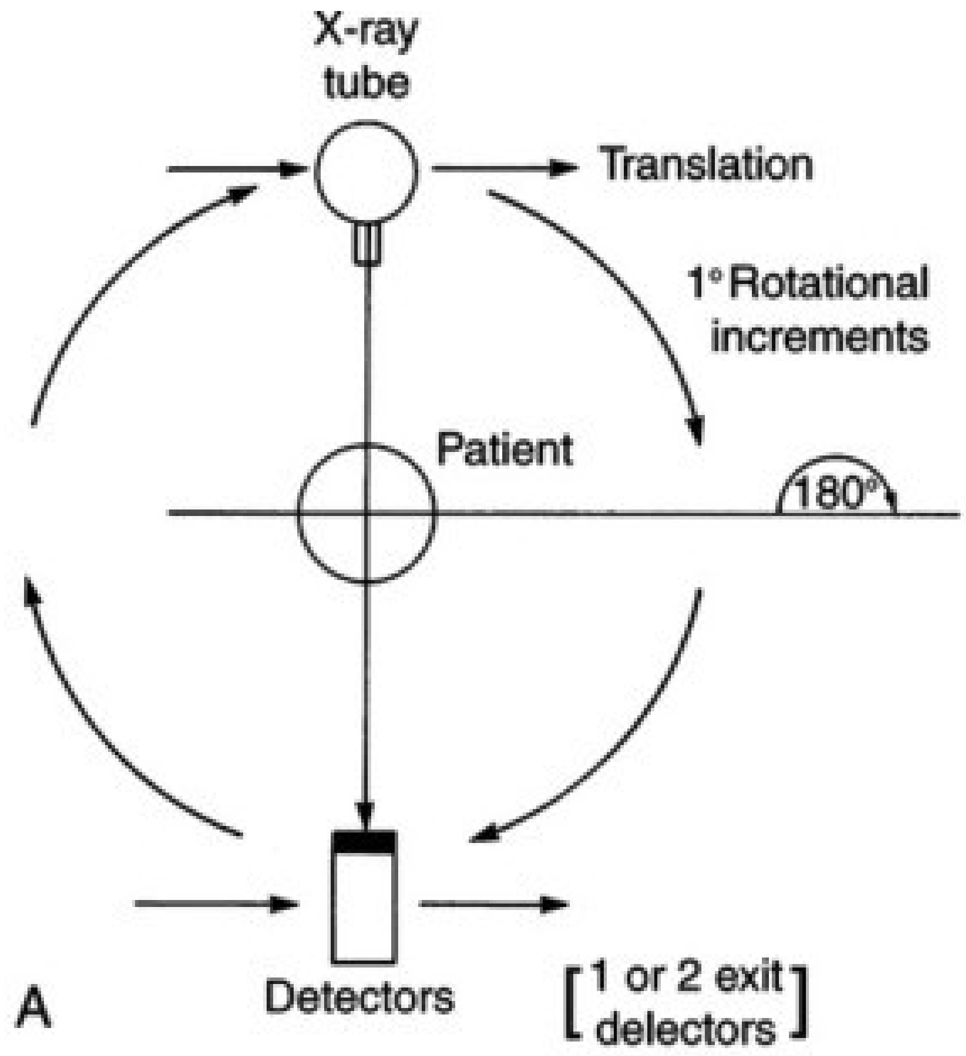
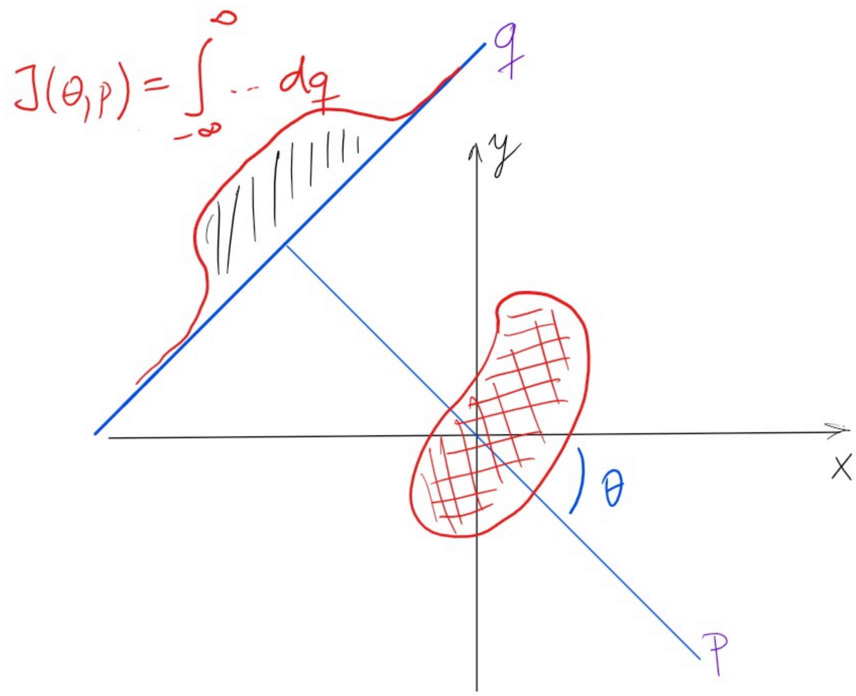
Lab09 CT Radon Transform

First generation CT



Radon transform



$$J(\theta,p)=\int_{-\infty}^{\infty} f\left(p\cdot\cos(\theta)-q\cdot\sin(\theta),\, p\cdot\sin(\theta)+q\cdot\cos(\theta)\right)\mathrm{d}q$$

θ – angle of normal vector
p – distance from the origin
q – distance alongside the projected line
$f()$ – *image function*

Image transformations

- Image – function defined over a grid
  I(x,y)
  x, **y** – coordinate vectors
  x×y – grid, cartesian coordinate system
- Image transformation
  transformation of the grid
  interpolation

Exercise 1.1

Find the smallest range $[-p_m, p_m] \times [-q_m, q_m]$ needed so that all points from $(x, y) \in [-x_m, x_m] \times [-y_m, y_m]$ fall into this range. Provide either a graphical or an analytical solution.

When rotating a rectangle, either the corners are clipped or the image size increases. In order to construct the sinogram properly, we have to keep the image dimensions fixed and avoid any loss of information.

Exercise 1.2

Discretize the Radon function

$$J(\theta, p) = \int_{-\infty}^{\infty} f(p \cdot \cos(\theta) - q \cdot \sin(\theta), p \cdot \sin(\theta) + q \cdot \cos(\theta)) \, \mathrm{d}q$$

Now, we are working with images composed of pixels, not abstract 2D functions.

HW Task 1.3

Implement *myRadon* function

inputs

img - image

theta - vector of angles

output

sinogram

useful functions

linspace, meshgrid, ndgrid, interp2,

sind, cosd, sum

imshow, imagesc

HW Task 1.4

Transform the Shepp-Logan phantom (matlab function phantom) of size 256 px for projection angles $\theta \in \{0°, 1°, 2°, \ldots, 179°\}$ with your function. Show the input and the resulting sinogram. Pay attention to labelling the axis correctly.

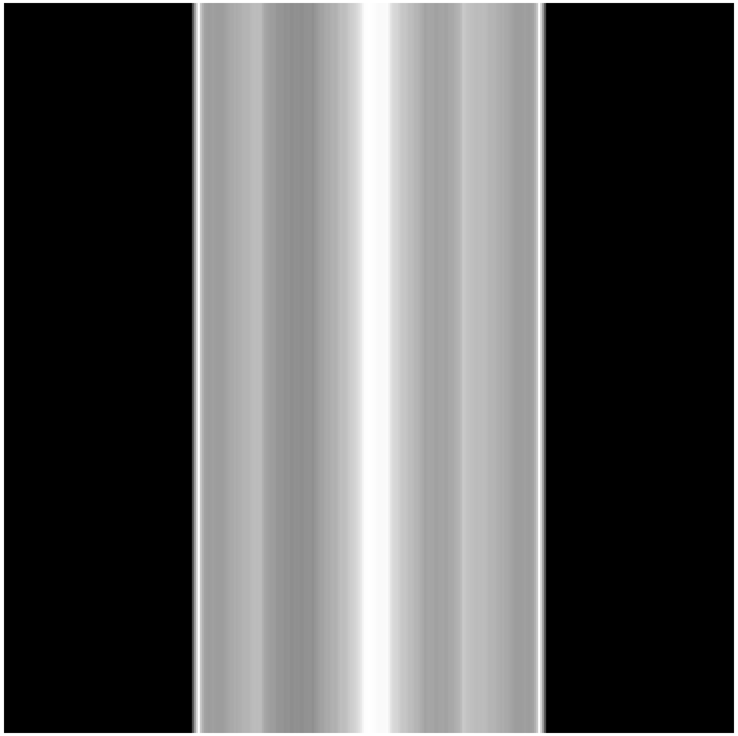
You may want to check your results against the results from function *radon*

Questions?

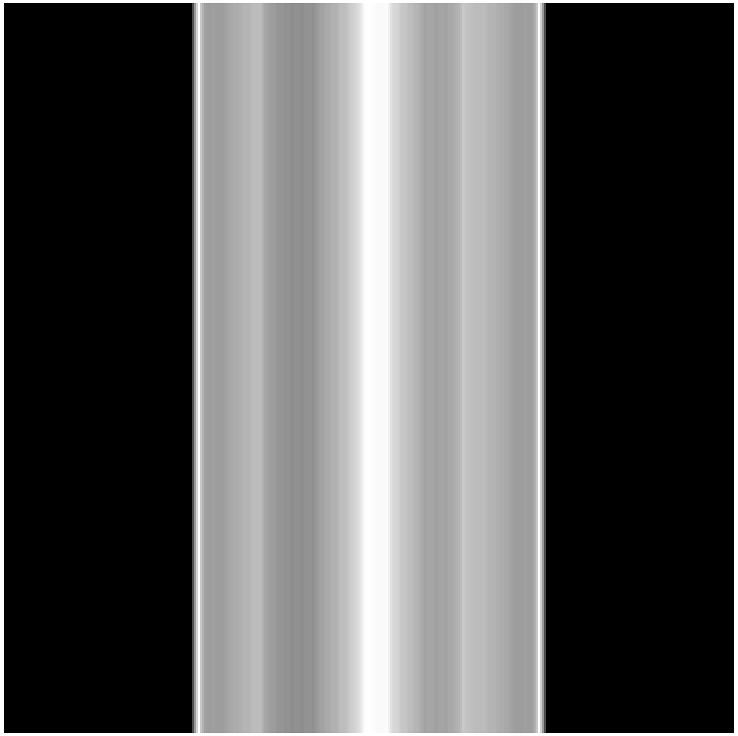
Naïve backprojection

- Creating projection images for each projection
- Distribute the projection values along an appropriate angle
- Summing all projection images

Naïve backprojection

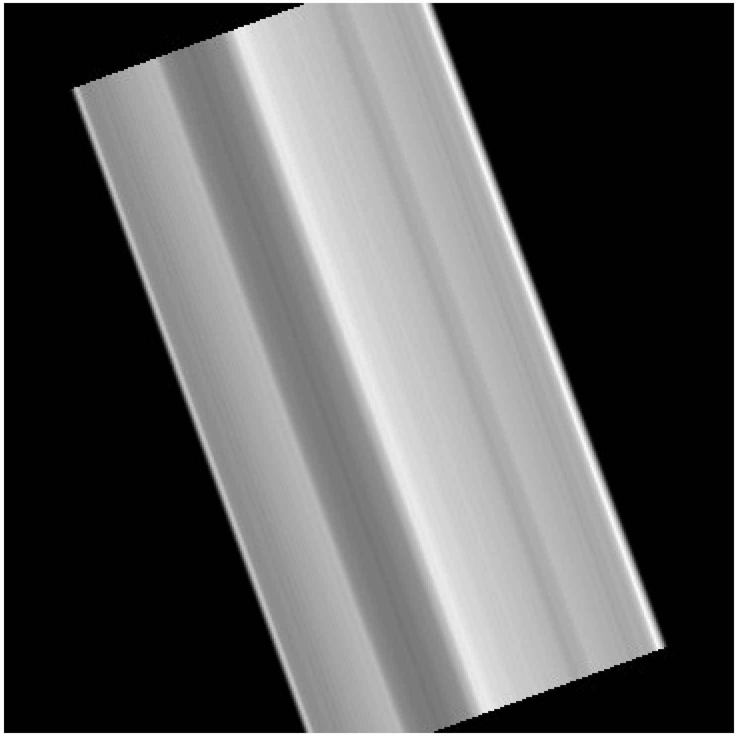


Projection image 0°

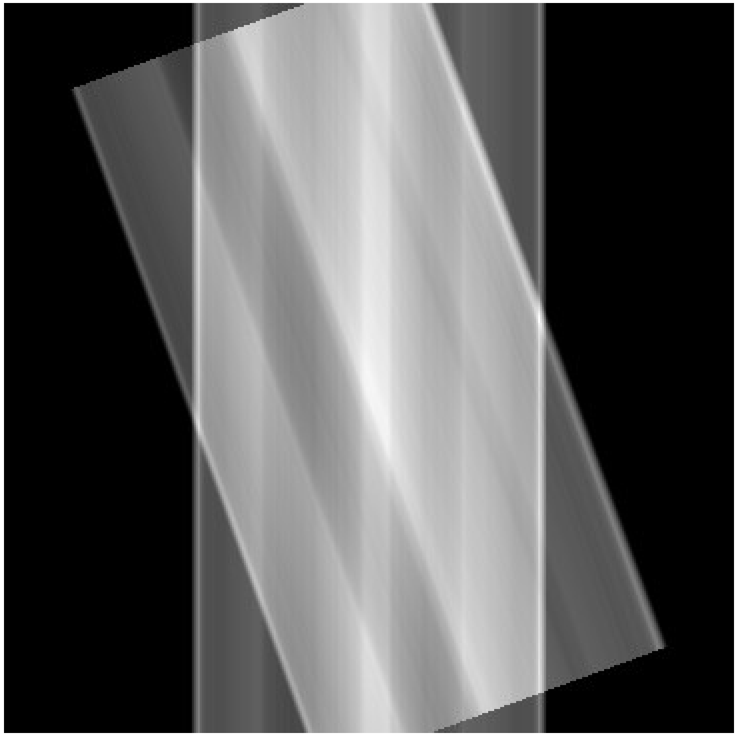


Backprojection (0°)

Naïve backprojection

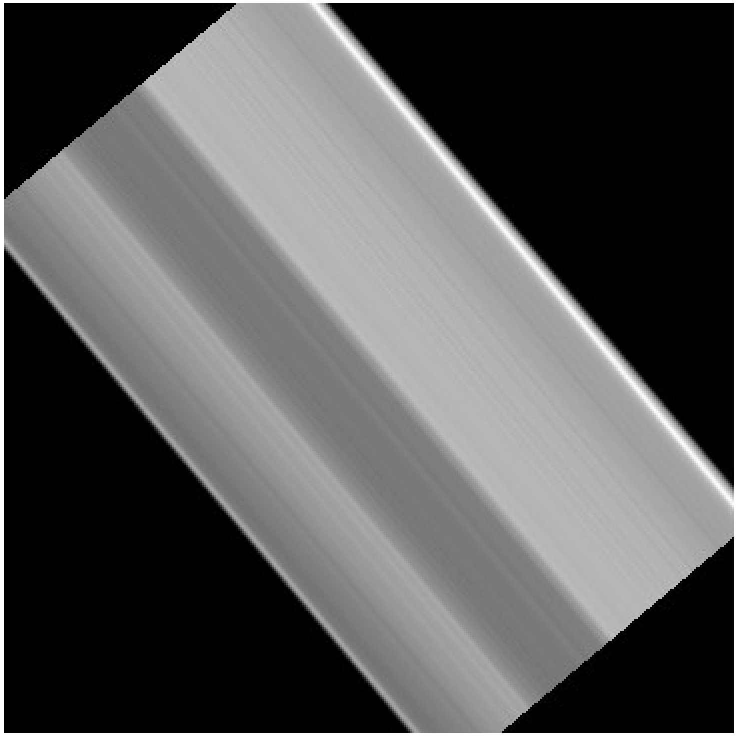


Projection image 20°

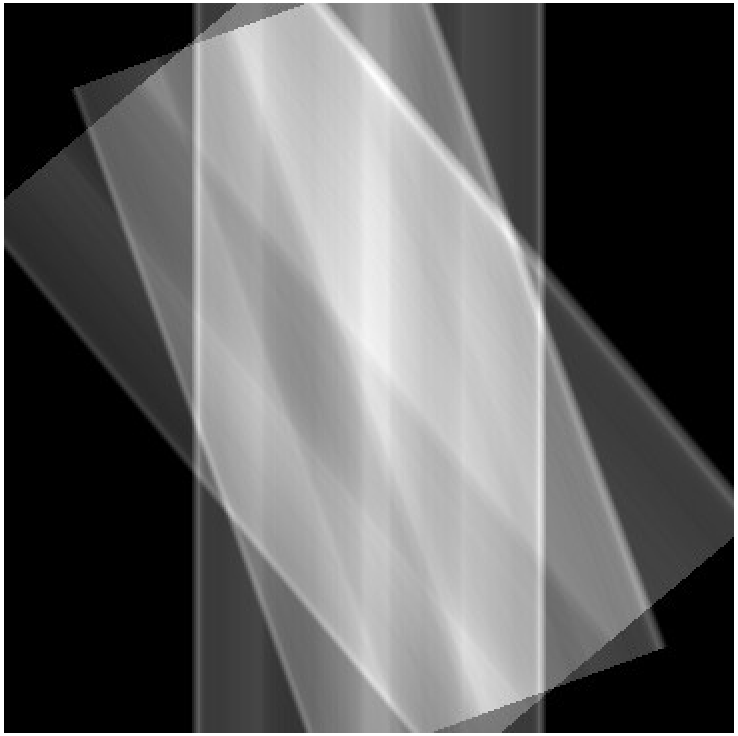


Backprojection 0°+ 20°

Naïve backprojection

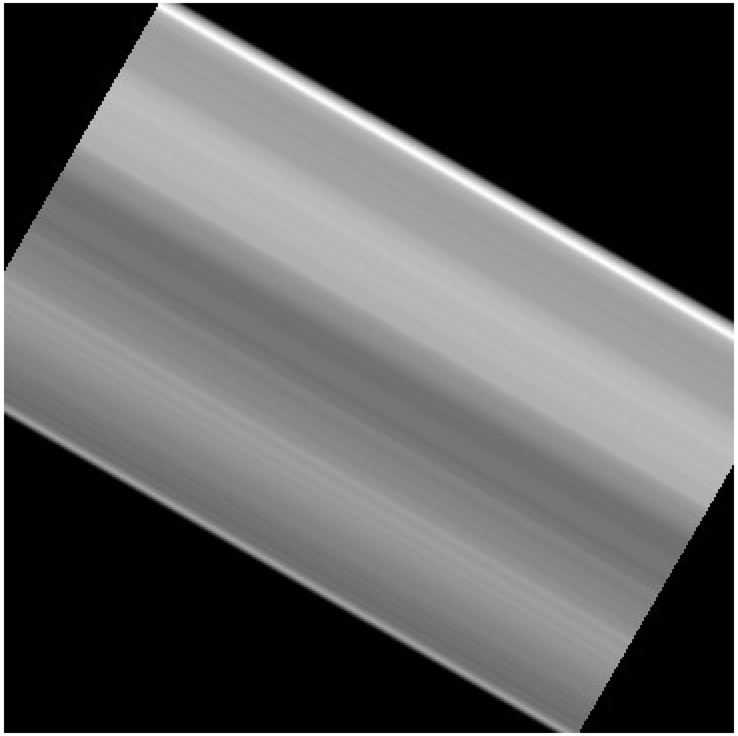


Projection image 40°

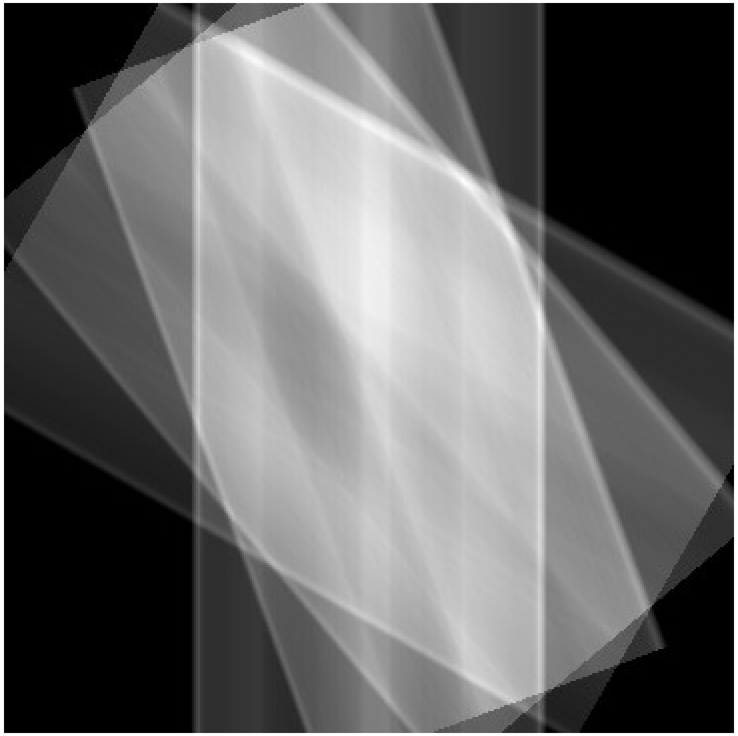


Backprojection (0°+ 20°+ 40°)

Naïve backprojection

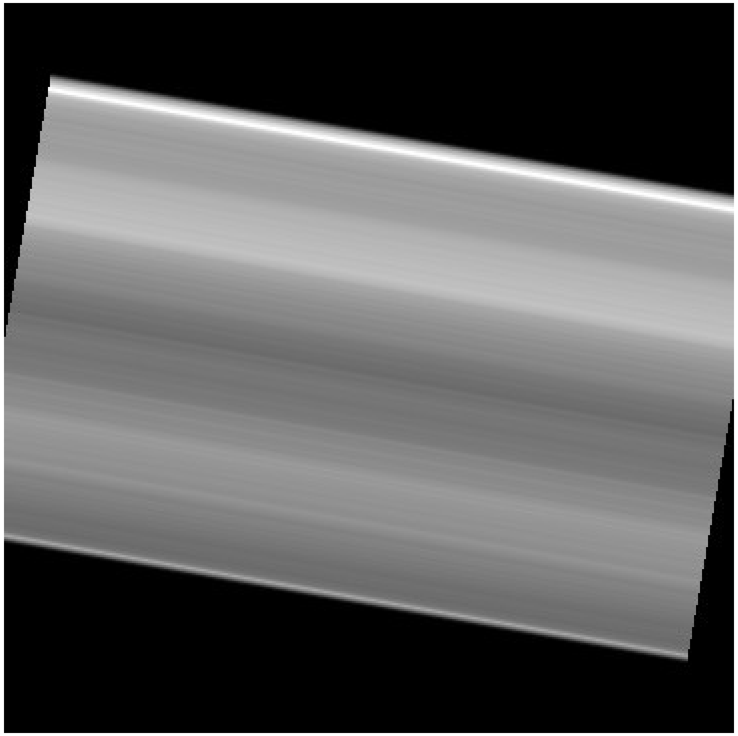


Projection image 60°

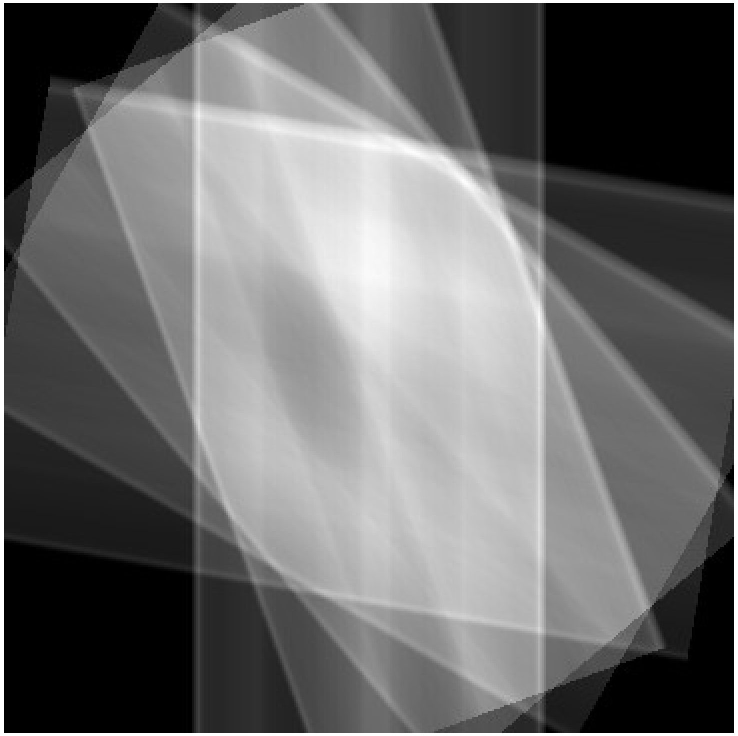


Backprojection (0° + … + 60°)

Naïve backprojection

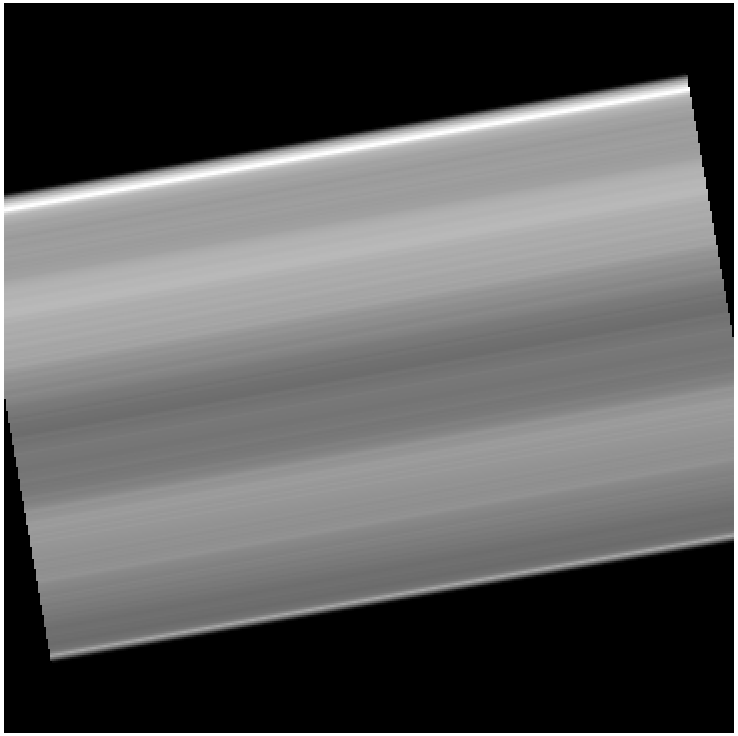


Projection image 80°

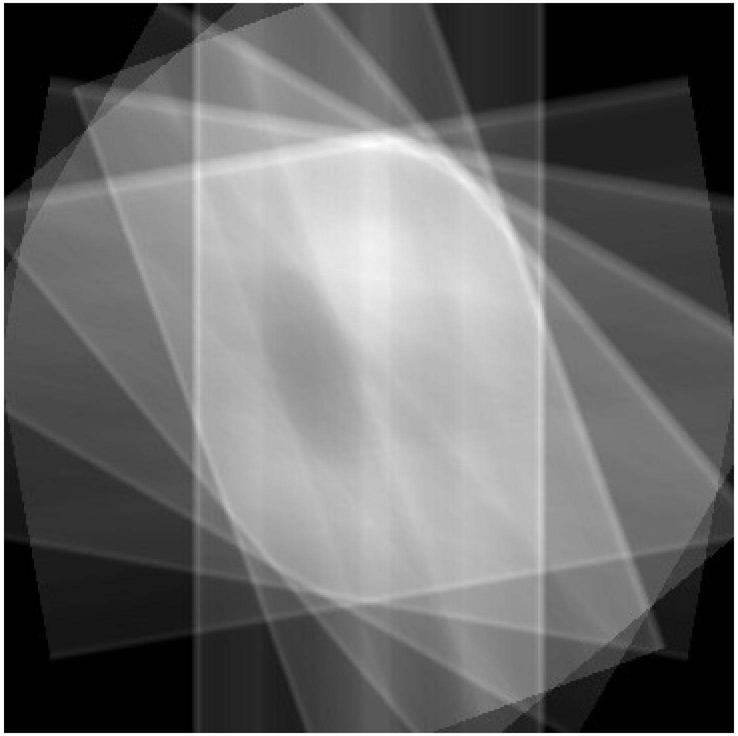


Backprojection (0° + … + 80°)

Naïve backprojection

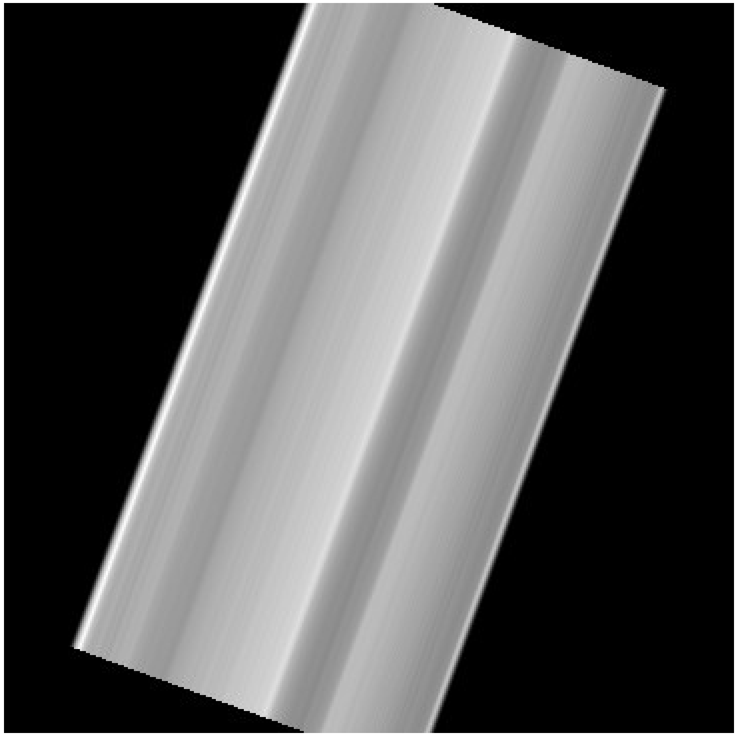


Projection image 100°

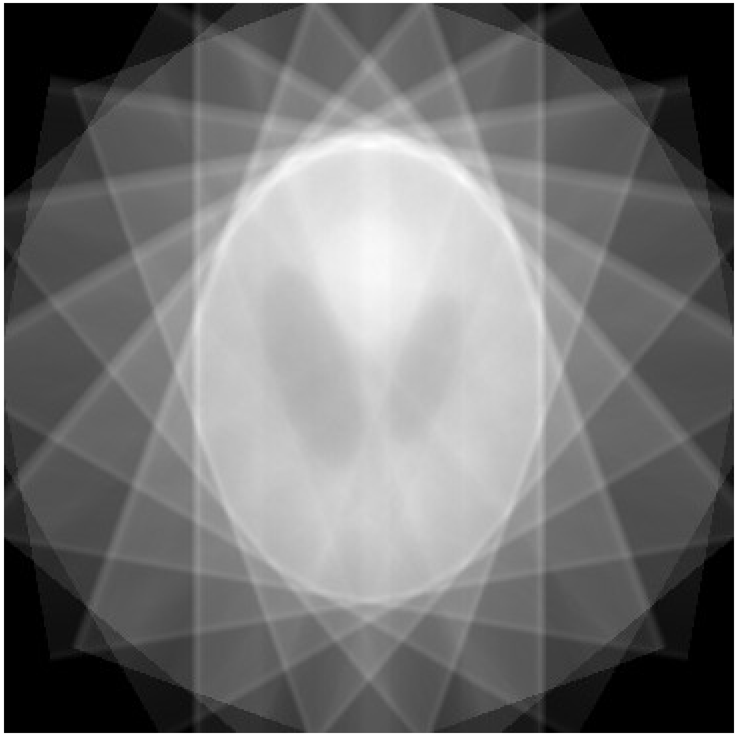


Backprojection (0° + … + 100°)

Naïve backprojection



Projection image 160°

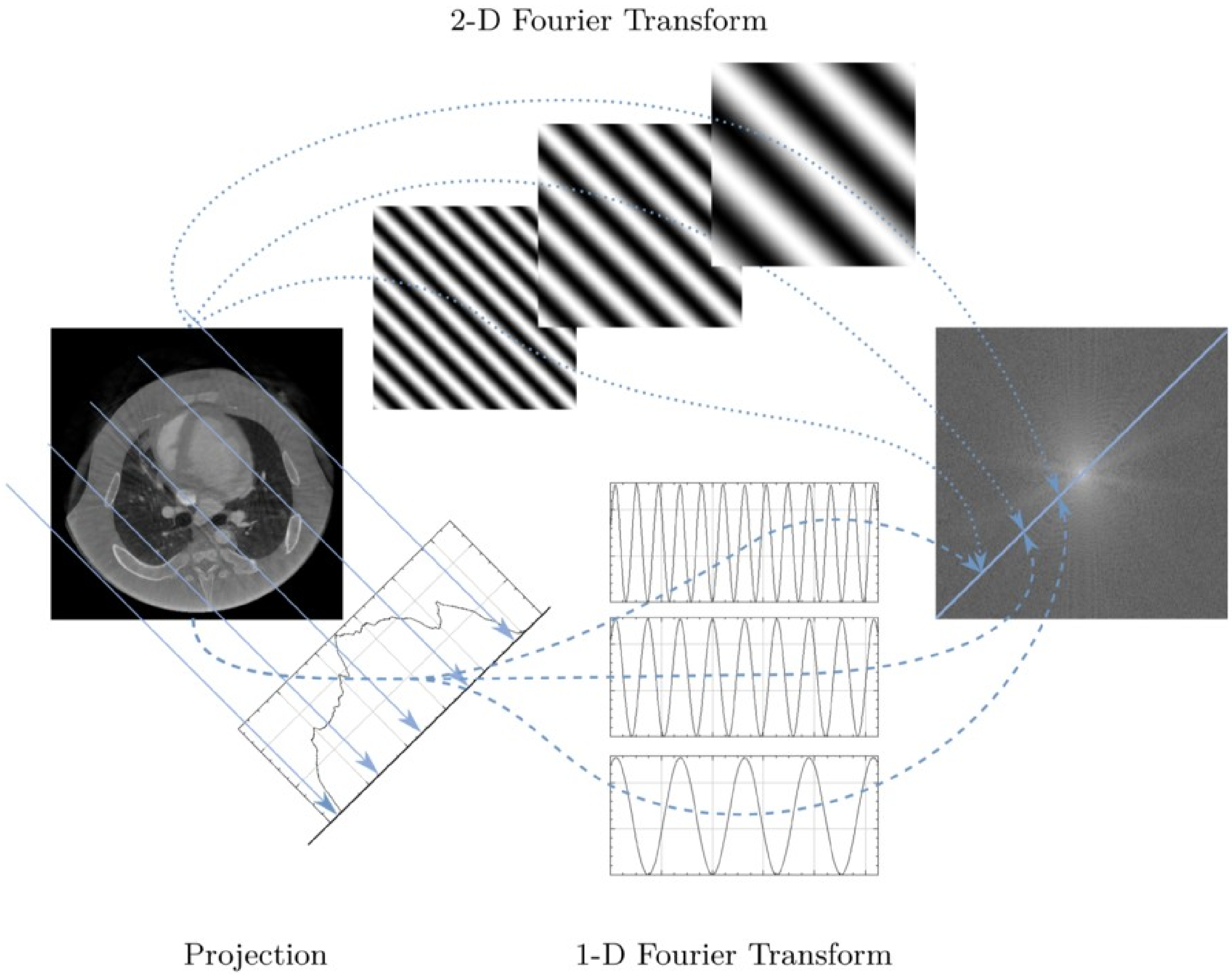


Backprojection (0° + … + 160°)

Filtered backprojection

- **Central slice theorem** - central slice of 2D Fourier transform of an image at an angle φ is equivalent to a 1D Fourier transform of a projection by Radon transform at the angle φ
- **Projections** obtained by Radon transform are in polar coordinates (Θ,p), when transforming to Cartesian coordinates, we have to compensate for the change by Jacobian matrix/determinant of Jacobian matrix when integrating => |ω| ramp filter

Filtered backprojection



Filtered backprojection

- Following the central slice theorem, we could implement the analytical solution - arrange the filtered projections at appropriate angles and perform inverse 2D Fourier transform
- Due to the uneven sampling (polar to Cartesian coordinates) this is not practical
- Filtered backprojection algorithm is used instead

Filtered backprojection

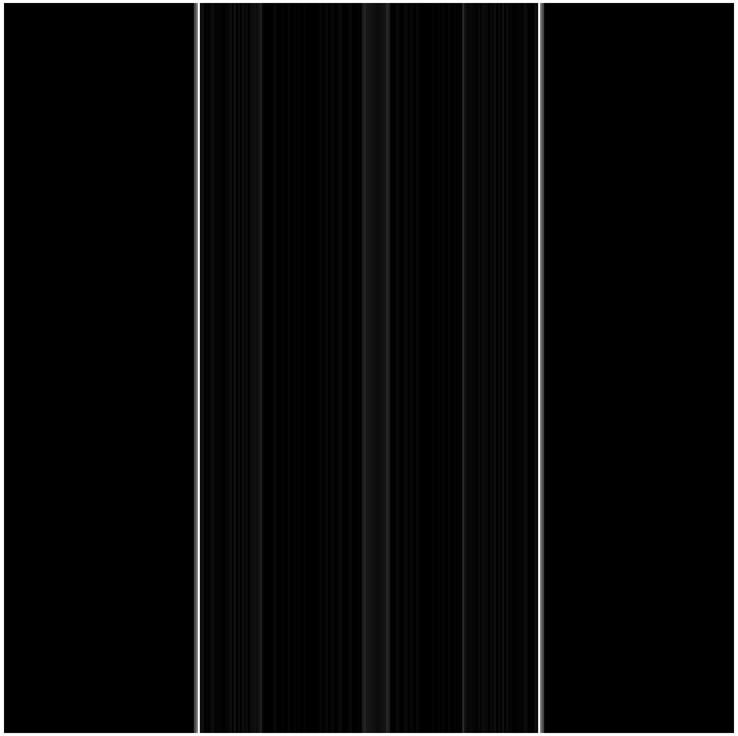
Algorithm

- The same backprojection approach as in the naïve backprojection
- Radon projections are filtered by a filter (ramp, ram-lak,hamming,...)
- Yields very good reconstruction of the image

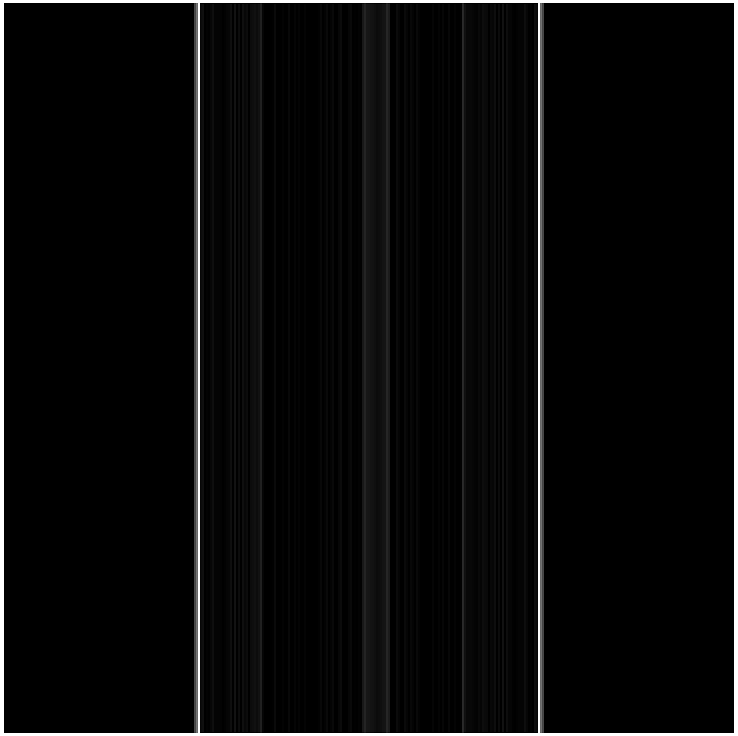
Filtered backprojection

- Digital filtering of Radon projections
  - convolution in time domain
  - multiplication in frequency domain
- Discrete Fourier transform
  - signals of finite length – window functions
  - zero padding

Naïve backprojection

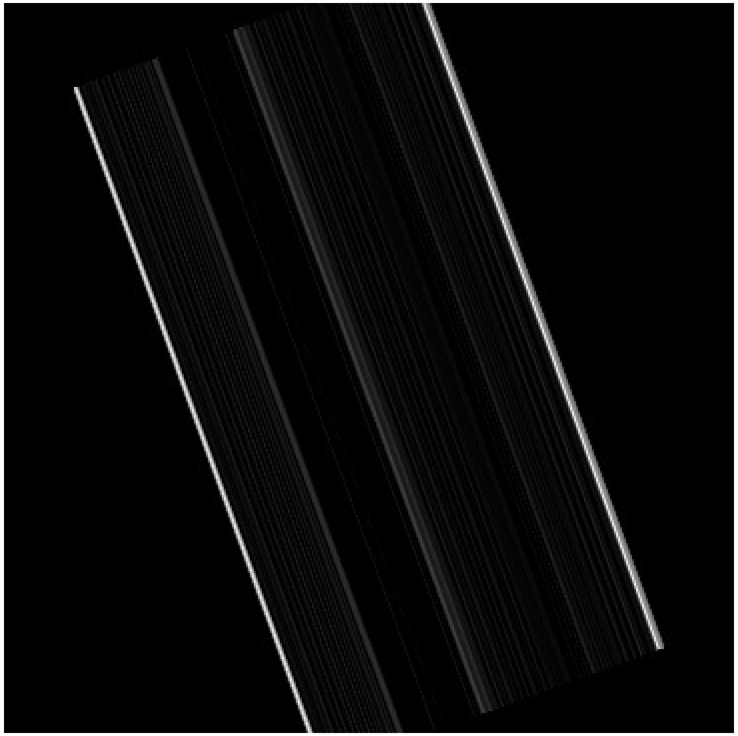


Projection image 0°

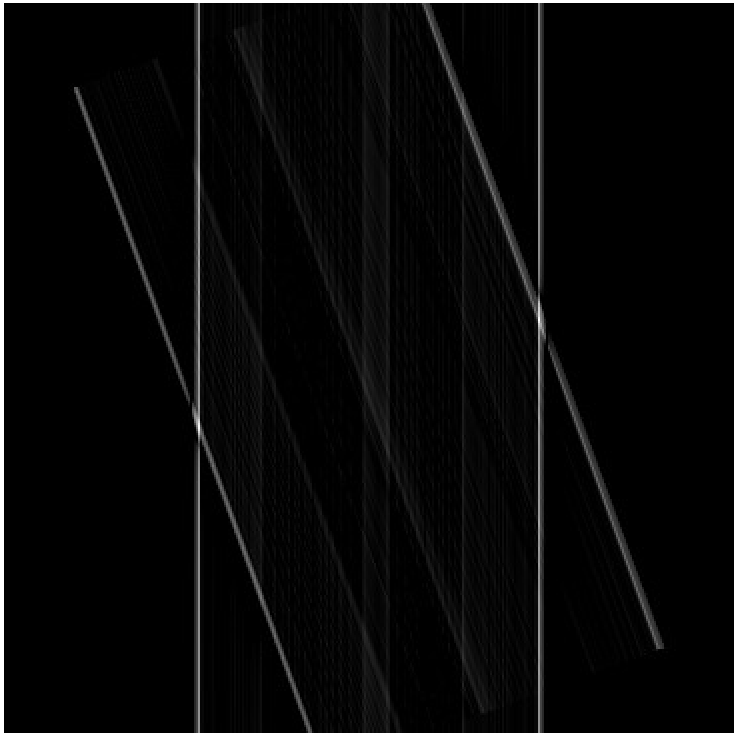


Backprojection (0°)

Naïve backprojection

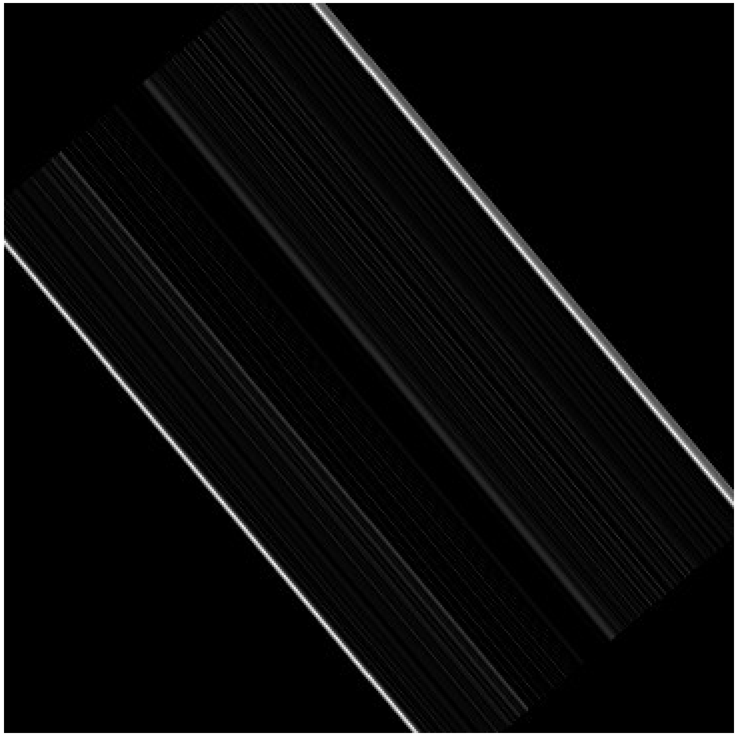


Projection image 20°

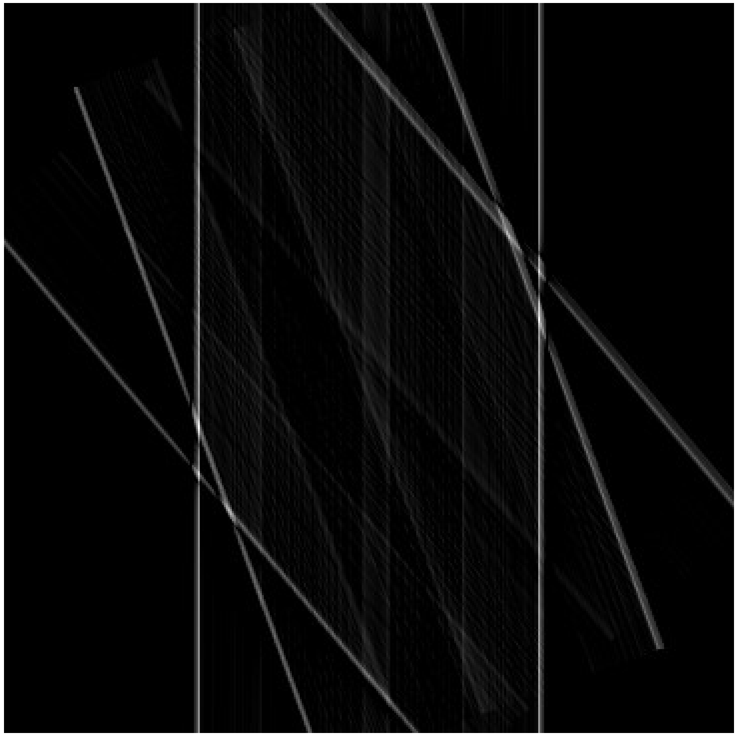


Backprojection 0°+ 20°

Naïve backprojection

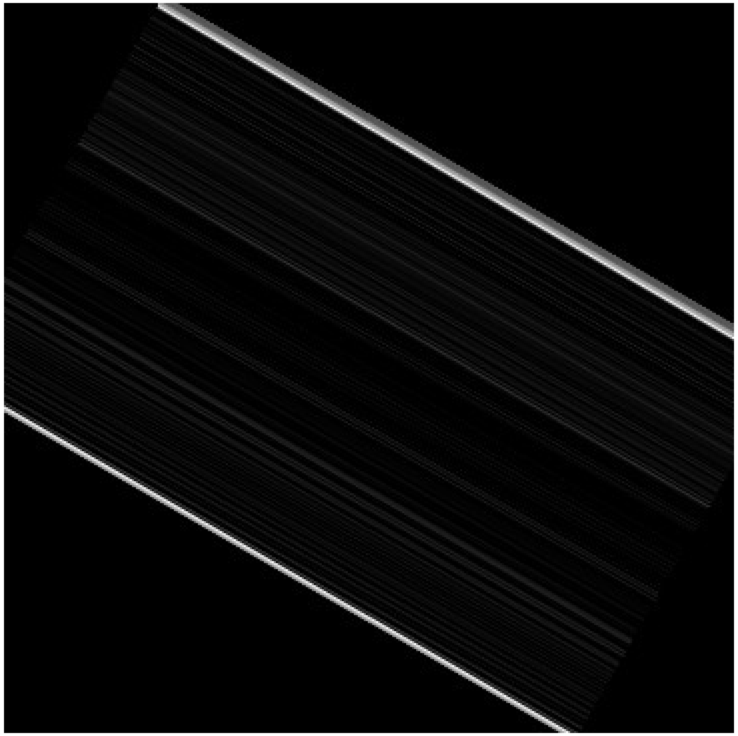


Projection image 40°

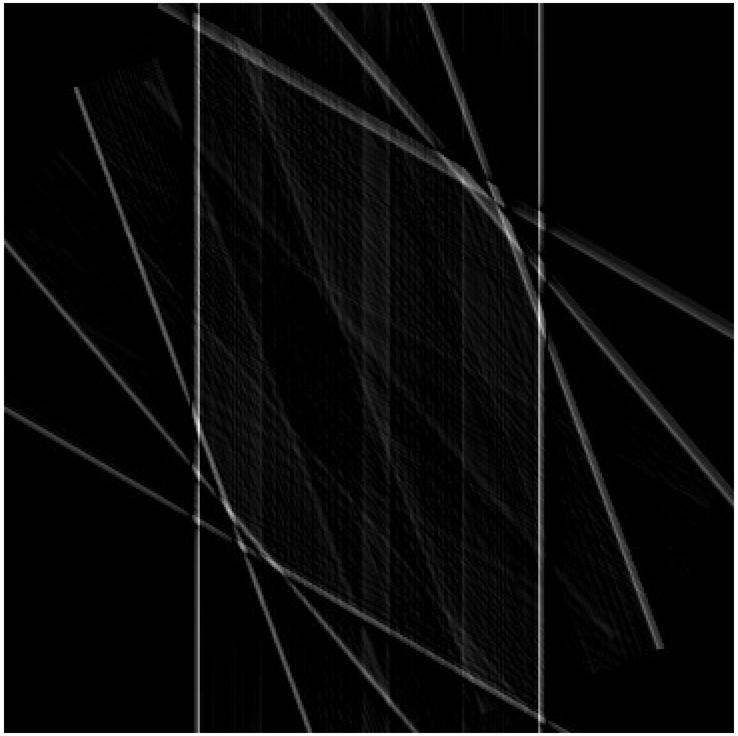


Backprojection (0°+ 20°+ 40°)

Naïve backprojection

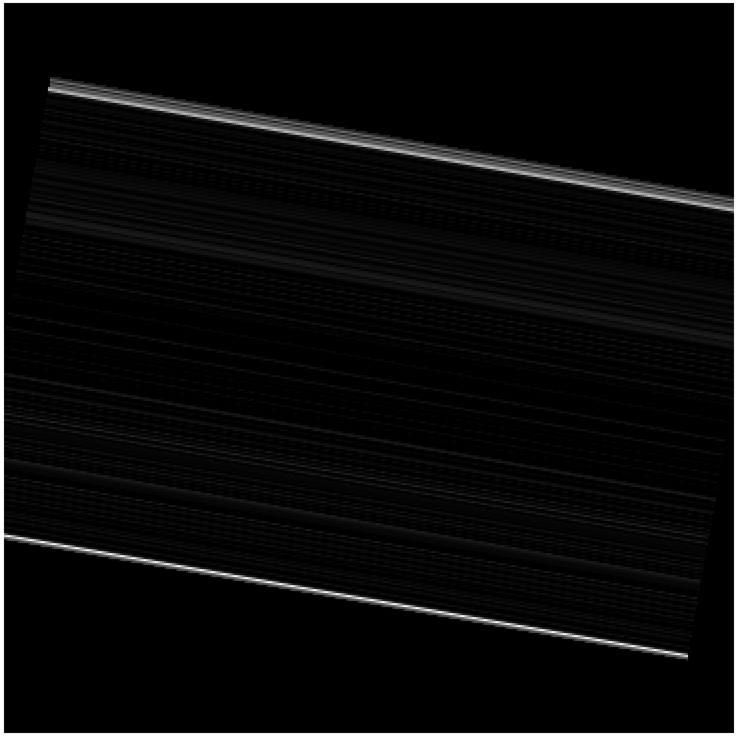


Projection image 60°

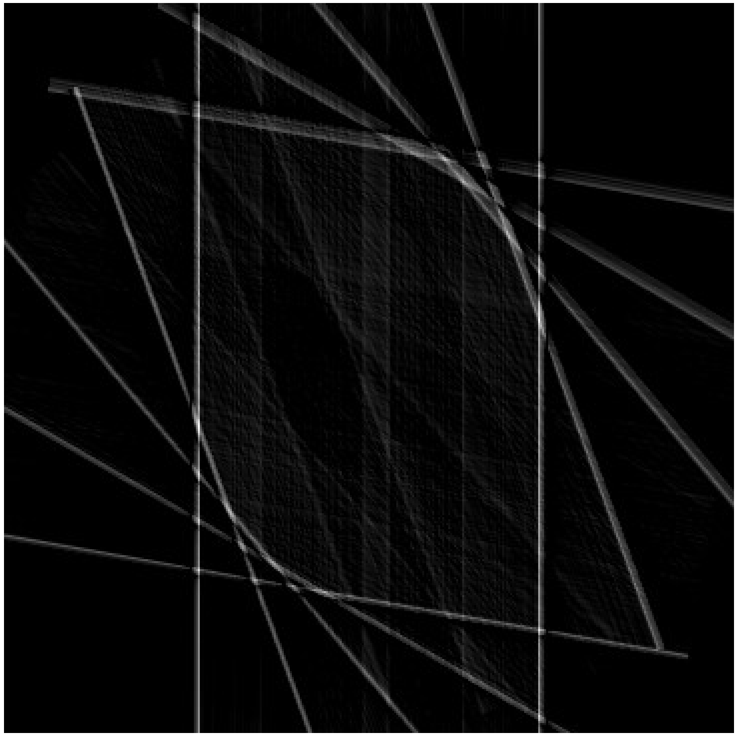


Backprojection (0° + … + 60°)

Naïve backprojection

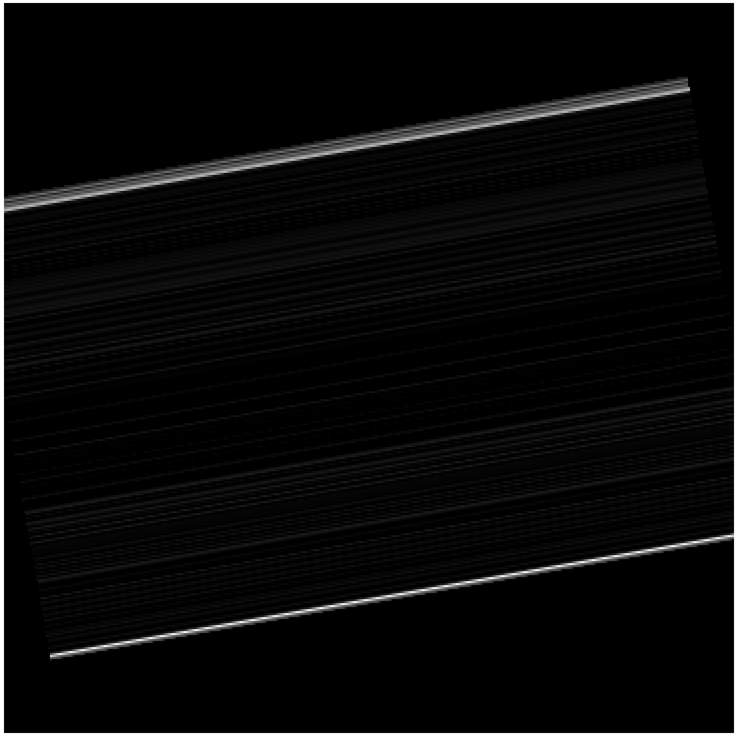


Projection image 80°

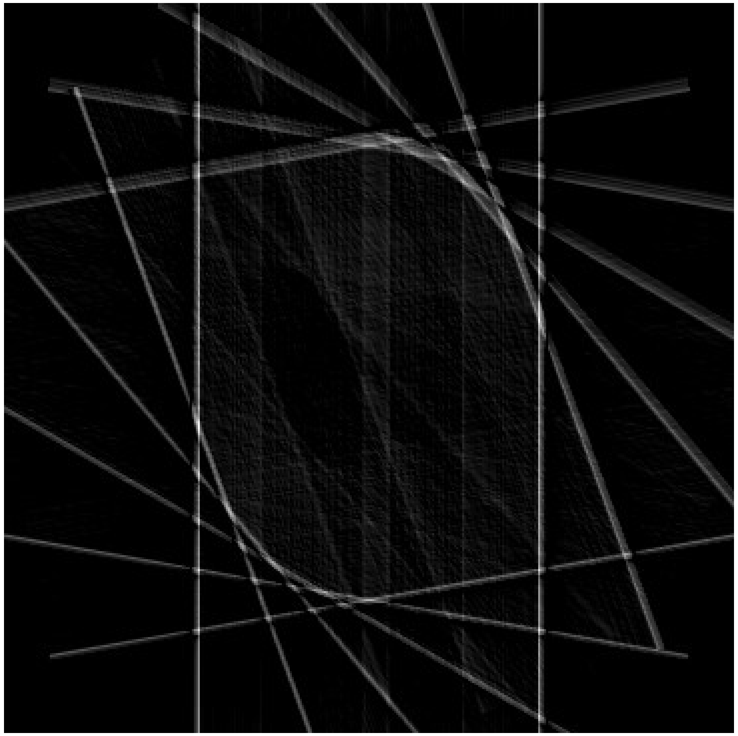


Backprojection (0° + … + 80°)

Naïve backprojection

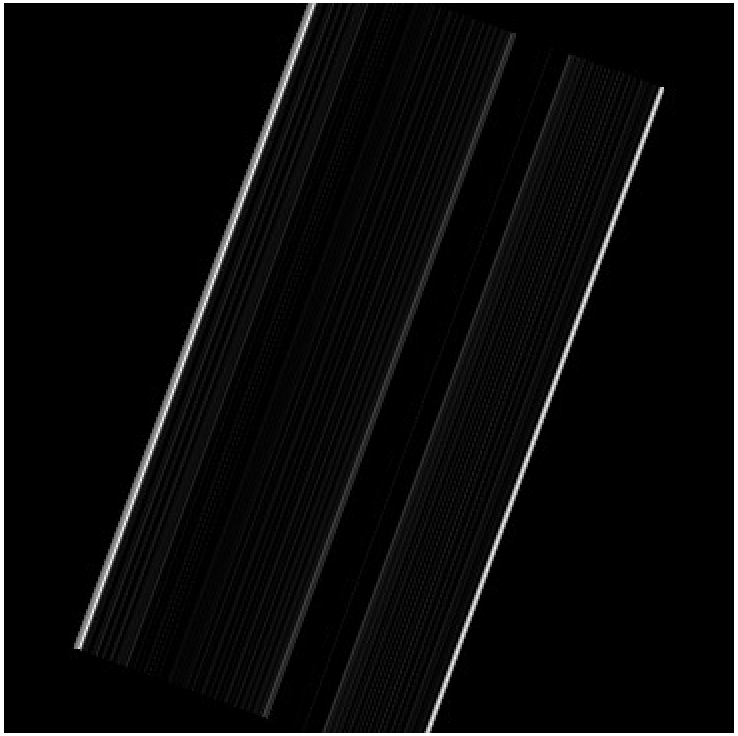


Projection image 100°

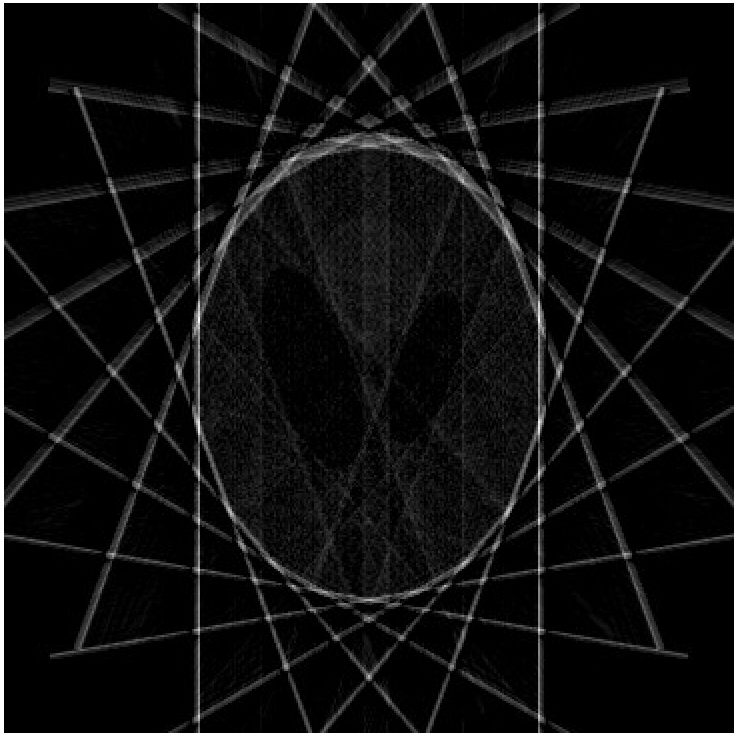


Backprojection (0° + … + 100°)

Naïve backprojection

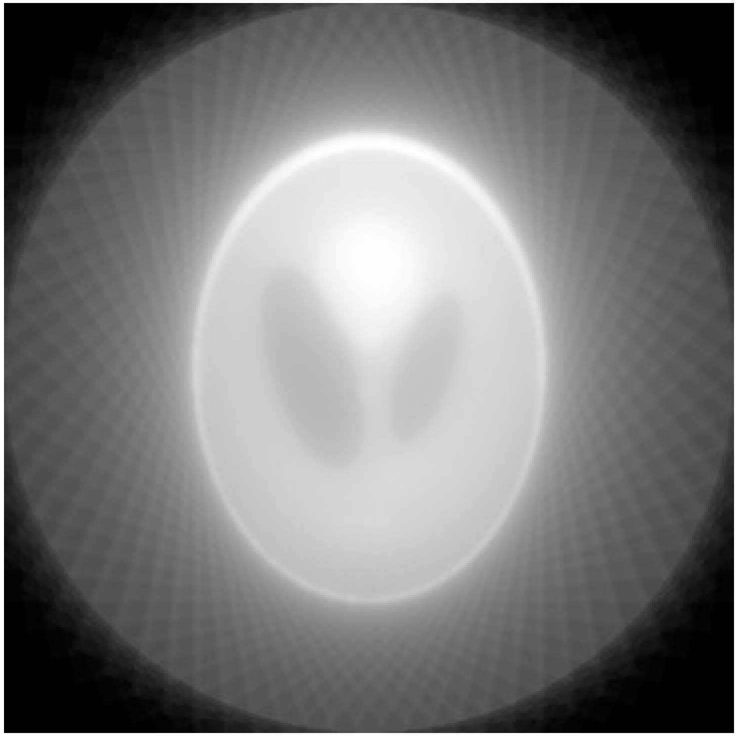


Projection image 160°



Backprojection (0° + … + 160°)

Comparison



Naïve backprojection

Filtered backprojection

HW Task 2.1

- Implement backprojection

  `myIradon(projim, thetas, f_type, f_d)`

  `projim` – image projections (sinogram)
  `thetas` – angles of the projections
  `f_type` – optional, type of filter
  `f_d` – optional, cutoff frequency of the filter f_type

HW Task 2.1

- Use designFilter(filter,len,d) function
  - provides a Fourier transform of a filter type *filter* with cutoff frequency *d* of length *len*
  - always returns filter of *even length* => zero padding of *odd length* signals when filtering in frequency domain
- Useful functions – meshgrid, interp2, fft, ifft, fftshift, conv, ...

HW Task 2.2

- Apply myIradon on sinogram from the file noisy_radon.mat with different settings:
  - naïve backprojection
  - filtered backprojection with ram-lak and hamming filters and each with cutoff frequencies 0.7 and 1.0
- Visualize filters in time and frequency domain, reconstructed images and report reconstruction error

HW Task 2.3

- Estimate reconstruction error for the different reconstruction settings

$$R = \sqrt{\frac{1}{m \cdot n} \frac{\sum_{x,y} \left(I_{rec}(x,y) - I_{ref}(x,y) \right)^2}{\sum_{x,y} I_{ref}(x,y)^2}}$$

HW Task 2.4 bonus

- Find the best filter and cutoff frequency settings with respect to the reconstruction error
- Use fewer reconstruction angles

Questions?