Tutorial: Introduction to tomographic reconstruction

1 X ray tomography

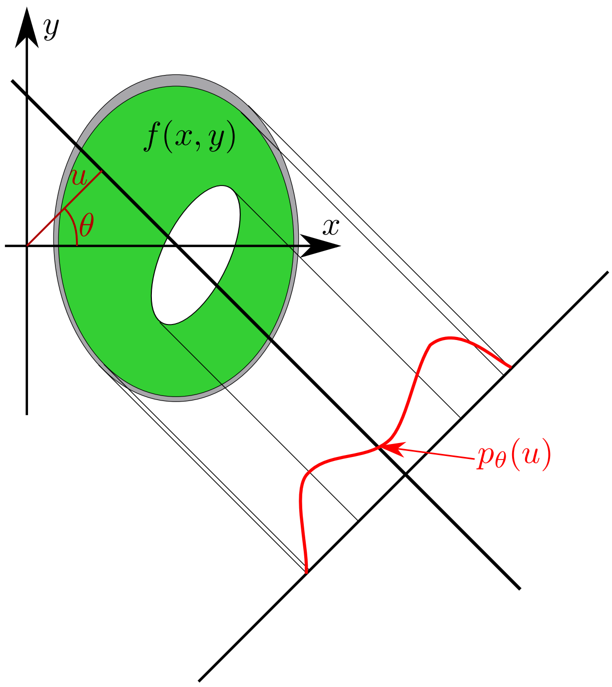


Figure 1: Radon transform notations.

An X-ray beam (considered as monochromatic) going through objects is attenuated, following a logarithmic law (f is the linear attenuation, v an elementary volume, I the intensity of the beam entering the volume dv and $I+dI$ the intensity exiting the volume):

$$\log \frac{I+dI}{I} = -f dv$$

By integrating this formula on a line D (direction of the beam), yields:

$$I = I_0 \exp\left(\int_D f(x,y)dv\right)$$

where $f(x,y)$ is the attenuation at point (x,y). This principle is used in scanners, scintigraphy, PET...

2 Acquisition simulation

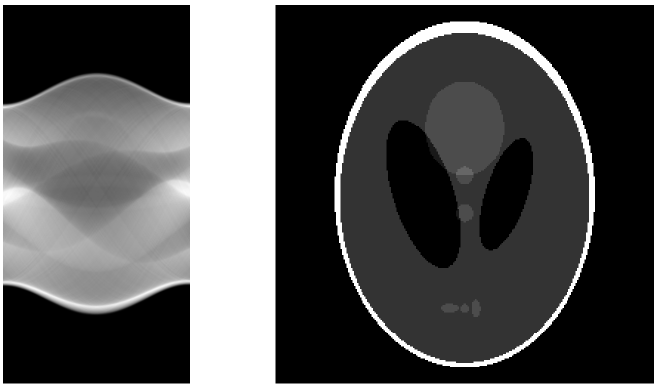
This first exercise will allow us to simulate the projections. The mathematical operator behing this is the Radon transform.

?

- Generate a "phantom" image of size 256×256. Display it.
- To simulate the attenation process, considere the sum of all gray levels of all pixels. Angles of projections will be between 0 and 180 degres with a unit step. The result is presented as a sinogram $p_\theta(u)$ (see Fig. 2), with u the length (in ordinates), and θ the projection angle (in abscissa).

Informations

Use the matlab function phantom.



(a) Sinogram. (b) Phantom image.

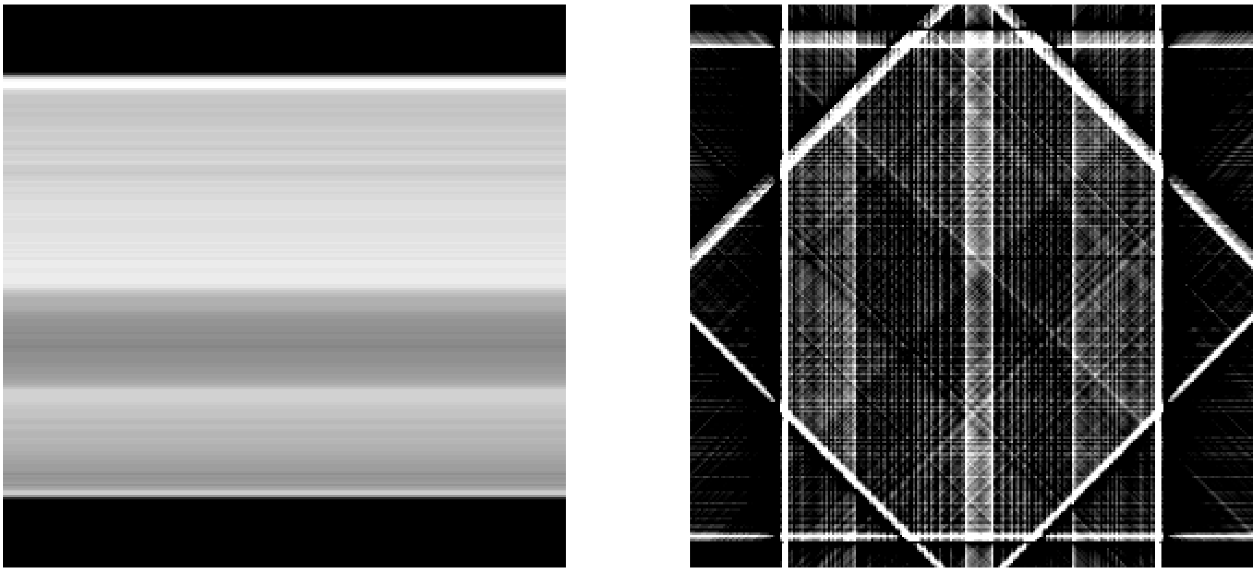
Figure 2: Simulated sinogram of the phantom image.

3 Backprojection algorithm

A non exact reconstruction is the backprojection operator B (it is not the inverse Radon transform). It attributes the value $p_\theta(u)$ to each point of the projection line that gave this attenation, and sum up all contributions from all projections (at the different angles).

$$B[p](x,y) = \int_0^\pi p_\theta(x\cos\theta + y\sin\theta)d\theta$$

A simple (but slow) method is described to compute the backprojection. $p(:,\theta)$ is the attenuation vector for a given angle θ (with a matlab syntax).



(a) Contribution of a line D before rotation.

(b) Reconstruction after projection every 45 degrees (4 projections).

Figure 3: Backprojection

- Use a command to replicate a matrix to obtain the contribution of each line D, like on Fig. 3a.
- Each contribution line D should be turned of a certain angle to be added to the overall contribution. This backprojection is really fuzzy compared to the original phantom image (see Fig. 3b).

Informations

The function repmat replicates a matrix a certain number of times.

4 Filtered backprojection

It can be shown that backprojection is in fact the convolution of f by a filter. A solution would be to make a deconvolution in the Fourier domain. The solution used here is the filtered backprojection.

Numerous filters can be used. Among them, the Ram-Lak filter RL is approximated by :

$$\begin{aligned} k \in [-B; B] \subset \mathbb{N}, RL(k) &= \frac{\pi}{4} \text{ if } k = 0 \\ &= 0 \text{ if } |k| \text{ is even} \\ &= \frac{-1}{\pi k^2} \text{ if } |k| \text{ is odd} \end{aligned}$$

- Write a function that generates this filter (see prototype). The parameter *width* is the number of points of the resulting vector.
- Modify the backprojection algorithm to filter (by convolution of the projection vector) each contribution line before the summation. Observe the improved result.

Informations

- Define a function with prototype: `function ramlak = RamLak(width)`.
- MATLAB has builtin functions `radon` and `iradon` for projection and filtered back-projection algorithms. Test them.