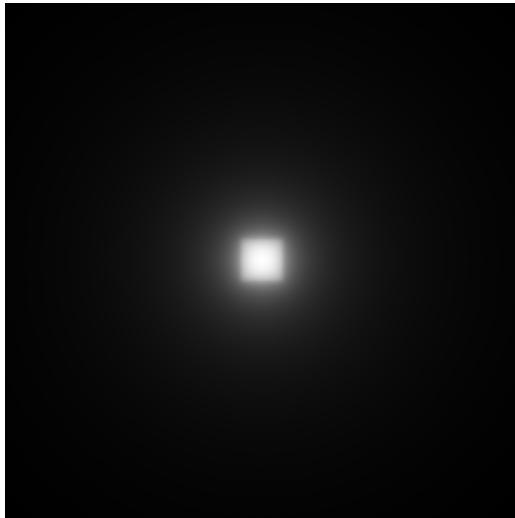
Fix the angle  $\Theta_k$  and for all  $x$  and  $y$ , compute the value of  $\rho$ . Copy  $g(\rho, \Theta_k)$  to  $\hat{f}_{\theta_k}(x, y)$ , which is the image obtained when you back-project along angle  $\Theta_k$ .

$$\hat{f}_{\theta_k}(x, y) = g(\rho, \theta_k) = g(x \cos \theta_k + y \sin \theta_k, \theta_k)$$



$$\hat{f}(x, y) = \int_0^\pi g(x \cos \theta + y \sin \theta, \theta) d\theta = \int_0^\pi \hat{f}_\theta(x, y) d\theta \approx \sum_{k=1}^K \hat{f}_{\theta_k}(x, y) = \sum_{\theta=0}^\pi \hat{f}_\theta(x, y)$$

The back-projection operator is NOT the same as the inverse of the Radon transform! So this does not yield back the true signal  $f(x, y)$ , but the signal  $f(x, y)$  blurred with the kernel  $(x^2 + y^2)^{-0.5}$ .  
**More on this a few slides down, when we do filtered back-projection.**



The blur is a painful consequence of (1) discretization of the angle  $\Theta$ , and (2) the inherent blurring with the kernel  $(x^2+y^2)^{-0.5}$ .

These images are reconstructed at 0.5 degree changes in  $\Theta$ .

How do we get rid of this blur? Wait for a few slides!

# Fourier transform of the Radon Transform

- The Radon transform is given as:

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

- Its 1D Fourier transform w.r.t.  $\rho$  (keeping  $\Theta$  fixed to some value) is given by:

$$G(\mu, \theta) = \int_{-\infty}^{+\infty} g(\rho, \theta) \exp(-j2\pi\mu\rho) d\rho$$

$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) \exp(-j2\pi\mu\rho) dx dy d\rho$$

$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \left[ \int_{-\infty}^{+\infty} \delta(x \cos \theta + y \sin \theta - \rho) \exp(-j2\pi\mu\rho) d\rho \right] dx dy$$

$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \exp(-j2\pi\mu(x \cos \theta + y \sin \theta)) dx dy$$

$G(\mu, \theta)$  is the Fourier transform of the projection of  $f(x, y)$  along some direction  $\Theta$ .

# Fourier transform of the Radon Transform

- Its 1D Fourier transform w.r.t.  $\rho$  (keeping  $\Theta$  fixed) is given by:

$$G(\mu, \theta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \exp(-j2\pi(x\mu \cos \theta + y\mu \sin \theta)) dx dy$$

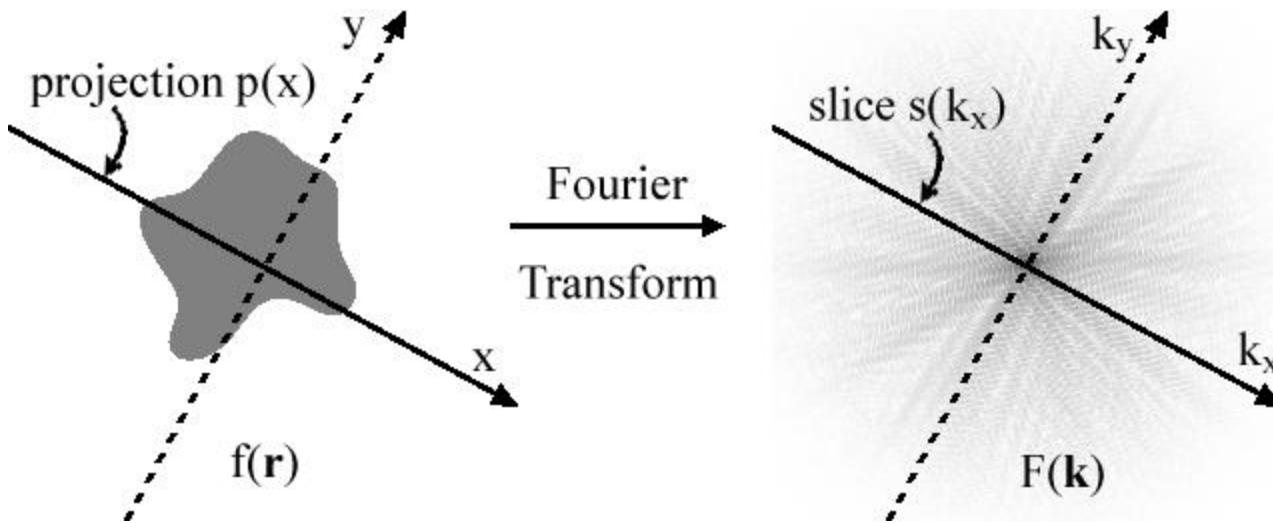
$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \exp(-j2\pi(xu + yv)) dx dy$$

where we define  $u = \mu \cos \theta, v = \mu \sin \theta$

$$\therefore G(\mu, \theta) = [F(u, v)]_{u=\mu \cos \theta, v=\mu \sin \theta} = F(\mu \cos \theta, \mu \sin \theta)$$

The RHS of this equation is a slice of the 2D Fourier transform of  $f(x, y)$ , i.e.  $F(u, v)$ , along the angle  $\Theta$  in the frequency plane, and passing through the origin

This equation above is called the **Projection Slice Theorem** or the **Fourier Slice Theorem**. It states that the Fourier transform of a projection of the 2D object along some direction  $\Theta$  (i.e.  $G(\mu, \Theta)$ ) is equal to a slice of the 2D Fourier transform of the object along the same direction  $\Theta$  (in the frequency plane), passing through the origin.



The **Projection Slice Theorem** or the **Fourier Slice Theorem** states that the following two are equivalent:

- (1) Project a 2D object along a certain direction  $\mathbf{d}$ . Take its 1D Fourier Transform called as  $\mathbf{F}_1$ .
- (2) Compute the 2D Fourier transform of the same object. Take a slice of this Fourier transform along a direction parallel to  $\mathbf{d}$  (but in the frequency plane). Call this slice as  $\mathbf{F}_2$ .

Now  $\mathbf{F}_1 = \mathbf{F}_2$ .

Source of image: [https://en.wikipedia.org/wiki/Projection-slice\\_theorem](https://en.wikipedia.org/wiki/Projection-slice_theorem)

# Filtered Back-projection

- Consider the 2D inverse Fourier transform of  $F(u,v)$ , giving us:

$$f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u, v) \exp(j2\pi(xu + yv)) du dv$$

- Consider  $u = \mu \cos(\Theta)$ ,  $v = \mu \sin(\Theta)$ . Then:

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} F(\mu \cos \theta, \mu \sin \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta$$
$$\mu = \sqrt{u^2 + v^2}$$

Note: we are doing a change of variables from  $(u,v)$  to  $(\mu,\Theta)$ . Hence  $du dv = \mu d\mu d\theta$ .

# Filtered Back-projection

- By projection slice theorem, this becomes:

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta$$

- Further simplification will give the following (see next slide)

# Filtered Back-projection

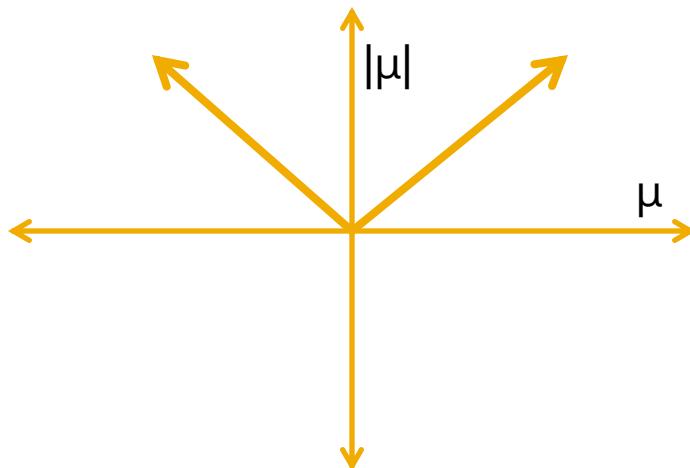
$$\begin{aligned}
 f(x, y) &= \int_0^{\pi} \int_0^{+\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta + \int_{\pi}^{2\pi} \int_0^{+\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta \\
 &= \int_0^{\pi} \int_0^{+\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta + \int_0^{\pi} \int_0^{+\infty} \underbrace{G(\mu, \theta + \pi)}_{?} \exp(j2\pi\mu(x \cos(\theta + \pi) + y \sin(\theta + \pi))) \mu d\mu d\theta \\
 &= \int_0^{\pi} \int_0^{+\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta + \int_0^{\pi} \int_0^{+\infty} \underbrace{G(-\mu, \theta)}_{?} \exp(j2\pi(-\mu)(x \cos \theta + y \sin \theta)) \mu d\mu d\theta \\
 &= \int_0^{\pi} \int_0^{+\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta + \int_0^{\pi} \int_0^{+\infty} G(-\mu, \theta) \exp(j2\pi(-\mu)(x \cos \theta + y \sin \theta)) (-\mu) d\mu d\theta \\
 &= \int_0^{\pi} \int_0^{+\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta + \int_0^{\pi} \int_0^{-\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta \\
 &= \int_0^{\pi} \int_0^{+\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) \mu d\mu d\theta + \int_0^{\pi} \int_{-\infty}^0 G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) |\mu| d\mu d\theta \\
 &= \int_0^{\pi} \int_{-\infty}^0 G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) |\mu| d\mu d\theta
 \end{aligned}$$

# Filtered Back-projection

$$\begin{aligned}\therefore f(x, y) &= \int_0^{\pi} \int_{-\infty}^{+\infty} |\mu| G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) d\mu d\theta \\ &= \int_0^{\pi} \left( \int_{-\infty}^{+\infty} |\mu| G(\mu, \theta) \exp(j2\pi\mu\rho) d\mu \right) d\theta\end{aligned}$$



This is a 1D Inverse Fourier Transform with an added term  $|\mu|$  (a ramp filter). But this function is not integrable as  $|\mu|$  grows unboundedly. Hence the inverse Fourier transform does not exist!



# Filtered Back-projection versus back-projection

$$\begin{aligned}\therefore f(x, y) &= \int_0^{\pi} \int_{-\infty}^{+\infty} |\mu| G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) d\mu d\theta \\ &= \int_0^{\pi} \left[ \int_{-\infty}^{+\infty} |\mu| G(\mu, \theta) \exp(j2\pi\mu\rho) d\mu \right] d\theta\end{aligned}$$



This is a 1D Inverse Fourier Transform with an added term  $|\mu|$  (**a ramp filter**). But this function is not integrable as  $|\mu|$  grows unboundedly. Hence the inverse Fourier transform does not exist!

Now suppose that additional term  $|\mu|$  were absent. We would then obtain simple back-projection:

$$\begin{aligned}\hat{f}(x, y) &= \int_0^{\pi} \int_{-\infty}^{+\infty} G(\mu, \theta) \exp(j2\pi\mu(x \cos \theta + y \sin \theta)) d\mu d\theta \\ &= \int_0^{\pi} g(\rho, \theta) d\theta = F^{-1} \left( \frac{F(f)(\mu \cos \theta, \mu \sin \theta)}{|\mu|} \right)(x, y) = \left( f * F^{-1} \left( \frac{1}{|\mu|} \right) \right)(x, y) = \\ &= \left( f * F^{-1} \left( \frac{1}{\sqrt{u^2 + v^2}} \right) \right)(x, y) = [f * (x^2 + y^2)^{-0.5}](x, y), u = \mu \cos \theta, v = \mu \sin \theta\end{aligned}$$

By Fourier convolution theorem

# Filtered Back-projection algorithm: version 1

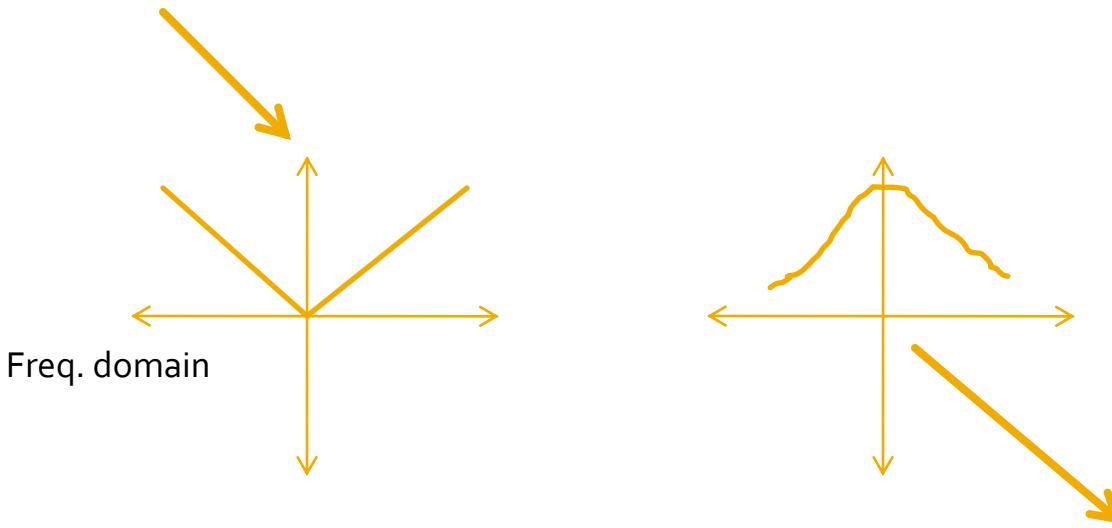
- The X-Ray or CT machine gives you projections (i.e. Radon transforms) of the data in different directions.
- Perform simple back-projection to get a 2D image.
- Find the 2D Fourier transform of the back-projected 2D image, multiply it with  $(u^2+v^2)^{0.5}$  and find the 2D inverse Fourier Transform of the product.

# Filtered Back-Projection

- In practice, the ramp filter is multiplied in the frequency domain by a **windowing function** (equivalent to convolution with the corresponding kernel in the spatial domain).
- It has the advantage of controlling some amount of noise as it band-limits the ramp filter.
- The simplest windowing function is a rect filter in the frequency domain.
- This formula is called the **Ramachandran Lakshminarayanan (Ram-Lak) filter**.

<http://www.pnas.org/content/68/9/2236>

Ram-Lak filter: ramp multiplied by rect  
(both in frequency domain)



Other windowing functions can also be used!  
(eg: Hamming window, which is a truncated cosine).  
These windowing functions allow lesser ringing artifacts  
as compared to the rect windowing function.

$$f(x, y) = \int_0^{\pi} \left( \int_{-\infty}^{+\infty} |\mu| \text{rect}(\mu D) G(\mu, \theta) \exp(j2\pi\mu\rho) d\mu \right) d\theta$$

Ram-Lak filter

$$f(x, y) = \int_0^{\pi} \left( \int_{-\infty}^{+\infty} |\mu| H(\mu) G(\mu, \theta) \exp(j2\pi\mu\rho) d\mu \right) d\theta$$

$$H(\mu) = c + (c-1) \cos\left(\frac{2\pi\mu}{M-1}\right), 0 \leq \mu \leq M-1$$

= 0, otherwise

Ram-Lak filter  
with Hamming  
window

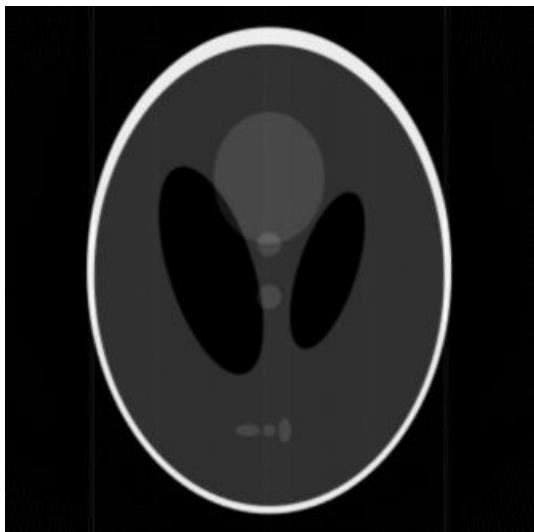
# Filtered Back-projection algorithm: version 2 (faster)

- Compute the 1D Fourier transform of each projection of the object (the projections are measured by the X-Ray or CT machine).
- Multiply each Fourier transform by the ramp filter  $|\mu|$  multiplied by a windowing function (such as rect, or Hamming).
- Obtain the 1D inverse Fourier transform of the result.
- Sum up over all such results (one each per projection angle) from the previous step to give the final image.



$$\begin{aligned}f(x, y) &= \int_0^{\pi} g(x \cos \theta + y \sin \theta, \theta) d\theta \\&= \int_0^{\pi} \int_{-\infty}^{+\infty} G(\mu, \theta) \exp(j2\pi\mu\rho) d\mu d\theta\end{aligned}$$

Using backprojection – 180 angles



$$\therefore f(x, y) = \int_0^{\pi} \left( \int_{-\infty}^{+\infty} |\mu| G(\mu, \theta) \exp(j2\pi\mu\rho) d\mu \right) d\theta$$

Using filtered backprojection – 180 angles (Ram-Lak filter)

# Tomography in MATLAB

- The 2D Radon transform can be computed in MATLAB using the function *radon*.
- Filtered back-projection with a choice of various filters
  - Ram-Lak, Hamming, Hanning, etc., is implemented in MATLAB in a function called *iradon*.
- 3D transforms generally work slice by slice for parallel beam projections and are implemented in toolboxes written by various authors:  
<https://web.eecs.umich.edu/~fessler/code/index.html>

# Tomography and Compressed Sensing

- In most tomography applications, the number of angles of projection is limited due to cost, energy and health considerations.
- So the problem is of practical interest in the “angle starved” case.
- Hence tomography can be considered a compressed sensing problem.

# Tomography and Compressed sensing

- The filtered backprojection algorithm does not exploit one important property of images – their sparsity or compressibility in standard bases such as DCT.
- Instead one can use CS principles to frame the tomographic reconstruction problem as follows:

$$E(\beta) = \|y - RU\beta\|^2 + \lambda \|\beta\|_1$$

Vector created by concatenating 1D projections in various angles

Forward model matrix – representing Radon transform computation in those angles

# Tomography and Compressed sensing

- Instead one can use CS principles to frame the tomographic reconstruction problem as follows:

$$E(\beta) = \|\mathbf{y} - \mathbf{R}\mathbf{U}\beta\|^2 + \lambda \|\beta\|_1$$

Vector created by concatenating 1D projections in various angles

Forward model matrix – representing Radon transform computation in those angles

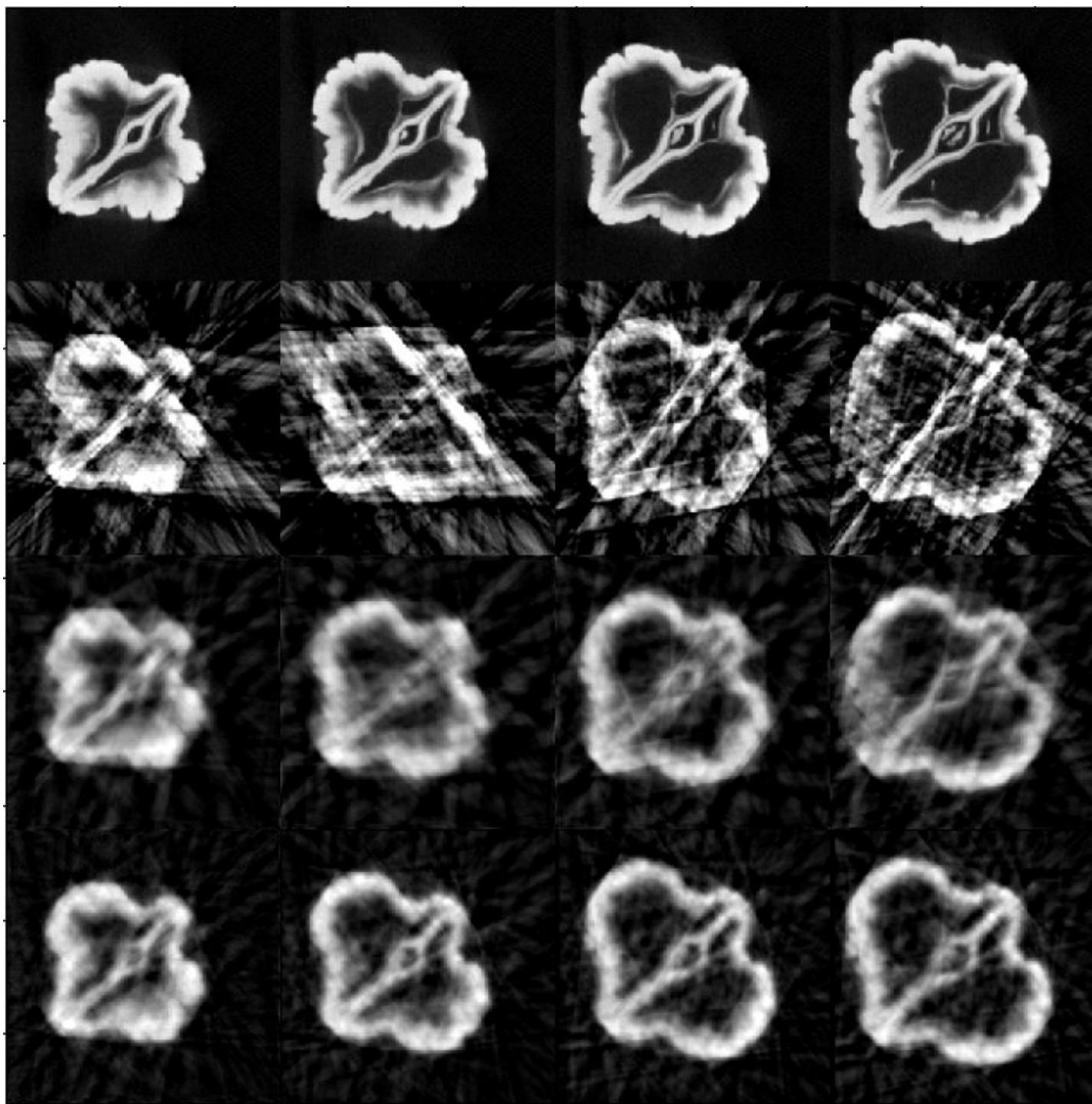
- In practice (say while using ISTA), the  $\mathbf{R}$  matrix is implemented in function handles that invoke the radon function in MATLAB.
- $\mathbf{R}^T$  is implemented using iradon.

# Tomography and Compressed sensing

- The **R** matrix is too large to store in memory.
- Likewise the **U** matrix is never allocated (it is too large to store in memory).
- Instead a function handle (`idct2` or `idwt2`) is invoked.
- See more about MATLAB function handles [here](#).
- Function handles for **R**, **U** offer computational speed and do not require too much memory.

# Tomography and compressed sensing

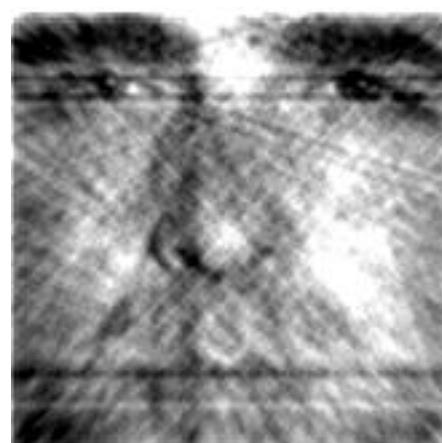
- The reconstruction results with CS are generally seen to be consistently superior to those by FBP in the angle-starved case.
- Next slide shows comparative results: original  $240 \times 240$  cross-sectional images of a walnut (top row), FBP reconstructions (2<sup>nd</sup> row), CS reconstructions (3<sup>rd</sup> row) and coupled CS reconstructions (4<sup>th</sup> row – explained later). All reconstructions with 20 angles.





Original image, size  $\sim 200 \times 200$

Reconstructions using FBP (left) and CS (right) using 15 angles (top row), 30 angles (middle row) and 40 angles (bottom row)



# Great results, but...

- The  $\mathbf{R}$  matrix as defined earlier on, however, is not known to obey the RIP or exhibit low coherence with standard bases.
- Hence the theoretical treatment of CS for tomography has not been fully established.

# Augmenting CS for tomography

- In multi-slice tomographic reconstruction, one can make use of additional redundancy in the data – for instance, the difference between two consecutive slices is sparse.
- This can be used to improve the tomographic reconstruction quality for the same number of measurements – if one chooses different angles for different slices.

# Augmenting CS for tomography

- The objective function (for two consecutive slices) is as follows:

$$\begin{aligned} E(\beta_1, \beta_2) &= \|y_1 - R_1 U \beta_1\|^2 + \|y_2 - R_2 U \beta_2\|^2 + \lambda \|\beta_1\|_1 + \lambda \|\beta_2 - \beta_1\|_1 \\ &= \|y_1 - R_1 U \beta_1\|^2 + \|y_2 - R_2 U (\beta_1 + \Delta \beta)\|^2 + \lambda \|\beta_1\|_1 + \lambda \|\Delta \beta\|_1 \\ &= \left\| \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} R_1 U & 0 \\ R_2 U & R_2 U \end{pmatrix} \begin{pmatrix} \beta_1 \\ \Delta \beta \end{pmatrix} \right\|^2 + \lambda \|\Delta \beta\|_1 \end{aligned}$$

Here **x1** and **x2** represent two consecutive slices of an organ (each slice is a 2D image), and **y1** and **y2** represent their tomographic projections expressed as 1D vectors.

# Augmenting CS for tomography

- See the walnut dataset 3-4 slides earlier for a sample reconstruction to see the advantage of this model.
- Note that  $R_1$  and  $R_2$  denote the Radon-based forward models for different angle sets (the number of angles in the two sets may or may not be equal).

# Tomography under unknown angles

# Problem statement

- The previous algorithms for tomographic reconstruction assumed that the angles of Radon projection were accurately known.
- In certain applications, this assumption is surprisingly invalid.
- This is called **tomography under unknown angles**.

# Problem statement: Applications

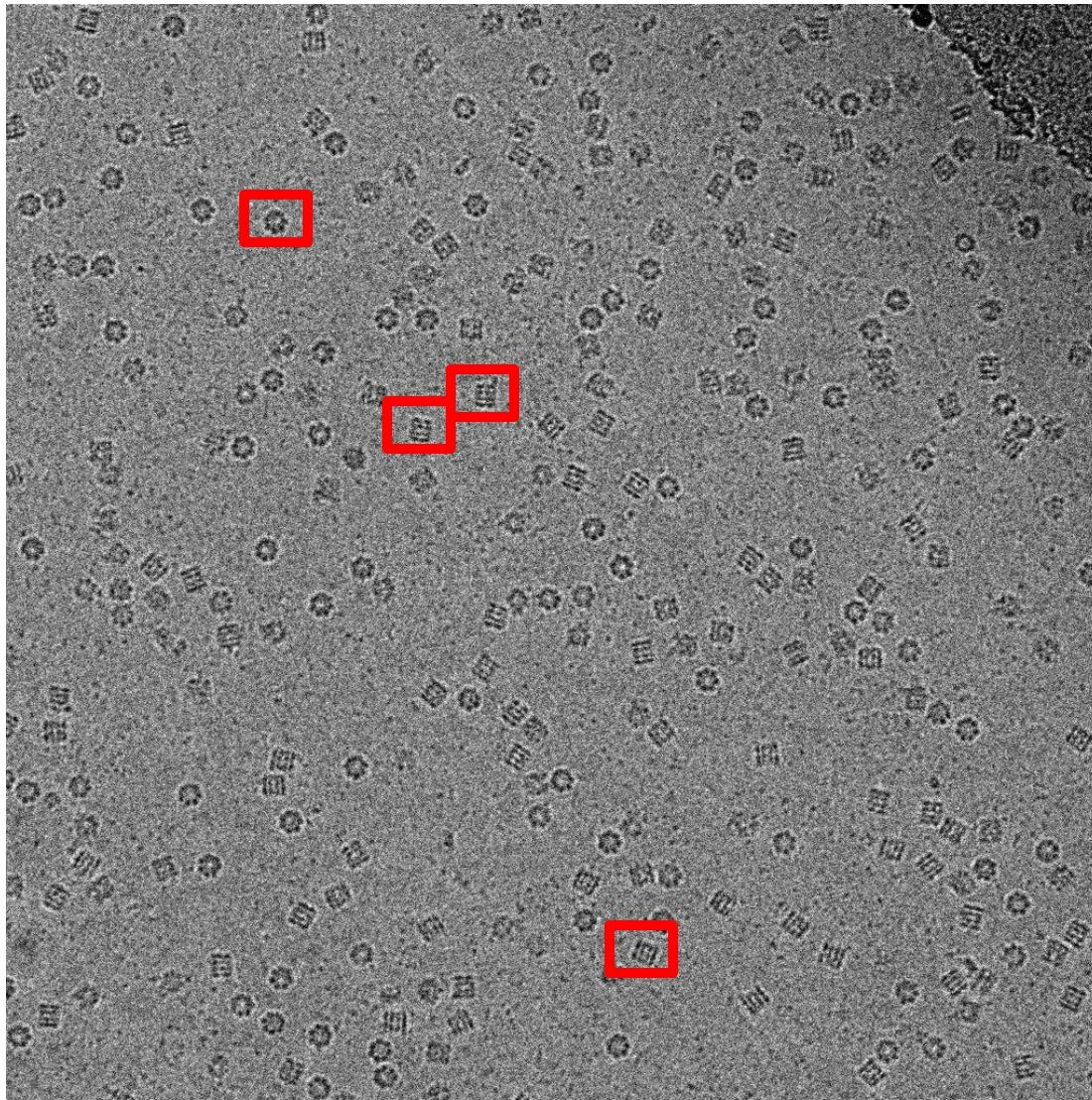
- Application 1: Patient motion during CT scanning
- Application 2: Moving insect tomography
- Application 3: Cryo-electron microscopy/tomography

# Problem statement: Applications

- Application 3: Cryo-electron microscopy
- In this, one collects multiple (nearly) identical samples of a structure (such as a virus/macromolecule) which we wish to image. In fact, we are interested in their complete 3D structure.
- Each slide contains thousands of macromolecules (i.e. samples) packed in a substrate such as ice.
- A tomographic projection is obtained by passing an electron-ray beam through all macromolecules, through some angle (usually vertically downward).

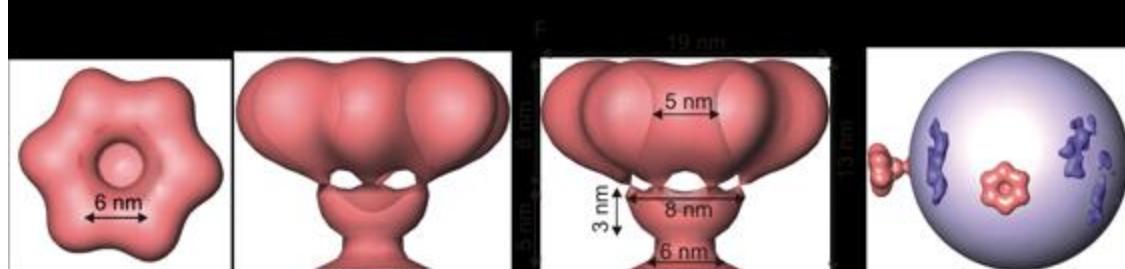
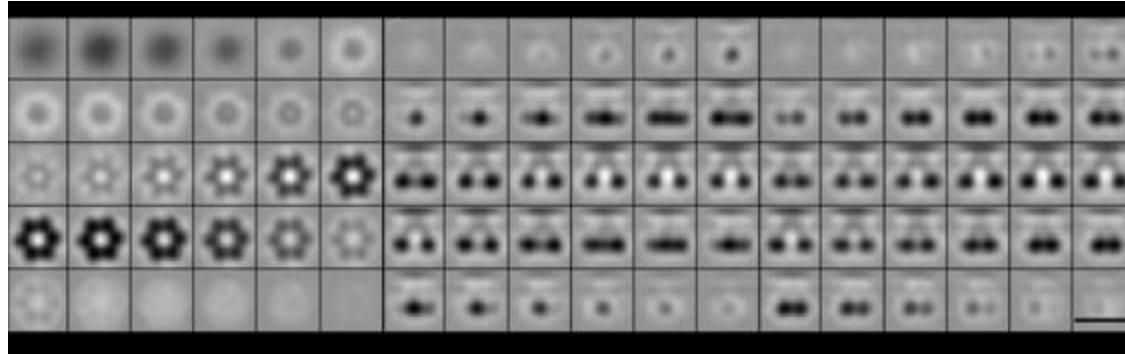
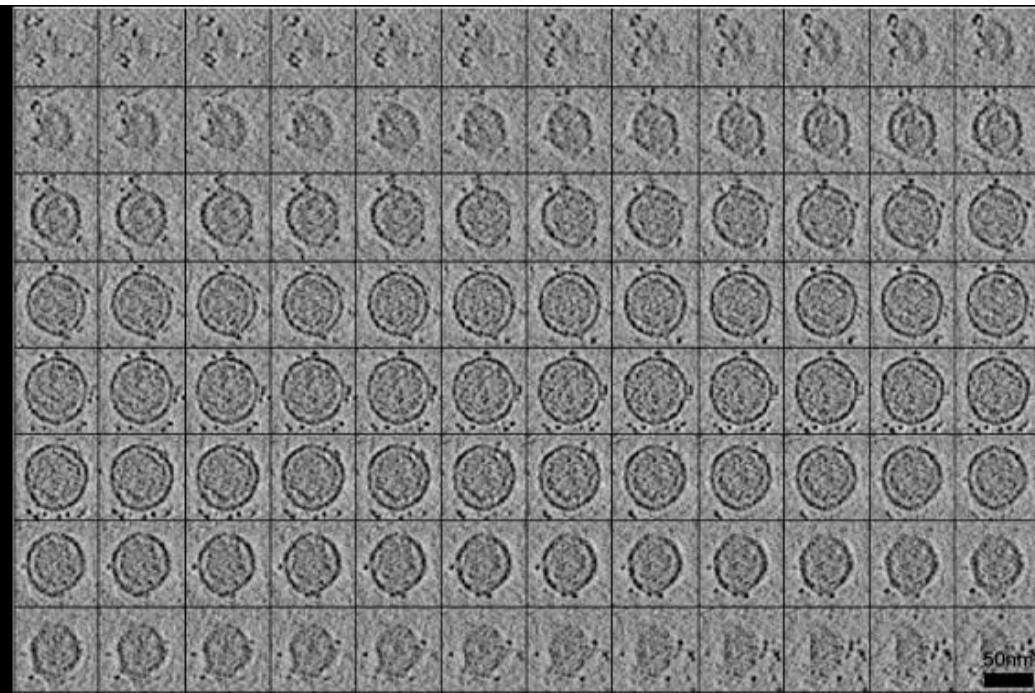
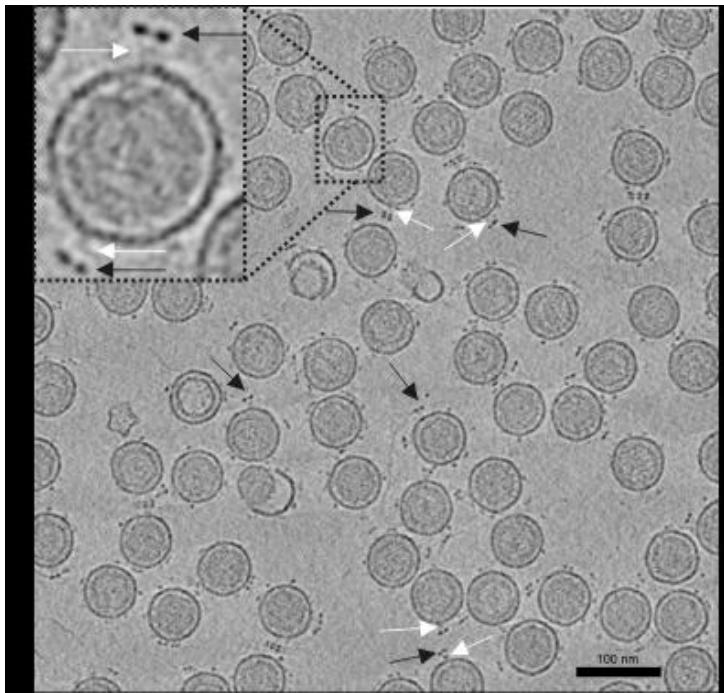
# Problem statement: Applications

- The electron beam usually destroys the sample, and hence another tomographic projection of a different sample (containing virus particles of the same type) is acquired.
- The problem is that each virus or macromolecule copy will be oriented randomly, and all the orientations are unknown!
- To make matters worse, the low power of the electron beam produces measurements that are extremely noisy.
- In such applications, however several hundred or even thousand projections (all under unknown angles) are acquired.

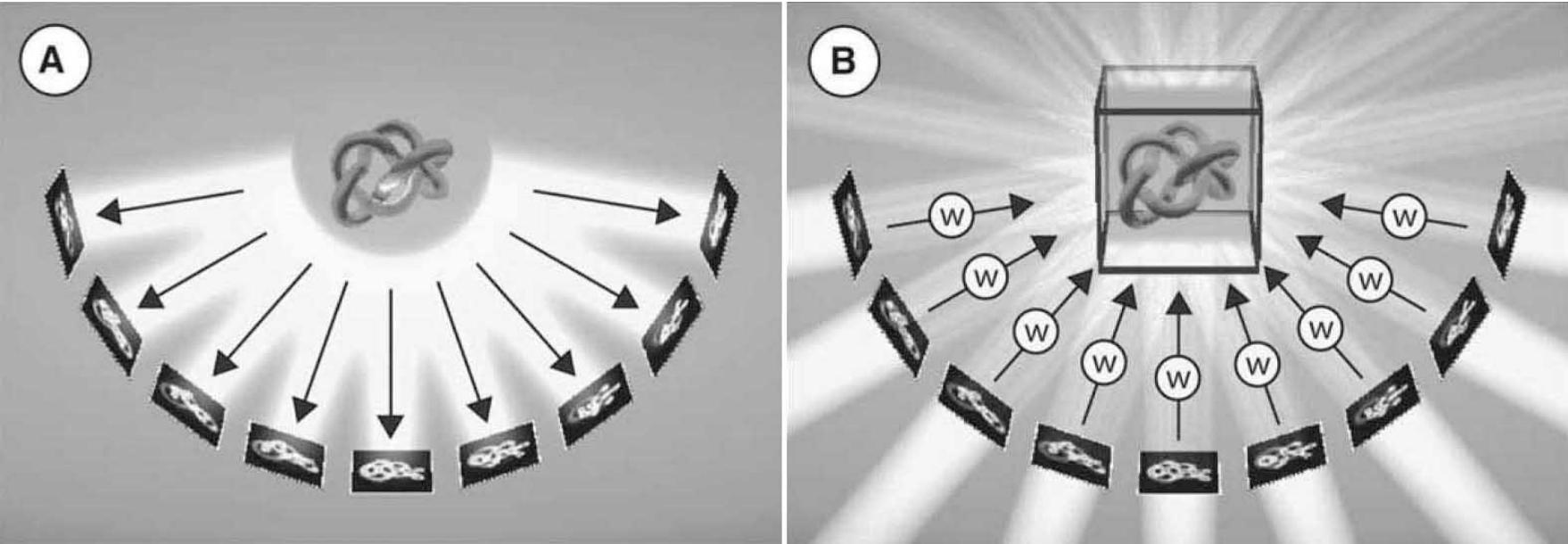


Such an image is called a **micrograph**. It contains tomographic projections of each macromolecule, against a background containing only ice. Each such projection image is called a **particle**.

[https://en.wikipedia.org/wiki/Cryogenic\\_electron\\_microscopy](https://en.wikipedia.org/wiki/Cryogenic_electron_microscopy)



<https://med.nyu.edu/skirball-lab/stokeslab/phi12.html>



<https://ki.se/en/research/core-facility-for-electron-tomography-o>

# Steps in Cryo-em microscopy

- Particle picking from noisy micrographs
- In some algorithms, similar particles are clustered and averaged to reduce noise
- Given the series of particle images, we then seek to solve jointly for the angles of projection and the underlying structure

# Cryo-electron microscopy wins the Nobel!

- Nobel in Chemistry in 2017

- More details here below:

[https://www.nobelprize.org/nobel\\_prizes/chemistry/laureates/2017/advanced-chemistryprize2017.pdf](https://www.nobelprize.org/nobel_prizes/chemistry/laureates/2017/advanced-chemistryprize2017.pdf)

# Two Different Approaches

- Moment-based approach
- Ordering-based approach
- Approach using dimensionality reduction  
(similar to ordering-based approach).

# Moment-based approach

# Moment-based approach

- We shall restrict ourselves to 2D images and 1D tomographic projections although the theory is extensible to 3D images (and their 2D projections)
- The **moment of order  $(p, q)$**  of an image  $f(x, y)$  is defined as follows:

$$M_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) x^p y^q dx dy,$$

# Moment-based approach

- The moment of order  $(p,q)$  where  $k=p+q$  of an image  $f(x,y)$  is defined as follows:

$$M_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) x^p y^q dx dy,$$

- Note that there can exist multiple pairs of  $(p,q)$  which sum up to  $k$ , and these are all called **order  $k$  image moments**.

# Moment-based approach

- The order  $n$  moment of a tomographic projection at angle  $\theta$  is defined as follows:

$$M_{\theta}^{(n)} = \int_{-\infty}^{\infty} P_{\theta}(s) s^n ds.$$

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

- Substituting the definition of  $P_{\theta}(s)$  into  $M_{\theta}(n)$ :

$$M_{\theta}^{(n)}$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - s) s^n dx dy ds.$$

$$M_{\theta}^{(n)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) (x \cos \theta + y \sin \theta)^n dx dy.$$

# Moment-based approach: image moments and projection moments

- Using the binomial theorem, we have:

$$(x \cos \theta + y \sin \theta)^n = \sum_{l=0}^n C_n^{n-l} (x \cos \theta)^{n-l} (y \sin \theta)^l.$$

- We will use this to derive a neat relationship between the tomographic projection moments and the image moments!
- See next slide.

# Moment-based approach: image moments and projection moments

$$M_{\theta}^{(n)} = \sum_{l=0}^n C_n^{n-l} (\cos \theta)^{n-l} (\sin \theta)^l$$

$$\times \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^{n-l} y^l f(x, y) dx dy,$$

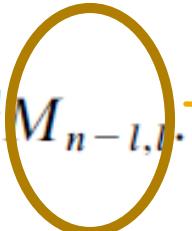
$$M_{\theta}^{(n)} = \sum_{l=0}^n C_n^{n-l} (\cos \theta)^{n-l} (\sin \theta)^l M_{n-l, l}.$$


Image moment of order  
( $n-l, l$ )

# Moment-based approach: image moments and projection moments

$$M_\theta^0 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy = M_{0,0}.$$

Substituting  $n = 0$ , with measurements at one angle.

$$M_{\theta_1}^{(1)} = \cos \theta_1 M_{1,0} + \sin \theta_1 M_{0,1},$$

Substituting  $n = 1$ , with measurements at two angles.

$$M_{\theta_2}^{(1)} = \cos \theta_2 M_{1,0} + \sin \theta_2 M_{0,1}.$$

$$\begin{pmatrix} M_{1,0} \\ M_{0,1} \end{pmatrix} = \begin{pmatrix} \cos \theta_1 & \sin \theta_1 \\ \cos \theta_2 & \sin \theta_2 \end{pmatrix}^{-1} \begin{pmatrix} M_{\theta_1}^{(1)} \\ M_{\theta_2}^{(1)} \end{pmatrix}.$$

# Moment-based approach: image moments and projection moments

Substituting  $n = k$ , with measurements at  $k+1$  different angles.

$$\begin{aligned} & \begin{pmatrix} M_{k,0} \\ M_{k-1,1} \\ \vdots \\ M_{0,k} \end{pmatrix} \\ &= \begin{pmatrix} \cos^k \theta_1 & C_k^{k-1} \cos^{k-1} \theta_1 \sin^1 \theta_1 & \cdots \sin^k \theta_1 \\ \cos^k \theta_2 & C_k^{k-1} \cos^{k-1} \theta_2 \sin^1 \theta_2 & \cdots \sin^k \theta_2 \\ \vdots & \vdots & \vdots \quad \vdots \\ \cos^k \theta_{k+1} & C_k^{k-1} \cos^{k-1} \theta_{k+1} \sin^1 \theta_{k+1} & \cdots \sin^k \theta_{k+1} \end{pmatrix}^{-1} \\ & \cdot \begin{pmatrix} M_{\theta_1}^{(k)} \\ M_{\theta_2}^{(k)} \\ \vdots \\ M_{\theta_{k+1}}^{(k)} \end{pmatrix}. \end{aligned} \tag{25}$$

# Moment-based approach: image moments and projection moments

$$IM^{(k)} = \begin{pmatrix} M_{k,0} \\ M_{k-1,1} \\ \vdots \\ M_{1,k-1} \\ M_{0,k} \end{pmatrix}, \quad PM^{(k)} = \begin{pmatrix} M_{\theta_1}^{(k)} \\ M_{\theta_2}^{(k)} \\ \vdots \\ M_{\theta_k}^{(k)} \\ M_{\theta_{k+1}}^{(k)} \end{pmatrix}.$$

These equations are called the Helgason-Ludwig consistency conditions (HLCC), and they give relations between image and projection moments.



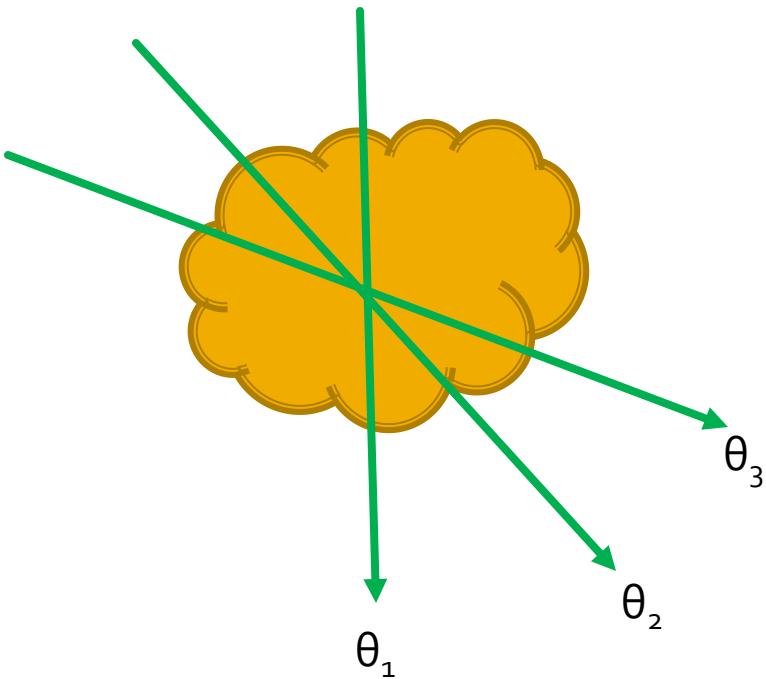
$$IM^{(k)} = A^{-1} PM^{(k)}.$$

$$A = \begin{pmatrix} \cos^k \theta_1 & C_k^{k-1} \cos^{k-1} \theta_1 \sin^1 \theta_1 & \dots & \sin^k \theta_1 \\ \cos^k \theta_2 & C_k^{k-1} \cos^{k-1} \theta_2 \sin^1 \theta_2 & \dots & \sin^k \theta_2 \\ \vdots & \vdots & \vdots & \vdots \\ \cos^k \theta_k & C_k^{k-1} \cos^{k-1} \theta_k \sin^1 \theta_k & \dots & \sin^k \theta_k \\ \cos^k \theta_{k+1} & C_k^{k-1} \cos^{k-1} \theta_{k+1} \sin^1 \theta_{k+1} & \dots & \sin^k \theta_{k+1} \end{pmatrix}.$$

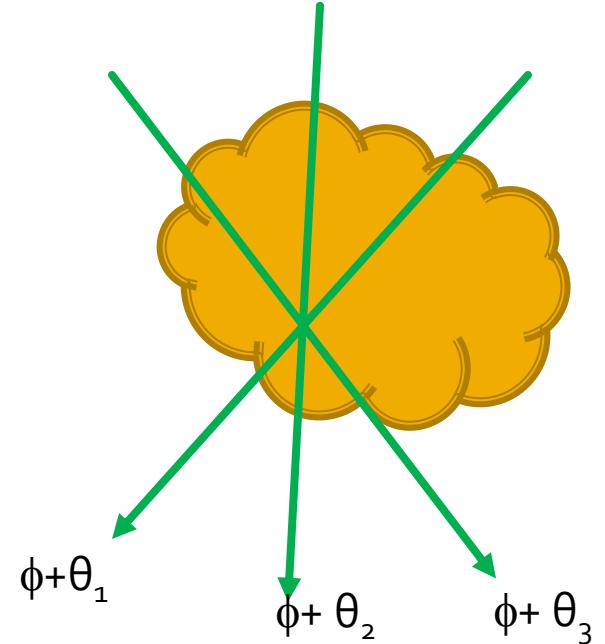
One can prove that the matrix **A** is invertible if and only if the projections are acquired at  $k+1$  distinct angles. In fact, unique  $k+1$  angles are necessary and sufficient for estimation of the image moments of order 0 through to order  $k$ .

# Tomography under unknown angles: rotation ambiguity

- In the tomography under unknown angles problem, we would know neither the image moments nor the angles of acquisition.
- In such a case, the underlying image can be obtained only up to an unknown rotation.
- To understand why, see the next slide.



In the first case you took 3 tomographic projections (denoted  $p_1, p_2, p_3$ ) of an object respectively at three angles  $\theta_1, \theta_2, \theta_3$



In the second case you took tomographic projections (denoted  $q_1, q_2, q_3$ ) of a version of the same object but rotated by  $\phi$  respectively at three angles  $\phi + \theta_1, \phi + \theta_2, \phi + \theta_3$

In both cases, the corresponding projections will be identical, ie  $p_1 = q_1, p_2 = q_2, p_3 = q_3$ ! The parameter  $\phi$  will always be indeterminate – but this is not a problem in most applications

# A surprising result

- Given tomographic projections of a 2D image in 8 or more distinct and unknown angles, the image moments of order 1 and 2, as well as the angles can be **uniquely** recovered – but up to the aforementioned rotation ambiguity.
- This result is true for almost any 2D image (i.e. barring a set of very rare “corner case” images).
- This result was proved in 2000 by Basu and Bresler at UIUC in a classic paper called “Uniqueness of tomography with unknown view angles”.
- In an accompanying paper called “Feasibility of tomography with unknown view angles”, they also proved that these estimates are **stable** under noise.
- The proof of the theorem and the discussion of the corner cases is outside the scope of our course.

# A surprising result

- In other words, systems of equations of the following form have a unique solution in the angles and the image moments, but modulo the rotation ambiguity:

$$PM_{\theta_i}^{(n)} = \sum_{l=0}^n C(n, l) \cos^{n-l} \theta_i \sin^l \theta_i M_{n-l, l} = A_{\theta_i}^{(n)} IM_n$$

Projection moments      Image moments      Column vector of image moments of order n

This is the  $n$ -th row of a matrix and it represents the linear combination coefficients for moments of order  $n$  and at angle  $\theta_i$ .

# Tomography under unknown angles: algorithm

- We can now build an algorithm for the aforementioned problem.
- Minimize the following objective function in an alternating fashion:

$$E(IM, \{\theta_i\}_{i=1}^Q) = \sum_{n=0}^N \sum_{i=1}^Q (PM_{\theta_i}^{(n)} - A_{\theta_i}^{(n)} IM_n)^2$$

Q = #projections  
N = max. order of image moments

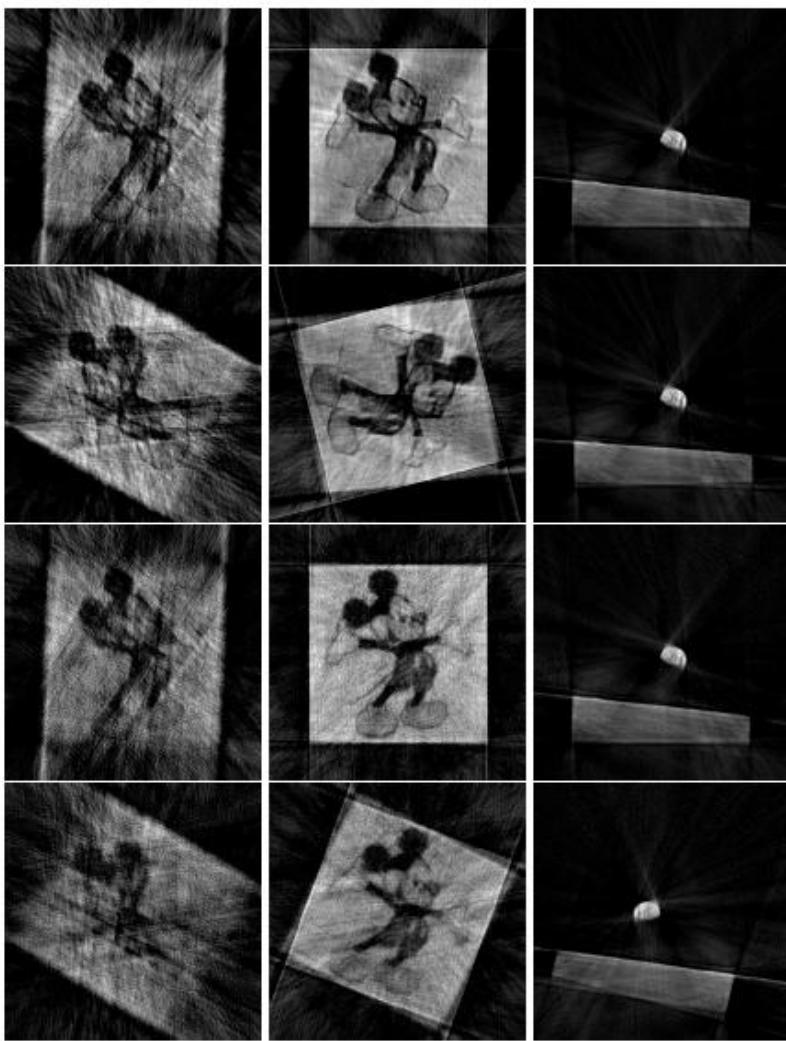
- Start with a random initial angle estimate
- (&) Compute the image moments by least squares inversion.

# Tomography under unknown angles: algorithm

- (\$) Next, do an independent brute force search over each angle  $\theta_i$ .
  - Repeat the earlier two steps (marked as & and \$) until convergence (i.e. Until the estimates of either thetas or the image moments do not change too much)
  - Perform a multi-start strategy for the best possible results – since this cost function is highly nonconvex.
- \* For every value of  $\theta_i$  sampled from 0 to 180, determine the image moments using that value, and hence determine the value of E.
- Choose the value of  $\theta_i$  corresponding to the least value of E.

# Tomography under unknown angles: algorithm

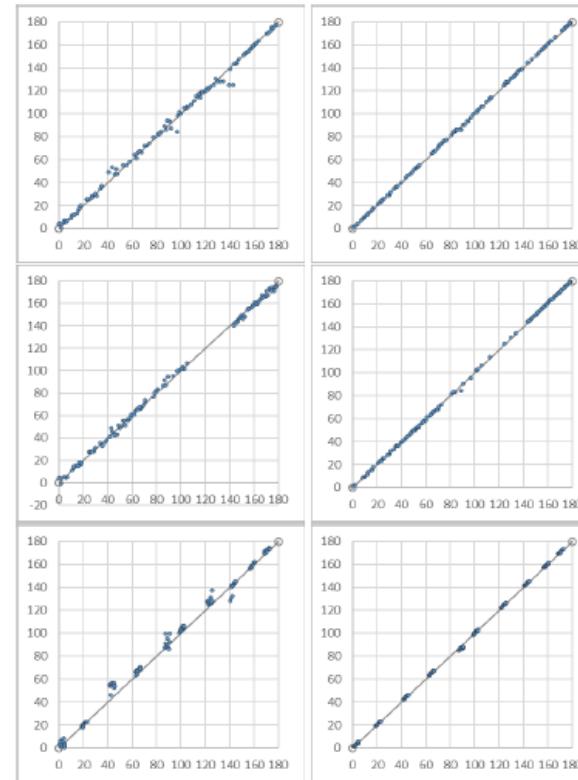
- Remember: these angles can be estimated only up to a global angular offset  $\phi$  which is indeterminate.
- Following the angle estimates, the underlying image can be reconstructed using FBP.



**Fig. 2.** FBP reconstruction using a non-uniform distribution of angles. Top to bottom: Using (a) 5% noise & actual angles, (b) 5% noise & estimated angles, (c) 10% noise & actual angles (d) 10% noise & estimated angles Reconstructions done with 30 angles (first column), 100 angles (second column), 100 angles (third column). Image canvas sizes have been expanded to allow for images to fit rotated reconstructions

**Table 1.** Statistics of errors in angle recovery, 5% noise

Error	Earthrise 30 angles	Earthrise 100 angles	Mickey 30 angles	Mickey 100 angles
$\leq 1^\circ$	13	94	20	66
$\leq 3^\circ$	29	99	29	96
$\leq 5^\circ$	30	100	29	100
$> 5^\circ$	0	0	1	0



**Fig. 3.** In each figure, X and Y axes show true and estimated angles respectively, under 10% noise. Left: image ‘mickey’; Right: image ‘earthrise’. Angles drawn from - Top row: a uniform distribution; Middle: a non-uniform distribution; Bottom: very peaky distribution. For visualization, the offset  $\phi$  in the estimated angles has been removed manually

Image source:  
Malhotra and  
Rajwade,  
“Tomographic  
reconstruction  
with unknown  
view angles  
exploiting  
moment-based  
relationships”

[https://www.cse.iitb.ac.in/~ajitvr/eeshan\\_icip2016.pdf](https://www.cse.iitb.ac.in/~ajitvr/eeshan_icip2016.pdf)

# Summary of moment-based method

- Advantage: simple, makes direct use of the consistency conditions and works even if the number of angles is small
- Disadvantage: moments are highly sensitive to noise

# Ordering-based approach

# Approach – the uniform distribution

- In this approach, the tomographic projections are “sorted” – i.e. arranged in order of increasing angles.
- The method relies on the availability of a large number of tomographic projections – all under unknown angles – but sampled independently from a uniform distribution on a unit circle.
- Given sufficiently many projections, the angles can be assumed to be equi-spaced.

# Approach – the uniform distribution

- The problem reduces to projection association – i.e. matching each projection to one of the angles, sampled evenly from the unit circle.
- If the angles are spaced sufficiently closely, and the variation of the projections w.r.t. angle, is smooth, then we can employ a nearest neighbour type of heuristic.
- This algorithm is summarized on the next slide.

# Algorithm

- Let us denote the list of available projections as  $L$ .
- Let  $q_i$  be the  $i$ -th tomographic projection and  $s_i$  be the  $i$ -th tomographic projection in the **sorted** list.
- WLOG, define the orientation of  $q_1$  to be  $o$  and set  $s_1 = q_1$ .
- Find the nearest neighbour of  $s_1$  in  $L$  as follows:

$$r = \arg \min_{q \in L} \min(\|q - s_1\|^2, \|q_{reverse} - s_1\|^2)$$

$$s_2 = q \text{ or } q_{reverse}$$

$$L = L - q \text{ (delete } q \text{ from } L)$$

- Repeat the earlier two steps with  $s_2$ .
- Continue these steps until  $L$  is empty.

# Output of the algorithm

- At the end of this algorithm, each projection will be associated with an angle at a grid point on the even sampling pattern from 0 to  $\pi$ .
- Remember: these angles can be estimated only up to a global angular offset  $\phi$  which is indeterminate.
- Following the angle estimates, the underlying image can be reconstructed using FBP.

# Comments on algorithm

- Note that essentially, each projection is being assigned an angle ranging from 0 to  $\pi$  in steps of  $\pi/Q$  where Q is the number of projections.
- The aforementioned algorithm is an approximate solution for the well-known **travelling salesman problem** in computer science.
- The role of  $q_{\text{reverse}}$  is to flip projections whose angles would have been in  $[\pi, 2\pi]$  because  $g(s, \theta + \pi) = g(-s, \theta)$  as per the definition of the Radon transform. If the angles were only from 0 to  $\pi$ , this step wouldn't be required.

# Comments on algorithm

- The accuracy of the angle estimate increases as the number of angles increases.
- Consider the  $k$ -th order statistic  $\theta_{(k)}$ , i.e. the  $k$ -th largest angle in the sorted sequence.
- The mean value of this order statistic is  $\pi k/Q$  where  $Q$  is the number of angles, assuming uniform distribution of the angles.
- The variance of this order statistic can be proved to be  $O(1/Q^2)$ .
- This method also relies on the fact that the angles were uniformly distributed, otherwise the angle grid points will need to be chosen differently – as per the order statistics of the other distribution.

# Unknown shifts

- Usually, the projection angles are unknown, but so are the projection shifts – due to uncontrolled translation of the object being imaged.
- However the unknown shifts can be solved for using a pre-processing step that assumes that the object is surrounded by vacuum.
- The tomographic projections can be normalized such that their centroid lies at the origin.
- The projection ordering step is performed later.

# Using dimensionality reduction

# Description of approach

- In principle, this method is similar to the ordering-based approach.
- However instead of doing pairwise searches, a dimensionality reduction step is applied.
- Each tomographic projection  $q_i$  (under unknown angles) can be regarded as a high-dimensional vector. There are  $Q$  tomographic projections in all.
- A dimensionality reduction method is applied to reduce their dimensionality to two (details later), i.e.  $q_i$  is mapped to  $y_i = (\phi_{1i}, \phi_{2i})$ .

# Description of approach

- Consequently the tomographic projection  $q_i$  will get mapped to the angle:

$$\vartheta_i = \tan^{-1} \left( \frac{\phi_{1i}}{\phi_{2i}} \right)$$

- These angles are then used as estimates of the true angles, prior to FBP-based reconstruction.

# Which method of dimensionality reduction?

- We need a method of dimensionality reduction which “preserves the neighbourhood structure” of the original high dimensional points.
- In other words, if  $\mathbf{q}_i$  and  $\mathbf{q}_j$  were close, their lower-dimensional projections  $\mathbf{y}_i$  and  $\mathbf{y}_j$  should also be close.
- To define “closeness” mathematically, consider the following proximity matrix  $\mathbf{W}$ :

$$W_{ij} = \exp\left(-\frac{\|\mathbf{q}_i - \mathbf{q}_j\|^2}{2\varepsilon}\right)$$

Higher values for  $W_{ij}$  means  $\mathbf{q}_i$  and  $\mathbf{q}_j$  are nearby in the Euclidean sense. Lower values means  $\mathbf{q}_i$  and  $\mathbf{q}_j$  are far apart.

# Which method of dimensionality reduction?

- Let the lower-dimensional projections be in 1D for now, i.e. each  $y_i$  is a scalar.
- To determine the projections, we now optimize the following:

$$\begin{aligned} E(\{y_i\}_{i=1}^Q) &= \sum_{i,j}^Q W_{ij} (y_i - y_j)^2 \\ &= \sum_{i,j} y_i^2 W_{ij} + \sum_{i,j} y_j^2 W_{ij} - 2 \sum_{i,j} y_i y_j W_{ij} = \sum_i y_i^2 D_{ii} + \sum_j y_j^2 D_{jj} - 2 \sum_{i,j} y_i y_j W_{ij} \\ &= 2 \left( \sum_i y_i^2 D_{ii} - \sum_{i,j} y_i y_j W_{ij} \right) = 2 \mathbf{y}^t (\mathbf{D} - \mathbf{W}) \mathbf{y} \\ &= 2 \mathbf{y}^t \mathbf{L} \mathbf{y} \end{aligned}$$

Define  $D_{ii} = \sum_j W_{ii}$ ,  $\mathbf{L} = \mathbf{D} - \mathbf{W}$

$\mathbf{D}$  is a diagonal weight matrix.  $\mathbf{y}$  is a Q-element vector containing the 1D projections of each tomographic projection.  $\mathbf{W}$  is a symmetric matrix.

# Which method of dimensionality reduction?

- So this becomes a constrained minimization problem:  
$$\therefore \mathbf{y}^* = \arg \min_{\mathbf{y}} \mathbf{y}' \mathbf{L} \mathbf{y} \text{ such that } \mathbf{y}' \mathbf{D} \mathbf{y} = 1$$
- The constraint is to prevent the trivial solution  $\mathbf{y} = \mathbf{0}$ .
- The trivial solution can also be prevented by imposing a unit norm constraint on  $\mathbf{y}$ , however the one used in the equation above assigns greater importance to points which are “popular”, i.e. for which the  $\mathbf{D}$  value is high.

# Which method of dimensionality reduction?

- Another trivial solution exists: one where  $\mathbf{y}$  contains all equal entries.
- Such a  $\mathbf{y}$  corresponds to an eigenvector with 0 eigenvalue.
- This solution is undesirable; so after solving the generalized eigenvalue problem, the eigenvector with eigenvalue 0 is just discarded.
- This can be interpreted as removal of translation in  $\mathbf{y}$ .  
Read pages 6 and 7 of [the journal paper on Laplacian eigenmaps](#) for more details on this aspect.

# Which method of dimensionality reduction?

- So this becomes a constrained minimization problem:

$$\therefore \mathbf{y}^* = \arg \min_{\mathbf{y}} \mathbf{y}^t \mathbf{L} \mathbf{y} \text{ such that } \mathbf{y}^t \mathbf{D} \mathbf{y} = 1$$

- The solution to this problem is obtained as follows:

$$\tilde{E}(\mathbf{y}) = \mathbf{y}^t \mathbf{L} \mathbf{y} - \lambda (\mathbf{y}^t \mathbf{D} \mathbf{y} - 1)$$

$$\frac{\partial \tilde{E}}{\partial \mathbf{y}} = \mathbf{0} \rightarrow \mathbf{L} \mathbf{y} = \lambda \mathbf{D} \mathbf{y}$$

- This is a generalized eigenvalue problem, and we are interested in the generalized eigenvector  $\mathbf{y}$  with the smallest generalized eigenvalue (why?).
- Note:  $\mathbf{L}$  must be positive semi-definite. If not,  $\mathbf{y}^t \mathbf{L} \mathbf{y} = \lambda \mathbf{y}^t \mathbf{y} < 0$ , since  $\lambda$  would be a negative eigenvalue. This is impossible as the cost function  $E(\{y_i\}) = \sum_{i,j} W_{ij} (y_i - y_j)^2$  must be non-negative.

# Which method of dimensionality reduction?

- Now consider that the lower-dimensional points were no longer just in 1D.
- In such a case, we seek to minimize:

$$\begin{aligned} E(\{\mathbf{y}_i\}_{i=1}^Q) &= \sum_{i,j}^Q W_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \\ &= \sum_{i,j} \mathbf{y}_i^t \mathbf{y}_i W_{ij} + \sum_{i,j} \mathbf{y}_j^t \mathbf{y}_j W_{ij} - 2 \sum_{i,j} \mathbf{y}_i^t \mathbf{y}_j W_{ij} = \sum_i \mathbf{y}_i^t \mathbf{y}_i D_{ii} + \sum_j \mathbf{y}_j^t \mathbf{y}_j D_{jj} - 2 \sum_{i,j} \mathbf{y}_i^t \mathbf{y}_j W_{ij} \\ &= 2 \sum_i \mathbf{y}_i^t \mathbf{y}_i D_{ii} - 2 \sum_{i,j} \mathbf{y}_i^t \mathbf{y}_j W_{ij} \\ &= 2 \text{trace}(\mathbf{Y}^T \mathbf{LY}) \text{ where } \mathbf{Y}^T = (\mathbf{y}_1 | \mathbf{y}_2 | \dots | \mathbf{y}_Q), \text{ size } d \times Q \end{aligned}$$

Normalization constraint:

$$\mathbf{Y}^T \mathbf{D} \mathbf{Y} = \mathbf{I}$$

# Which method of dimensionality reduction?

- This reduces to a generalized eigenvalue problem, i.e. to finding generalized eigenvectors of the following form, with the lowest eigenvalues:

$$\mathbf{Ly} = \lambda \mathbf{Dy}$$

- This technique is called “Laplacian Eigenmaps” since the matrix  $\mathbf{L}$  is called the (graph) Laplacian matrix, which is commonly used in spectral graph theory.
- This technique is due to Belkin and Niyogi – their NIPS 2003 paper “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation”.

<http://www2.imm.dtu.dk/projects/manifold/Papers/Laplacian.pdf>

# Toy example

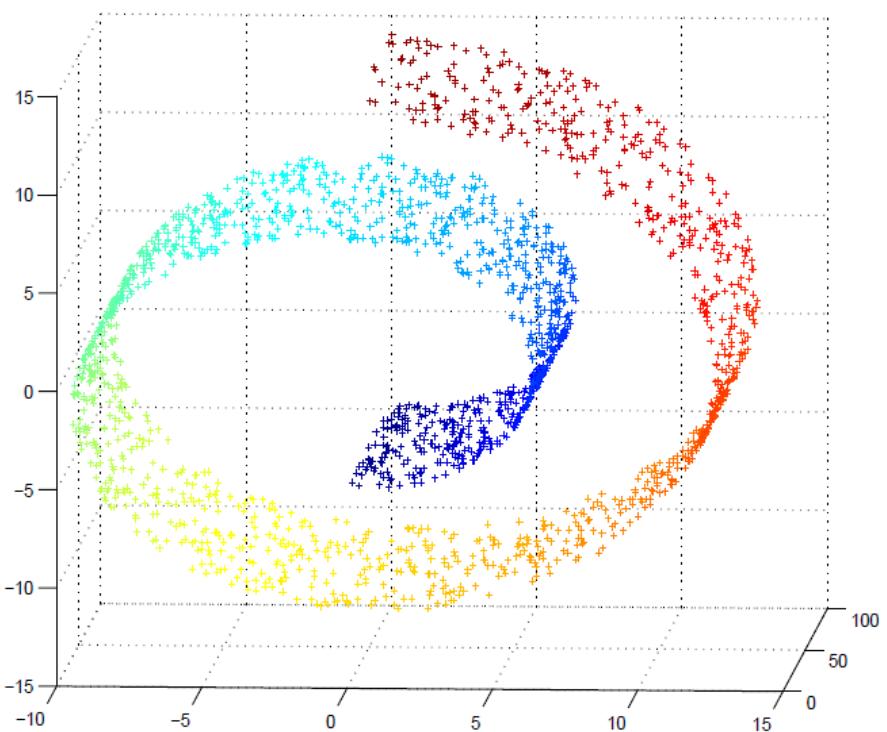


Figure 1: 2000 random data points on the “swiss roll”.

Image source:

<https://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class17.pdf>

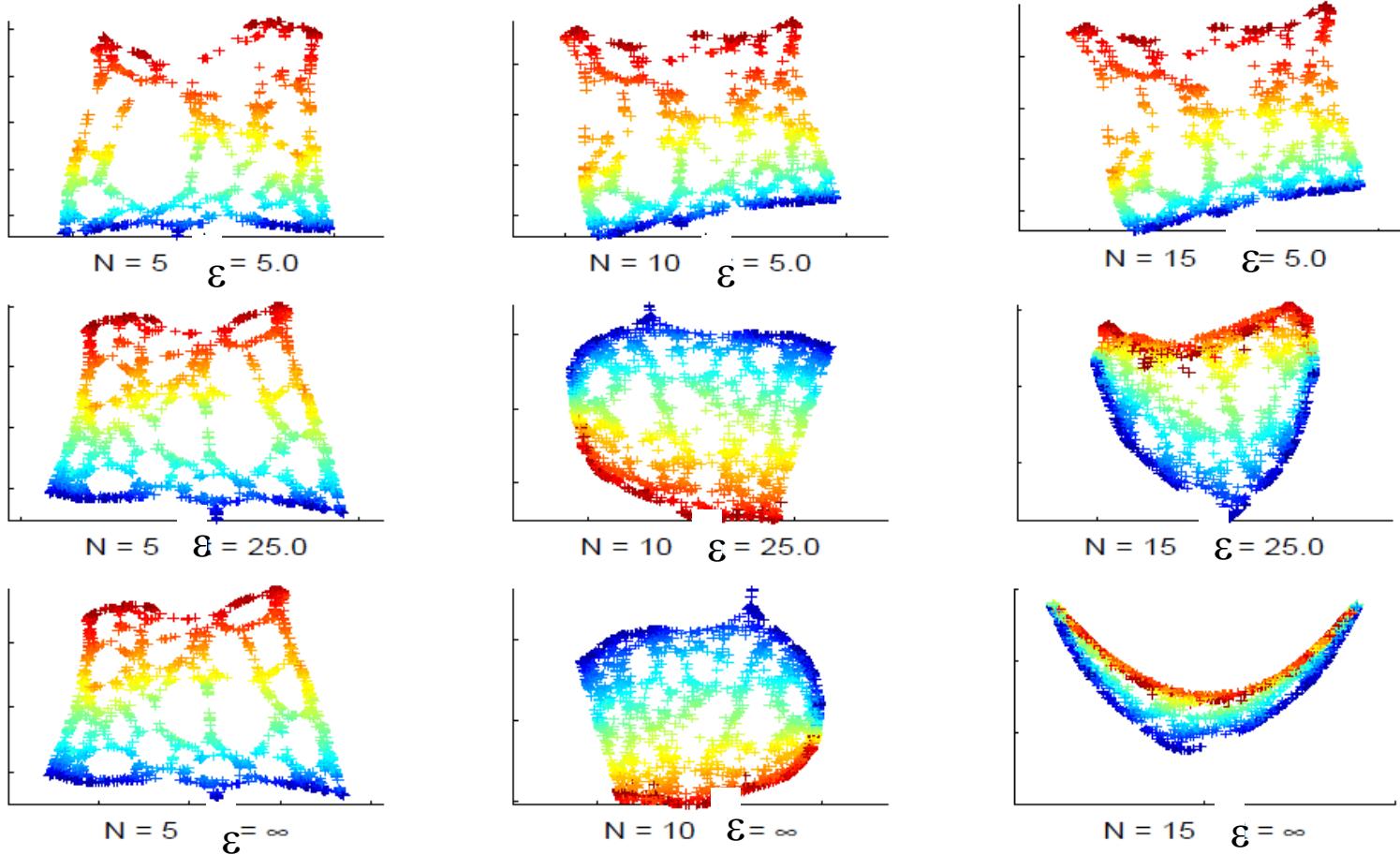


Figure 2: Two-dimensional representations of the “swiss roll” data, for different values of the number of nearest neighbors  $N$  and the heat kernel parameter  $t$ .  $t = \infty$  corresponds to the discrete weights.

On this slide,  $N$  refers to the number of nearest neighbors per point (the other distances are set to infinity). The parameter  $\varepsilon$  for the Gaussian kernel needs to be selected carefully, especially if  $N$  is high.

Image source:  
<https://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class17.pdf>

# Laplacian Eigenmaps in tomography under unknown angles

- For each of the  $Q$  tomographic projections, create a reversed copy (to take care of angles between  $\pi$  and  $2\pi$ ).
- Apply Laplacian Eigenmaps to reduce the dimensionality of these tomographic projections to 2, i.e. tomographic projection  $q_i$  is mapped to  $y_i = (\phi_{1i}, \phi_{2i})$ , i.e. angle  $\vartheta_i = \text{atan}(\phi_{1i}/\phi_{2i})$ .
- Sort the tomographic projections as per angles  $\vartheta_i$
- Reconstruct the image based on angle estimates of the form  $\pi i / Q$ .

# Results

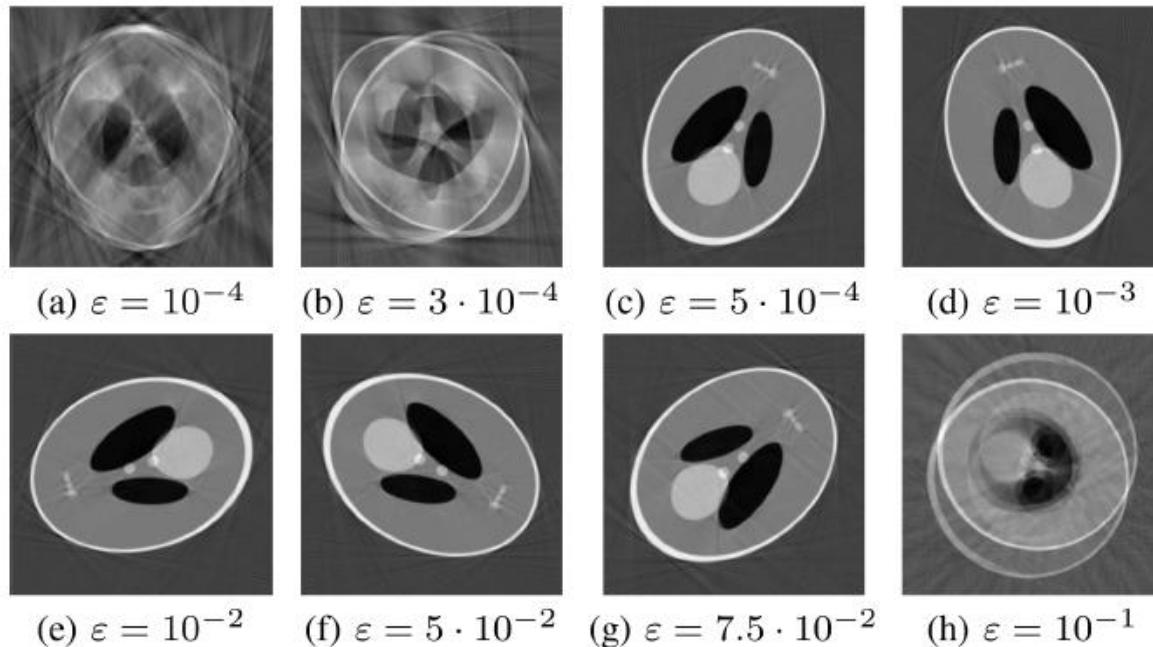
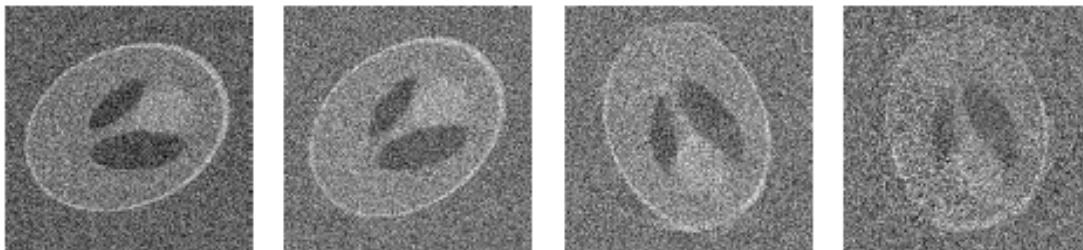


Fig. 5. Reconstructing the Shepp–Logan phantom from its random projections for different values of  $\varepsilon$  [increasing from (a) to (h)]. All reconstructions use  $N = 256$ ,  $mN = 4096$ , and  $n = 500$  pixels per projection. High-quality reconstructions are obtained for a wide range of  $\varepsilon$  values.

[Image source: Coifman et al, Graph Laplacian Tomography from Unknown Random Projections](#)

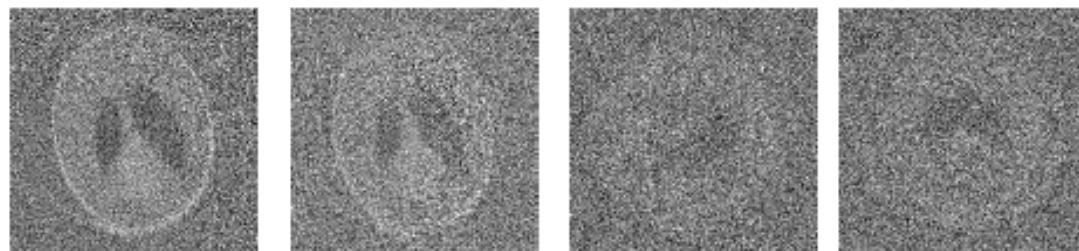


(a) SNR=2dB

(b) SNR=1dB

(c) SNR=0dB

(d) SNR=-1dB

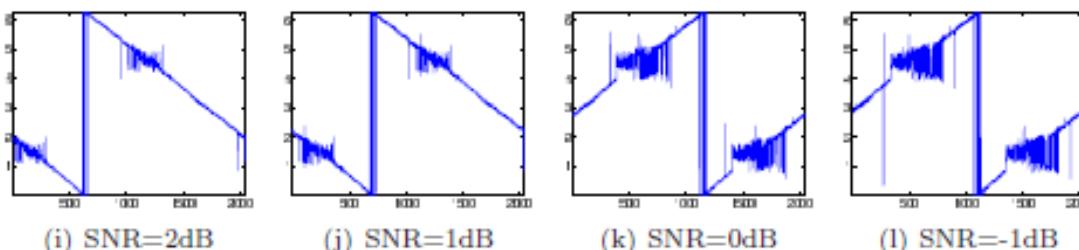


(e) SNR=-2dB

(f) SNR=-3dB

(g) SNR=-4dB

(h) SNR=-5dB

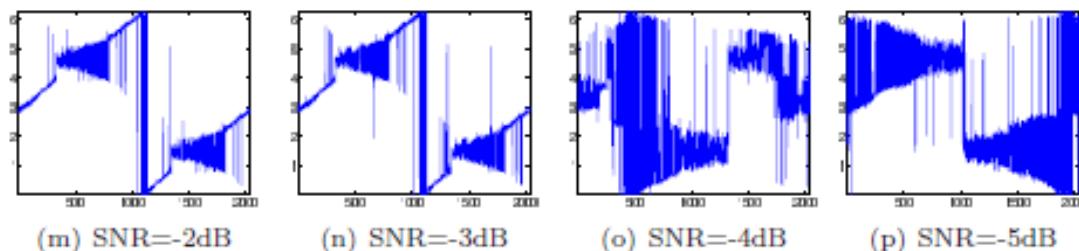


(i) SNR=2dB

(j) SNR=1dB

(k) SNR=0dB

(l) SNR=-1dB



(m) SNR=-2dB

(n) SNR=-3dB

(o) SNR=-4dB

(p) SNR=-5dB

Image source: Singer and Wu,  
Two-dimensional tomography  
from noisy projections taken at  
unknown random directions,

FIG. 7.2. Top: reconstruction from noisy projections at unknown directions using the algorithm described in Section 6 (excluding step 6) at different levels of noise with fixed  $\alpha = 6$  and  $\beta = 0.5$ . Bottom: estimated beaming directions (y-axis) against their correct ordering (x-axis).

# Taking care of noise

- For all three methods (moment-based, ordering-based, as well as based on dimensionality reduction), the tomographic projections presented to the algorithm may be very noisy (in actual applications).
- They need to be denoised, for optimal performance of the algorithm.

# Taking care of noise

- In these applications, the projections are extremely noisy.
- Prior to the ordering algorithm, a denoising step is required to prevent erroneous ordering.
- The denoising step can be performed using Principal Components Analysis (PCA).

# Taking care of noise

- Assuming a Gaussian noise model, the noise variance can be semi-automatically estimated from the regions in the projections which are completely blank (modulo noise).
- An algorithm similar to the patch-based PCA denoising (seen in CS 663 last semester) can be employed.
- Given a small patch in a projection vector, we can find small patches similar to it at spatially distant locations within the same projection vector or in projection vectors acquired from other angles.

# Use of PCA for denoising

- This “**non-local**” principle can be combined with PCA for denoising.
- Consider a set of clean projections  $\{I_r\}$ ,  $r=1$  to  $P$ , corrupted by additive Gaussian noise of mean zero and standard deviation  $\sigma$ , to give noisy projection  $J_r$  as follows:  
$$J_r = I_r + N_r, \quad N_r \sim \text{Gaussian distribution of mean } 0 \text{ and standard deviation } \sigma.$$
- Given each  $J_r$ , we want to estimate  $I_r$ , i.e. we want to denoise  $J_r$ .

# Use of PCA for denoising

- Consider a small  $p \times 1$  patch – denoted  $\mathbf{q}_{\text{ref}}$  - in some  $J_s$ .
- **Step 1:** We will collect together some  $L$  patches  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_L\}$  from  $\{J_r\}$ ,  $r = 1$  to  $R$ , that are structurally *similar* to  $\mathbf{q}_{\text{ref}}$  – pick the  $L$  nearest neighbors of  $\mathbf{q}_{\text{ref}}$ .
- Note: even if  $J_s$  is noisy, there is enough information in it to judge similarity if we assume  $\sigma \ll$  average intensity of the true projection  $I_s$ .
- **Step 2:** Assemble these  $L$  patches into a matrix of size  $p \times L$ . Let us denote this matrix as  $\mathbf{X}_{\text{ref}}$ .

# Use of PCA for denoising

- **Step 3:** Find the eigenvectors of  $\mathbf{X}_{\text{ref}} \mathbf{X}_{\text{ref}}^T$  to produce an eigenvector matrix  $\mathbf{V}$ .
- **Step 4:** Project each of the (noisy) patches  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_L\}$  onto  $\mathbf{V}$  and compute their eigen-coefficient vectors denoted as  $\{\alpha_1, \alpha_2, \dots, \alpha_L\}$  where  $\alpha_i = \mathbf{V}^T \mathbf{q}_i$ .
- **Step 5:** Now, we need to manipulate the eigencoefficients of  $\mathbf{q}_{\text{ref}}$  in order to denoise it.

# Use of PCA for denoising

- Step 5 (continued): We will follow a Wiener filter type of update:

$$\beta_{\text{ref}}(l) = \frac{1}{1 + \frac{\sigma^2}{\bar{\alpha}^2(l)}} \alpha_{\text{ref}}(l), 0 \leq l \leq p - 1$$

Noise variance (assumed known or can be estimated)

Estimate of coefficient squared of true signal

Note:  $\alpha_{\text{ref}}$  is a vector of eigencoefficients of the reference (noisy) patch and contains  $p$  elements, of which the  $l$ -th element is  $\alpha_{\text{ref}}(l)$ .  $\beta_{\text{ref}}$  is the vector of eigencoefficients of the filtered patch.

$$\bar{\alpha}^2(l) = \max(0, \frac{1}{L} \sum_{i=1}^L \alpha_i^2(l) - \sigma^2)$$

Why this formula? We will see later.

- Step 6: Reconstruct the reference patch as follows:  $q_{\text{ref}}^{\text{denoised}} = V\beta_{\text{ref}}$

# Use of PCA for denoising

- Repeat steps 1 to 6 for all  $p \times 1$  patches from noisy projection vector  $J_s$  (in a sliding window fashion).
- Since we take overlapping patches, any given pixel will be covered by multiple patches (as many as  $p$  different patches).
- Reconstruct the final projection by averaging the output values that appear at any pixel.

# Comments: Use of PCA for denoising

- Note that a **separate** eigenspace is created for each reference patch. The eigenspace is always created from patches that are similar to the reference patch.
- Such a technique is often called as **spatially varying PCA** or **non-local PCA**.

# Patch similarity: Use of PCA for denoising

- To compute  $L$  nearest neighbors of  $\mathbf{q}_{\text{ref}}$ , restrict your search to a window around  $\mathbf{q}_{\text{ref}}$ .
- For every patch  $\mathbf{s}$  within the window, compute the sum of squared differences with  $\mathbf{q}_{\text{ref}}$ , i.e. compute:  $\sum_{i=1}^p (q_{\text{ref}}(i) - s(i))^2$  .
- Pick  $L$  patches with the least distance.

# Use of PCA in denoising: why Wiener-like update?

$$\beta_i(l) = k(l)\alpha_i(l), 0 \leq l \leq p-1$$

$$k^* = \arg \min_{k(l)} E(\beta_i(l) - k(l)\alpha_i(l))^2$$

$$= \arg \min_{k(l)} E((\beta_i(l))^2 + k(l)^2 \alpha_i^2(l) - 2k(l)\alpha_i(l)\beta_i(l))$$

Eigen-coefficients of the “true patch”. We are looking for a linear update which motivates this equation.

$$\text{Consider: } \mathbf{q}_i^{\text{noisy}} = \mathbf{q}_i^{\text{true}} + \mathbf{n}_i \rightarrow V^T \mathbf{q}_i^{\text{noisy}} = V^T (\mathbf{q}_i^{\text{true}} + \mathbf{n}_i)$$

$$\therefore \alpha_i(l) = \beta_i(l) + \gamma_i(l)$$

$\mathbf{n}_i$  represents a vector of pure noise values which degrades the true patch to give the noisy patch. Its projection onto the eigenspace gives vector  $\mathbf{Y}_i$ .

We will drop the index  $l$  and subscript  $i$  for better readability :

$$\therefore k^* = \arg \min_k E(\beta^2 + k^2 \alpha^2 - 2k\beta\alpha)$$

$$= \arg \min_k E(\beta^2 + k^2(\beta + \gamma)^2 - 2k\beta(\beta + \gamma))$$

$$= \arg \min_k E(\beta^2) + k^2 E(\beta^2 + \gamma^2) - 2kE(\beta^2), \text{ as } E(\beta\gamma) = E(\beta)E(\gamma) = 0$$

$$\text{since } E(\gamma) = E(V^T n) = V^T E(n) = 0$$



As the noise is zero mean

As the image and the noise are independent

# Use of PCA in denoising: why Wiener-like update?

Setting derivative w.r.t.  $k$  to 0, we get

$$kE(\beta^2 + \gamma^2) = E(\beta^2) \rightarrow k = \frac{E(\beta^2)}{E(\beta^2 + \gamma^2)} = \frac{E(\beta^2)}{E(\beta^2) + \sigma^2}$$

How should we estimate  $E(\beta^2)$ ?

Recall:

$$\alpha_i(l) = \beta_i(l) + \gamma_i(l) \approx \beta(l) + \gamma_i(l)$$

$$\therefore E(\alpha_i^2(l)) = E(\beta^2(l)) + E(\gamma_i^2(l))$$

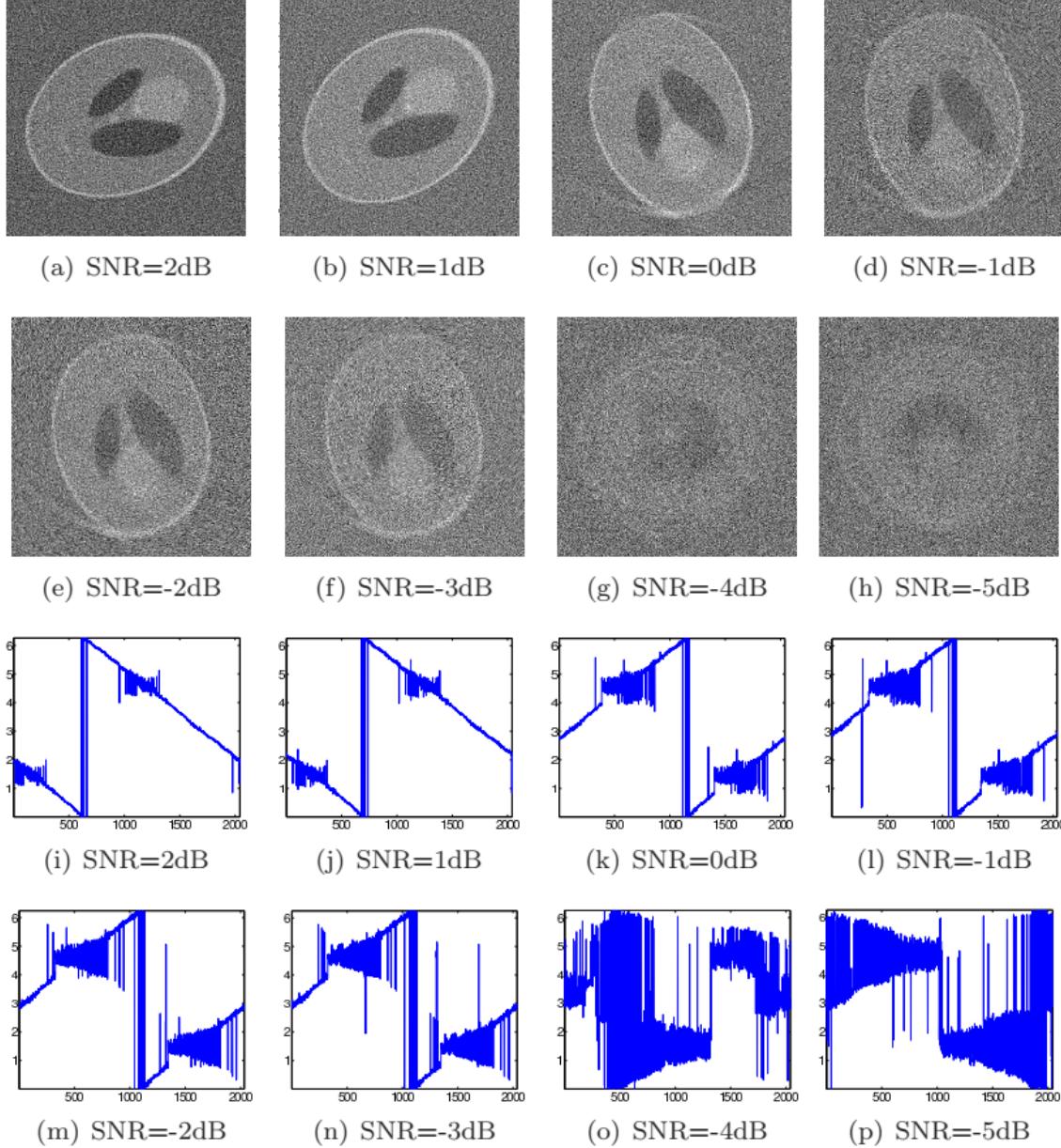
$$\therefore E(\beta^2(l)) = E(\alpha_i^2(l)) - \sigma^2$$

$$= \frac{1}{L} \sum_{i=1}^L \alpha_i^2(l) - \sigma^2$$

This may be a negative value, so we set it to be

$$E(\beta^2(l)) = \max(0, \frac{1}{L} \sum_{i=1}^L \alpha_i^2(l) - \sigma^2)$$

Since we are dealing with  $L$  similar patches, we can assume (approximately) that the  $l$ -th eigen-coefficient of each of those  $L$  patches are very similar.



Reconstruction from noisy projections: notice deterioration at very low SNR and much better performance at higher SNR. Performance is in terms of reconstructed image quality as well as angle estimates. Image taken from [here](#).

FIG. 7.2. Top: reconstruction from noisy projections at unknown directions using the algorithm described in Section 6 (excluding step 6) at different levels of noise with fixed  $\alpha = 6$  and  $\beta = 0.5$ . Bottom: estimated beaming directions (y-axis) against their correct ordering (x-axis).

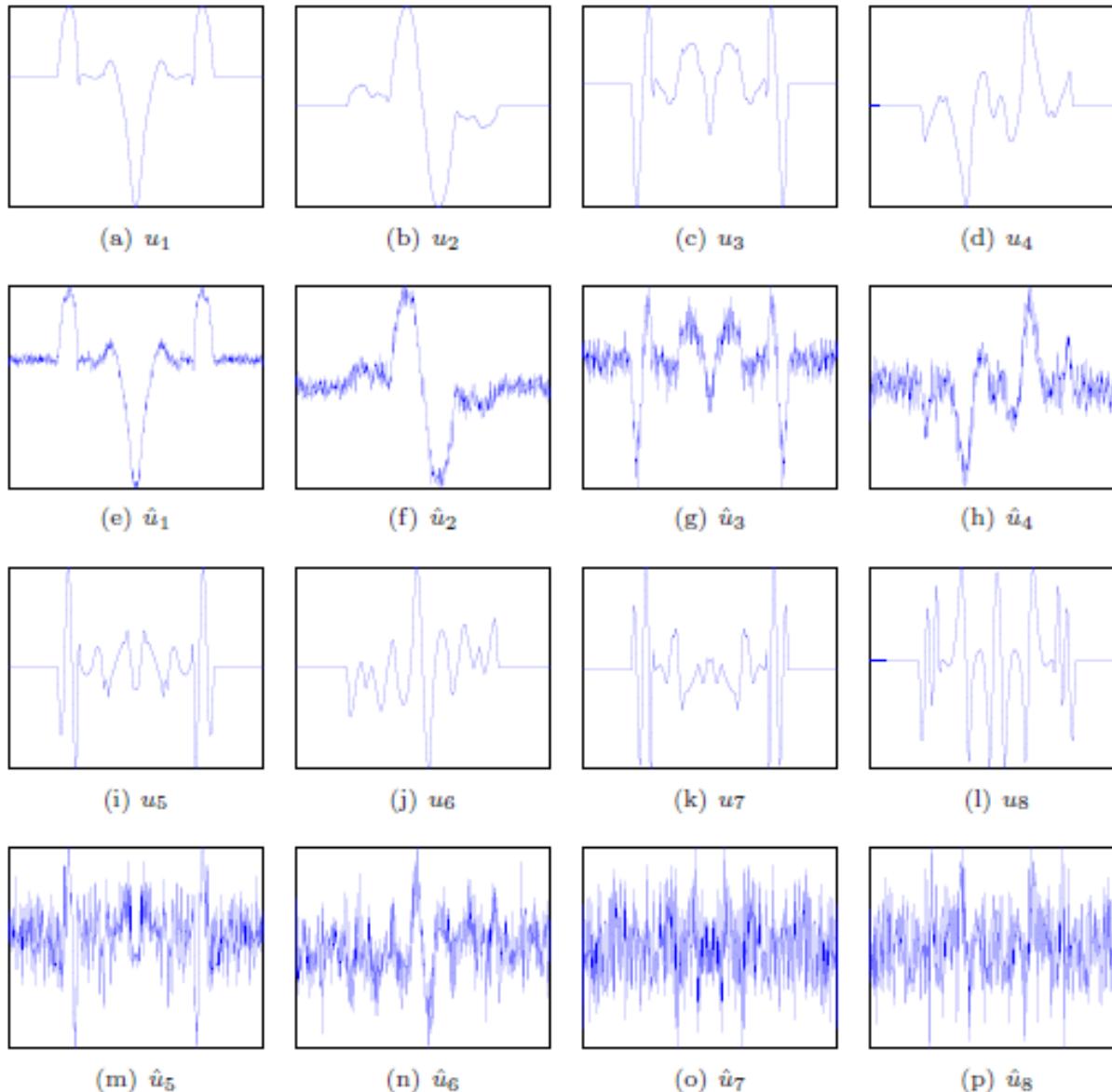


FIG. 7.6. The first 8 principal components for clean projections and the first 8 principal components for noisy projections with SNR =  $-5\text{dB}$ . Notice that the principal components are determined up to an arbitrary sign, and we choose the signs so that corresponding pairs of components are positively correlated.

Image source:  
Singer and Wu, Two-  
dimensional  
tomography from  
noisy projections  
taken at unknown  
random directions

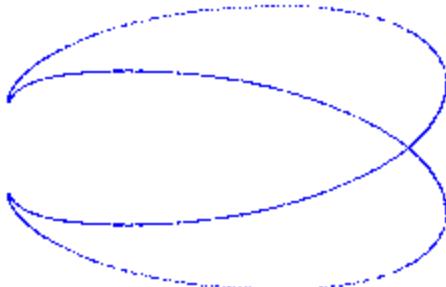


FIG. 7.7. *Projecting the curve  $C$  of clean projections of the Shepp-Logan phantom onto the linear subspace spanned by the top two principal components. The projected curve has a non-trivial self-intersection, implying the insufficiency of just two principal components.*

Please see this figure in the context of the previous one. It presents why PCA with 2 components is insufficient and in some sense motivates the use of Laplacian eigenmaps (after denoising using PCA).

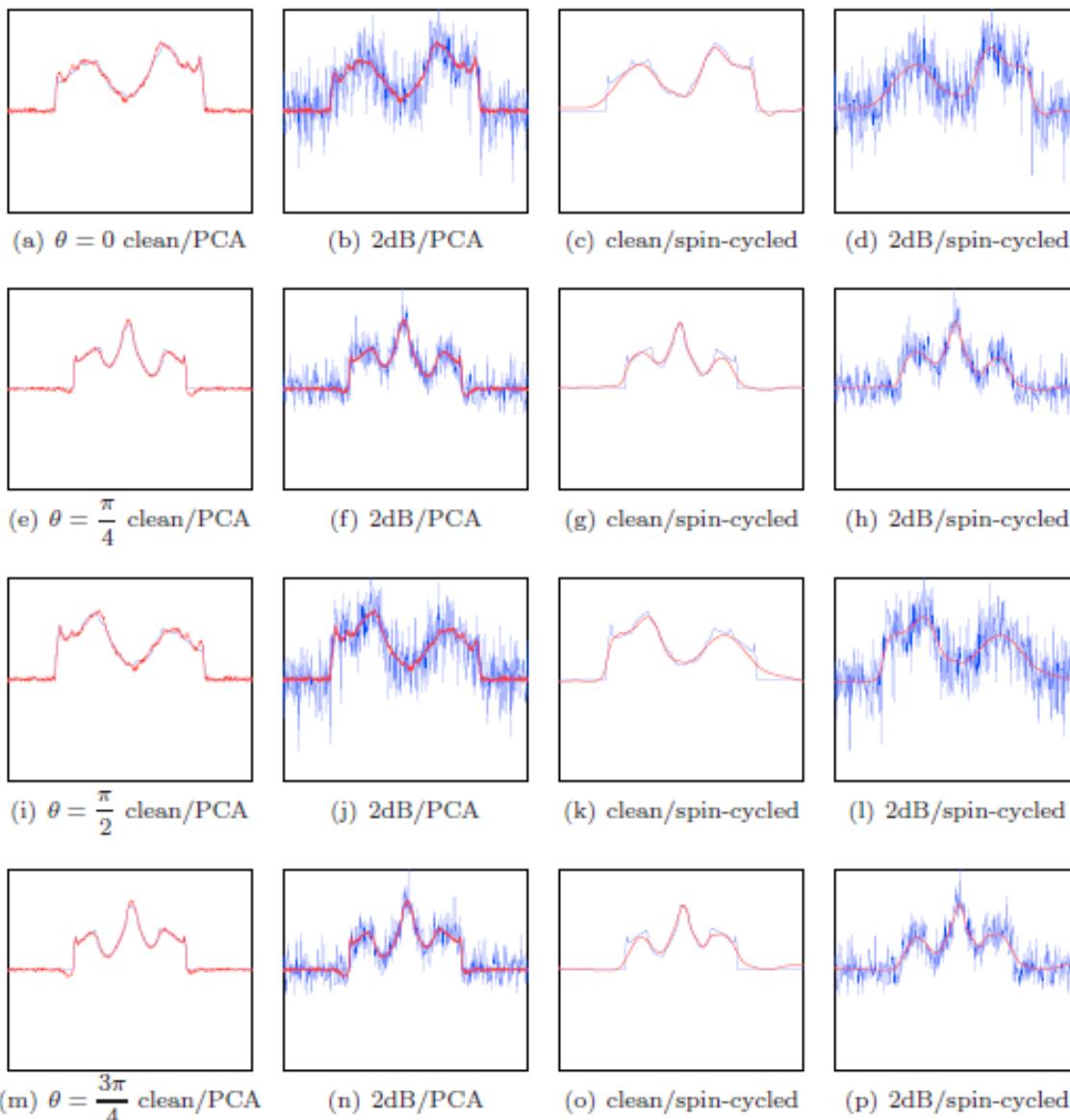


Image source: Singer and Wu,  
Two-dimensional tomography  
from noisy projections taken at  
unknown random directions

FIG. 7.9. Comparison between the combined PCA-Wiener filtering and wavelet denoising for four different noisy projections with SNR = 2dB taken at  $\theta = 0, \frac{\pi}{4}, \frac{\pi}{2}$ , and  $\frac{3\pi}{4}$ . (a) clean projection (Blue) and the PCA denoising of the noisy projection (Red); (b) noisy projection (Blue) its filtered version using PCA (Red); (c) clean projection (Blue) and the wavelet denoising of the noisy projection (Red); (d) noisy projection (Blue) and its filtered version using wavelets (Red). The number of principal components used by the Wiener filter is 12. RMSE: 0.464 for the PCA-Wiener scheme and 0.612 for wavelets.

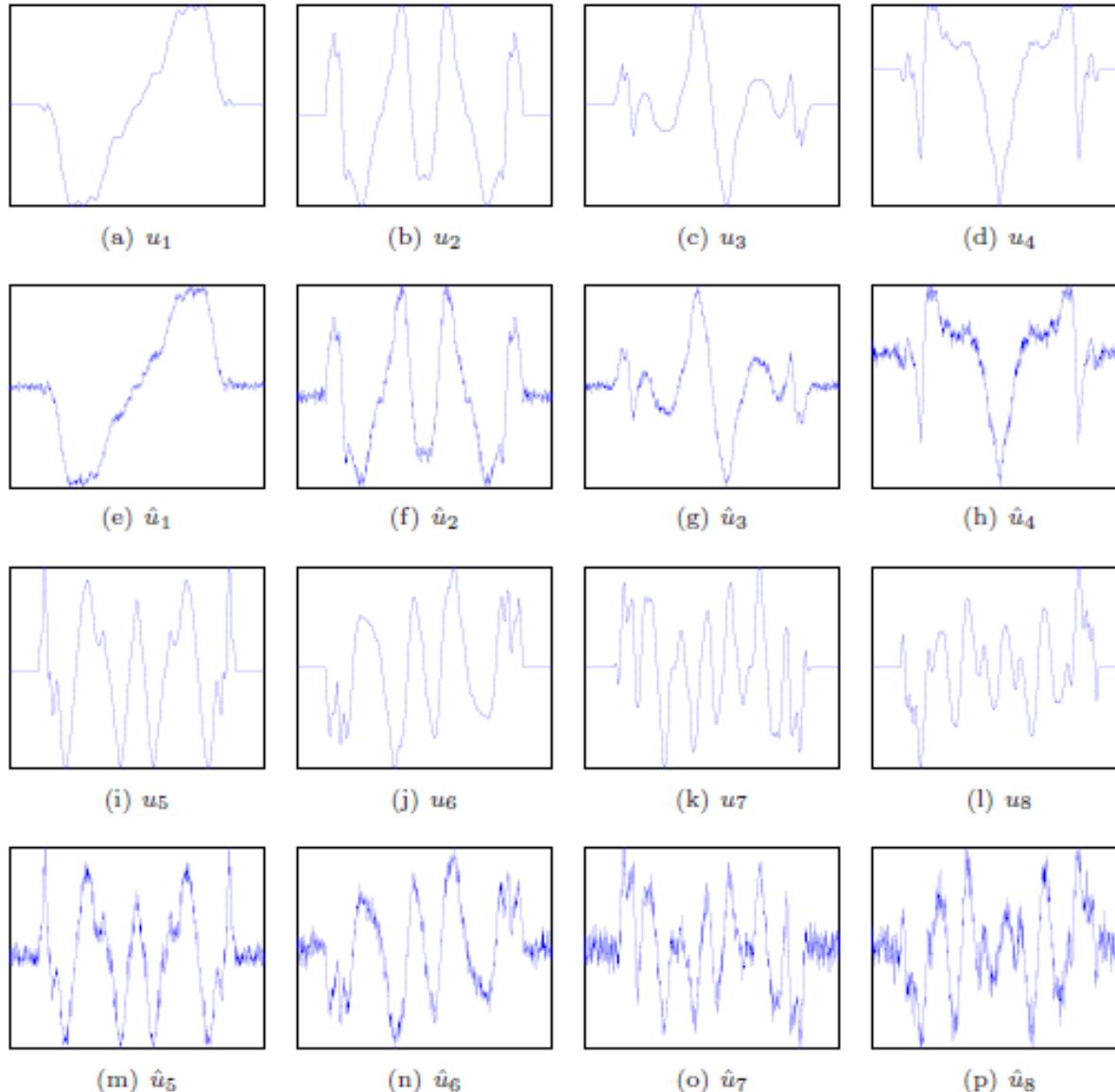


FIG. 7.14. The first 8 principal components for clean projections of the abdominal CT image and the first 8 principal components for noisy projections with  $SNR = 8dB$ . Notice that the principal components are determined up to an arbitrary sign, and we choose the signs so that corresponding pairs of components are positively correlated.

[Image source: Singer and Wu, Two-dimensional tomography from noisy projections taken at unknown random directions](#)

# Use of PCA for denoising

- We have seen how PCA can be used to denoise individual 1D tomographic projections.
- But PCA can also be used to denoise sinograms which are 2D signals of the form  $g(\rho, \theta)$  which consist of a collection of tomographic projections in sorted order of angle of projection  $\theta$ .
- In such cases, the patches will be 2D. But other principles remain the same.

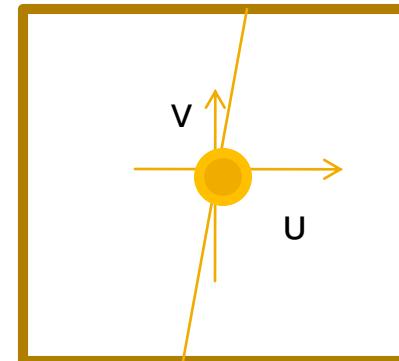
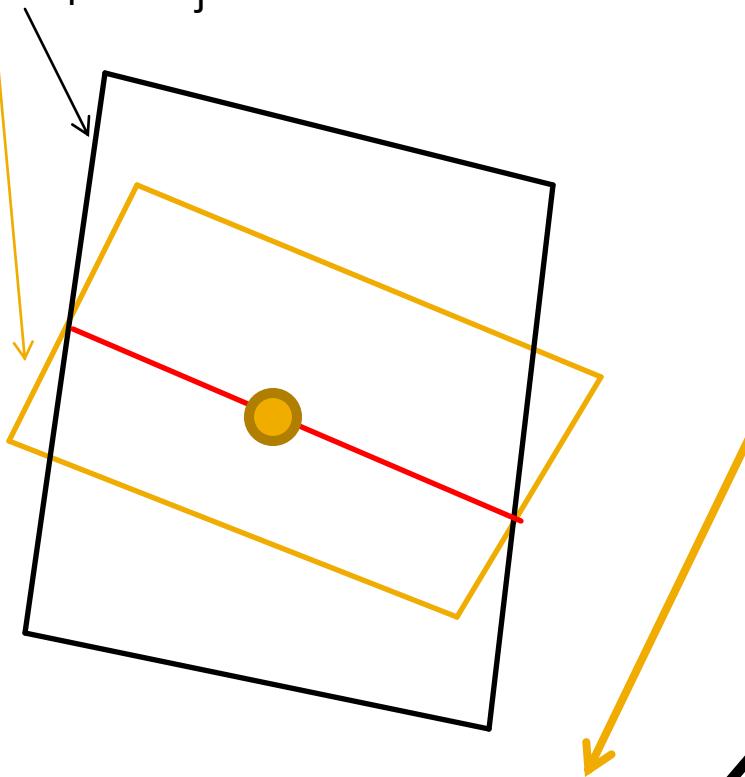
# Tomography under unknown angles: 3D images

- In this case, the image  $\mathbf{f}$  is in 3D and each tomographic projection is in 2D.
- Per Fourier slice theorem, 2D Fourier transform of a 2D projection in direction  $\mathbf{d}$  is equal to the 2D central slice (with normal vector  $\mathbf{d}$ ) through the 3D Fourier transform of  $\mathbf{f}$ .
- But in 3D we actually have *some more information* – which is absent in 2D!

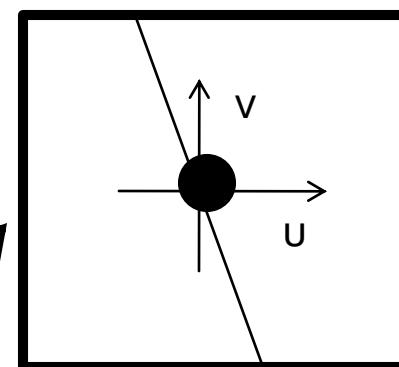
# Tomography under unknown angles: 3D images

- The planes of projection in directions  $d_i$  and  $d_j$  will intersect in a **common line**  $c_{ij}$ .
- Hence their corresponding central slices through the Fourier volume will also have a **common line**.
- The common line can be determined by a search in the Fourier space (or even in the original spatial domain)!

Central Fourier planes corresponding to directions  $d_i$  and  $d_j$



$$r_i = F(R_{d_i} f)$$



$$r_j = F(R_{d_j} f)$$

Intersection of the planes in the Fourier space leads to a common line. The directions of projection are unknown, but the common line (in the two local coordinate systems) can be found out by searching over pairs of directions in the Fourier space and finding the best match.

Note:  $r_i$  and  $r_j$  are Fourier transforms of the tomographic projections of image  $f$  in directions  $d_i$  and  $d_j$  respectively.

# The 3D case

- Consider the following equation:

$$\mathbf{R}_i \mathbf{c}_{ij} = \mathbf{b}_{ij} = (\cos \phi_{ij}, \sin \phi_{ij}, 0)^t$$

Angle made by  $\mathbf{c}_{ij}$  with local X axis

Rotation matrix giving the relationship between the global and  $i$ -th local coordinate system

Common line in global coordinate system

Common line expressed in local coordinate system of plane of projection in direction  $\mathbf{d}_i$

$$(\phi_{ij}, \phi_{ji}) = \arg \min_{\alpha, \beta} \|\mathbf{r}_i(\alpha) - \mathbf{r}_j(\beta)\|^2$$

Line through  $\mathbf{r}_i$  (in Fourier domain) passing through the origin and making an angle of  $\alpha$  with the U axis. See previous slide for definition of  $\mathbf{r}_i$

# The 3D case

- We see that

$$\mathbf{R}_i \mathbf{c}_{ij} = \mathbf{b}_{ij} = (\cos \phi_{ij}, \sin \phi_{ij}, 0)^t$$

$$\mathbf{b}_{ij} \bullet \mathbf{b}_{ik} = \cos(\phi_{ij} - \phi_{ik}) = \mathbf{R}_i \mathbf{c}_{ij} \bullet \mathbf{R}_i \mathbf{c}_{ik} = \mathbf{c}_{ij} \bullet \mathbf{c}_{ik}$$

- Consider the common lines  $\mathbf{c}_{12}, \mathbf{c}_{23}, \mathbf{c}_{31}$  for 3 viewing directions  $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ . Then we have

$$\mathbf{C} = (\mathbf{c}_{12} \quad \mathbf{c}_{23} \quad \mathbf{c}_{31}), \mathbf{M} = \mathbf{CC}^T$$

$$\mathbf{M} = \begin{pmatrix} 1 & \cos(\phi_{31} - \phi_{32}) & \cos(\phi_{23} - \phi_{21}) \\ \cos(\phi_{31} - \phi_{32}) & 1 & \cos(\phi_{12} - \phi_{13}) \\ \cos(\phi_{23} - \phi_{21}) & \cos(\phi_{12} - \phi_{13}) & 1 \end{pmatrix}$$

# The 3D case

- The matrix  $\mathbf{C}$  is rank 3 in the case that the common lines are not coplanar.
- To obtain  $\mathbf{C}$  from  $\mathbf{M}$ , we have

$$\mathbf{M} = \mathbf{CC}^T = \mathbf{UDU}^T \rightarrow \mathbf{C} = \mathbf{UD}^{0.5} \text{ (upto rotation/reflection ambiguity)}$$

$\mathbf{M}$  will be positive semi-definite with unit diagonal entries

# Rotation estimation

- We have the following relation:

$$\mathbf{R}_i \mathbf{C}_i = \mathbf{B}_i$$

This is a  $3 \times 2$  matrix of common lines for the  $i$ th direction expressed in a global coordinate system

This is a  $3 \times 2$  matrix of common lines involving expressed in the  $i$ -th local coordinate system

$$\mathbf{B}_i = \begin{pmatrix} \mathbf{b}_{ij} & \mathbf{b}_{ik} \end{pmatrix}$$

# Rigid transform estimation

- (dropping subscript  $i$  for convenience) To estimate  $\mathbf{R}$ :

$$E(\mathbf{R}) = \|\mathbf{RC} - \mathbf{B}\|_F^2 \text{ such that } \mathbf{RR}^T = I$$

Solution : Consider the SVD of  $\mathbf{BC}^T = \mathbf{USV}^T$

$$\mathbf{R}^T = \mathbf{UV}^T$$

This is called the **Orthogonal Procrustes problem**.  
See class handout for more details of this solution!

# Algorithm

- For all pairs of directions ( $\mathbf{d}_i, \mathbf{d}_j$ ) find the angle  $\phi_{ij}$ .  
Here  $i, j$  lie in {1,2,3}.
- Assemble the matrix  $\mathbf{B}$  and the matrix  $\mathbf{M}$  using the knowledge of the angles  $\phi_{ij}$ .
- Use eigen-value decomposition on  $\mathbf{M}$  to get  $\mathbf{C}$  and hence  $\{\mathbf{C}_i\}$ .
- Use the SVD method to obtain  $\mathbf{R}_i$  from  $\mathbf{C}_i$  and  $\mathbf{B}_i$ .
- That gives us the directions of tomographic projection!
- This algorithm is for a triple of 3 directions.
- We will later see an algorithm for arbitrarily many ( $N$ ) directions.

# Algorithm with N projections

- We have the following for any pair of projections (i,j):  $\mathbf{R}_i \mathbf{b}_{ij} = \mathbf{R}_j \mathbf{b}_{ji}$

- Main objective function:

$$\min_{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_N} \sum_{i \neq j} \left\| \mathbf{R}_i \mathbf{b}_{ij} - \mathbf{R}_j \mathbf{b}_{ji} \right\|^2 \text{ such that}$$

$$\forall i, \mathbf{R}_i^T \mathbf{R}_i = \mathbf{I}, \det(\mathbf{R}_i) = 1$$

The constraints are essential, otherwise the trivial solution will collapse all rotations to a 0 matrix.

- Equivalent form:

[https://web.math.princeton.edu/~amits/publications/common\\_lines\\_published.pdf](https://web.math.princeton.edu/~amits/publications/common_lines_published.pdf)

$$\max_{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_N} \sum_{i \neq j} \mathbf{R}_i \mathbf{b}_{ij} \bullet \mathbf{R}_j \mathbf{b}_{ji} \text{ such that}$$

$$\forall i, \mathbf{R}_i^T \mathbf{R}_i = \mathbf{I}, \det(\mathbf{R}_i) = 1$$

# Algorithm with N projections

- Define the following:

$$\underline{\boldsymbol{R}^1 = (R_1^1 \mid R_2^1 \mid \dots \mid R_N^1), \boldsymbol{R}^2 = (R_1^2 \mid R_2^2 \mid \dots \mid R_N^2)}$$

3 x N matrix created by concatenating columns of individual rotation matrices

$$R_j^i = \text{column } i \text{ of rotation matrix } R_j$$

- Define the following 3 x 2N matrix as well:

$$\mathbf{R} = (\boldsymbol{R}^1 \mid \boldsymbol{R}^2) = (R_1^1 \mid R_2^1 \mid \dots \mid R_N^1 \mid R_1^2 \mid R_2^2 \mid \dots \mid R_N^2)$$

- Define the following 2N x 2N Gram matrix:

$$\mathbf{G} = \mathbf{R}^T \mathbf{R}$$

# Algorithm with N projections

- $\mathbf{G}$  is clearly a rank-3 positive semi-definite matrix, which can be represented in block form as:

$$\mathbf{G} = \mathbf{R}^T \mathbf{R} = \begin{pmatrix} \mathbf{G}^{11} & \mathbf{G}^{12} \\ \mathbf{G}^{21} & \mathbf{G}^{22} \end{pmatrix} = \begin{pmatrix} \mathbf{R}^{1^T} \mathbf{R}^1 & \mathbf{R}^{1^T} \mathbf{R}^2 \\ \mathbf{R}^{2^T} \mathbf{R}^1 & \mathbf{R}^{2^T} \mathbf{R}^2 \end{pmatrix}$$

$$G_{ii}^{11} = G_{ii}^{22} = 1, G_{ii}^{12} = G_{ii}^{21} = 0$$

# Algorithm with N projections

- Further define  $2N \times 2N$  matrix  $\mathbf{S}$  as follows:

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}^{11} & \mathbf{S}^{12} \\ \mathbf{S}^{21} & \mathbf{S}^{22} \end{pmatrix}, \mathbf{b}_{ij} = (x_{ij}, y_{ij}, 0)$$

$$S_{ij}^{11} = x_{ij}x_{ji}, S_{ij}^{12} = x_{ij}y_{ji}, S_{ij}^{21} = y_{ij}x_{ji}, S_{ij}^{22} = y_{ij}y_{ji}$$

$$S_{ii}^{11} = S_{ii}^{22} = S_{ii}^{12} = S_{ii}^{21} = 0$$

- $\mathbf{S}$  is a symmetric matrix and stores information about the common lines

# Algorithm with N projections

- Now consider the objective function:

$$\max_{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_N} \sum_{i \neq j} \mathbf{R}_i \mathbf{b}_{ij} \bullet \mathbf{R}_j \mathbf{b}_{ji} \text{ such that}$$

$$\forall i, \mathbf{R}_i^T \mathbf{R}_i = \mathbf{I}, \det(\mathbf{R}_i) = 1$$

- The objective function can be proved to be equal to *trace* ( $\mathbf{S}\mathbf{G}$ ) subject to the same constraints.
- See next slide for proof.

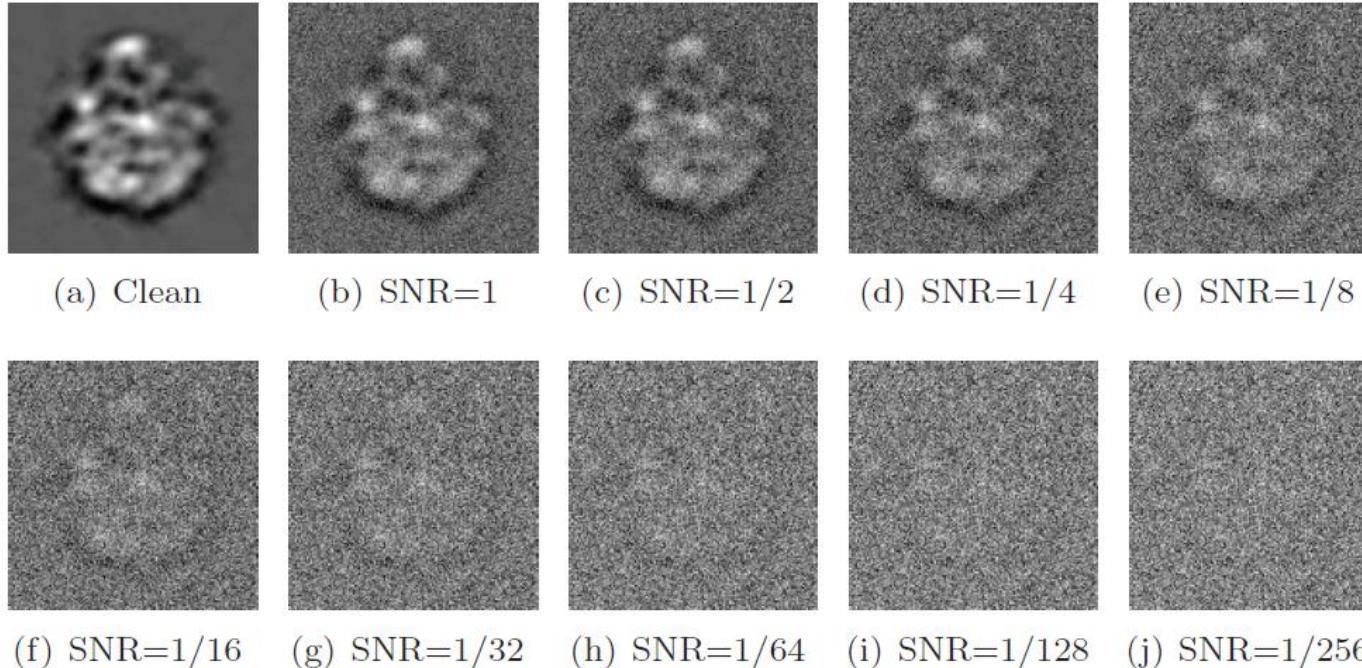
# Algorithm with N projections

$$\begin{aligned} \sum_{i \neq j} R_i b_{ij} \bullet R_j b_{ji} &= \sum_{i \neq j} x_{ij} x_{ji} R_i^1 \bullet R_j^1 + \sum_{i \neq j} x_{ij} y_{ji} R_i^1 \bullet R_j^2 + \\ &\quad \sum_{i \neq j} y_{ij} x_{ji} R_i^2 \bullet R_j^1 + \sum_{i \neq j} y_{ij} y_{ji} R_i^2 \bullet R_j^2 \\ &= \sum_{i \neq j} S_{ij}^{11} R_i^1 \bullet R_j^1 + \sum_{i \neq j} S_{ij}^{12} R_i^1 \bullet R_j^2 + \\ &\quad \sum_{i \neq j} S_{ij}^{21} R_i^2 \bullet R_j^1 + \sum_{i \neq j} S_{ij}^{22} R_i^2 \bullet R_j^2 = \text{trace}(SG) \end{aligned}$$

# Algorithm with N projections

- Final objective function:  $\max_G \text{trace}(\mathbf{S}\mathbf{G})$  such that  $\mathbf{G} \succ \mathbf{0}$ ,  
 $\forall i, G_{ii}^{11} = G_{ii}^{22} = 1, G_{ii}^{12} = G_{ii}^{21} = 0$
- This turns out to be a “semi-definite program”, for which efficient solutions exist in packages such as CVX.
- $\mathbf{R}$  (and hence all rotation matrices) is recovered from a Cholesky decomposition of  $\mathbf{G}$ .
- The rank of the estimate  $\mathbf{G}$  may be more than 3, in which case  $\mathbf{R}$  is projected onto the subspace spanned by the first three eigenvectors of  $\mathbf{G}$ .

# Numerical Results



**Figure 4.** Simulated projection with various levels of additive Gaussian white noise.

[https://web.math.princeton.edu/~amits/publications/common\\_lines\\_published.pdf](https://web.math.princeton.edu/~amits/publications/common_lines_published.pdf)

# Numerical Results

(a)  $N = 100$

$p$	MSE(eig)	MSE(sdp)
1	0.0055	4.8425e-05
0.5	0.0841	0.0676
0.25	0.7189	0.7140
0.15	2.8772	2.8305
0.1	4.5866	4.7814
0.05	4.8029	5.1809

(b)  $N = 500$

$p$	MSE(eig)	MSE(sdp)
1	0.0019	1.0169e-05
0.5	0.0166	0.0143
0.25	0.0973	0.0911
0.15	0.3537	0.3298
0.1	1.2739	1.1185
0.05	5.4371	5.3568

[https://web.math.princeton.edu/~amits/publications/common\\_lines\\_published.pdf](https://web.math.princeton.edu/~amits/publications/common_lines_published.pdf)

$p$  = proportion of correctly detected common lines. Due to noise in the particle images, the common lines may not be correctly detected.

# Numerical Results

**Table 2**

The proportion  $p$  of correctly detected common lines as a function of the SNR. As expected,  $p$  is not a function of the number of images  $N$ .

(a)  $N = 100$

SNR	$p$
clean	0.997
1	0.968
1/2	0.930
1/4	0.828
1/8	0.653
1/16	0.444
1/32	0.247
1/64	0.108
1/128	0.046
1/256	0.023
1/512	0.017

(b)  $N = 500$

SNR	$p$
clean	0.997
1	0.967
1/2	0.922
1/4	0.817
1/8	0.639
1/16	0.433
1/32	0.248
1/64	0.113
1/128	0.046
1/256	0.023
1/512	0.015

(c)  $N = 1000$

SNR	$p$
clean	0.997
1	0.966
1/2	0.919
1/4	0.813
1/8	0.638
1/16	0.437
1/32	0.252
1/64	0.115
1/128	0.047
1/256	0.023
1/512	0.015

# Numerical Results

**Table 3**

The MSE of the eigenvector and SDP methods for  $N = 100$ ,  $N = 500$ , and  $N = 1000$ .

(a)  $N = 100$

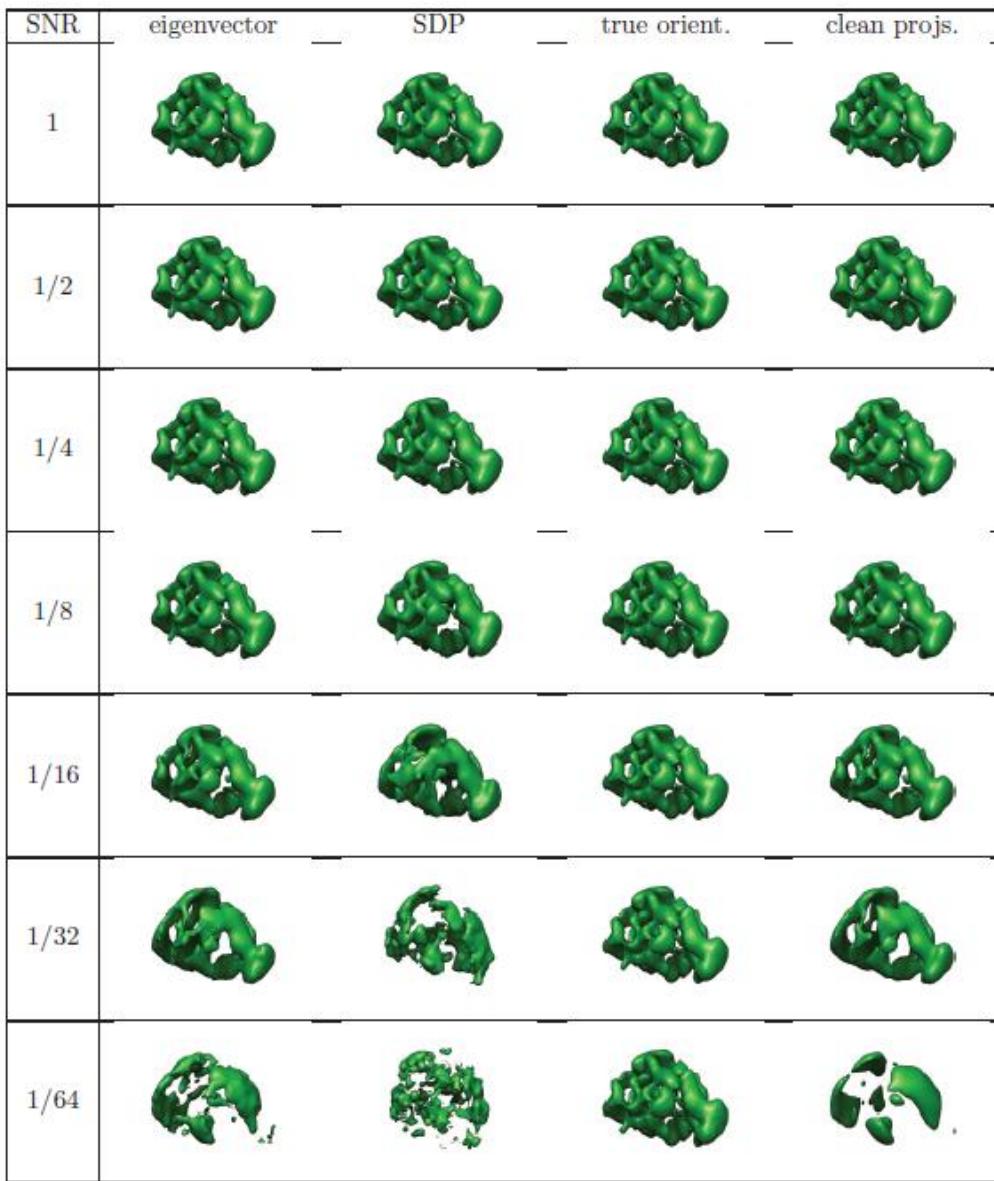
SNR	MSE(eig)	MSE(sdp)
1	0.0054	3.3227e-04
1/2	0.0068	0.0016
1/4	0.0129	0.0097
1/8	0.0276	0.0471
1/16	0.0733	0.1951
1/32	0.2401	0.6035
1/64	2.5761	1.9509
1/128	3.2014	3.1020
1/256	4.0974	4.1163
1/512	4.9664	4.9702

(b)  $N = 500$

SNR	MSE(eig)	MSE(sdp)
1	0.0023	2.4543e-04
1/2	0.0030	0.0011
1/4	0.0069	0.0071
1/8	0.0203	0.0414
1/16	0.0563	0.1844
1/32	0.1859	0.6759
1/64	1.7549	1.3668
1/128	2.6214	2.4046
1/256	3.4789	3.3539
1/512	4.6027	4.5089

(c)  $N = 1000$

SNR	MSE(eig)	MSE(sdp)
1	0.0018	2.3827e-04
1/2	0.0030	0.0011
1/4	0.0072	0.0067
1/8	0.0208	0.0406
1/16	0.0582	0.1899
1/32	0.1996	0.7077
1/64	1.7988	1.5370
1/128	2.5159	2.3243
1/256	3.5160	3.4365
1/512	4.6434	4.6013



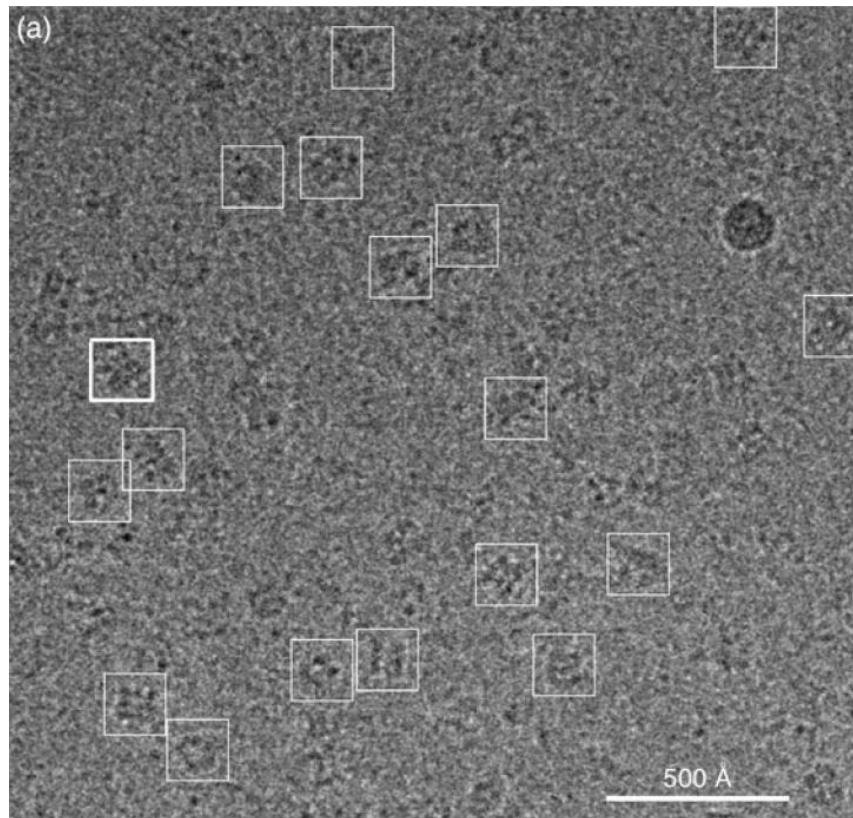
**Figure 8.** Reconstruction from  $N = 1000$  noisy projections at various SNR levels using the eigenvector method. Left column: reconstructions generated from noisy projections and orientations estimated using the eigenvector method. Middle-left column: reconstructions generated from noisy projections and orientations estimated using the SDP method. Middle-right column: reconstructions from noisy projections and the true orientations. Right column: reconstructions from estimated orientations (using the eigenvector method) and clean projections.

# Cryo-EM pipeline

- Cryo-EM reconstruction is typically referred to as “**single particle cryo-EM**” or SPR.
- This is so, even though we all know that a “single particle” is not going to be enough – we will have thousands of particles in a single micrograph, and sometimes, we will have multiple micrographs.

# Cryo-EM pipeline

- (1) Particle picking



<https://www.ncbi.nlm.nih.gov/pubmed/26705325>

# Cryo-EM pipeline

- (2) CTF correction: The micrograph observed is a blurred one, because of the blur induced by the electron microscope.
- The Fourier transform of the blur kernel is called the contrast transfer function (CTF). Correcting for this is a deblurring problem called CTF correction:

$$F(I_{obs}) = F(k)F(I_{true}) + F(\eta)$$

$$I_{obs} = k * I_{true} + \eta$$

# Cryo-EM pipeline

- (3) Particle clustering: The deblurred particles are clustered using any standard technique and then averaged.
- The averaging reduces a lot of the noise.
- We start from  $N$  particles and arrive at some  $K$  cluster centers,  $K \ll N$ .
- In some techniques, a coarse angle assignment was done assuming a known reference 3D structure. These techniques however suffer from a model bias.

# Cryo-EM pipeline

- (4) View parameter determination and 3D structure estimation: The 3D rotation and translation parameters are first estimated, followed by FBP.
- These processes are then repeated in an alternating fashion. The overall cost function

$$\text{is: } J(f, \{A_i = (\mathbf{R}_i | t_i)\}_{i=1}^N) = \sum_{i=1}^N \|y_i - \mathcal{R}(A_i f)\|_2^2 - \lambda \sum_{i,j, i \neq j} \mathbf{R}_i b_{ij} \bullet \mathbf{R}_j b_{ji}$$
$$s.t. \forall i, (\mathbf{R}_i)^t \mathbf{R}_i = I$$

# Cryo-electron tomography

- Some types of macromolecules tend to occur in a fairly narrow range of orientations.
- In such cases, 3D reconstruction becomes very difficult due to inadequate filling up of the Fourier space (due to Fourier slice theorem).
- Hence, in these cases, the slide containing the macromolecules is **tilted** in 3D space to different positions.
- Each time, a new micrograph is recorded. This is called **cryo-electron tomography**.

[https://link.springer.com/content/pdf/10.1007%2F978-3-030-00069-1\\_4.pdf](https://link.springer.com/content/pdf/10.1007%2F978-3-030-00069-1_4.pdf)

# Cryo-electron tomography

- Note the confusing terminology – in fact, even SPR is a form of tomographic reconstruction.
- This extra tilting gives rise to new effective orientations, and helps 3D reconstruction.
- Caveats: the SNR is now poorer because the beam power must be reduced.
- Sometimes, the tilting causes the macromolecules to move a little, and hence some motion correction needs to be performed.

# Cryo-electron tomography

- The tilting is typically done in increments of 0.5 degrees within a range of +/- 9 degrees.
- Sometimes, the tilts are bidirectional.