Computed Tomography

Contents

8.1. The physics of X-rays

X-rays are a form of electromagnetic radiation with higher energy and therefore shorter wavelength compared to visible or ultra-violet light. They were first described systematically by Wilhelm Röntgen in 1895. As their true nature was unclear to Röntgen at the beginning, he called them *X-radiation* referring to the letter X often used for the unknown in mathematical formulas. X-rays are produced whenever charged particles, typically electrons, hit a material with a sufficient kinetic energy. This can either happen if the electrons stimulate characteristic X-ray emission or if they are scattered and slowed down by the strong electric field near nuclei with a high proton number (a radiation still refereed to by the German name *Bremsstrahlung*). If X-rays transverse through material, they interact with it in three main ways:

- **Photoelectric absorption:** occurs if the incoming X-ray photon is absorbed by an atom of the material it hits. In this process, it transfers all its energy to one of the atom's electrons. This typically allows the electron to leave the atom as a free (negative) charge carrier and leaves the positively charged atom behind.
- **Compton scattering:** refers to the inelastic scattering of the X-ray photon by an electron of the outer atom shell. In this process, part of the energy of the X-ray photon is transferred to the electron, again allowing it to leave the atom. Due to momentum and energy conservation, the X-ray photon changes its direction and has a longer wavelength.
- **Rayleigh scattering:** refers to an elastic scattering of the X-ray photon, i.e., without loosing energy.

Depending on the energy spectrum of the X-rays and the material, these different processes contribute in different proportions to the overall interaction. As X-rays are ionizing radiation, their administration to biological tissue is harmful and needs to be well justified. An overview of where X-rays fit in the EM spectrum is shown in Fig. 8.1 Shortly after their discovery, images obtained by placing a target between an X-ray source and a photon detector such as shown Figure Fig. 8.2 were used for medical diagnosis and *projectional radiography* is still one of the most routinely applied examinations.

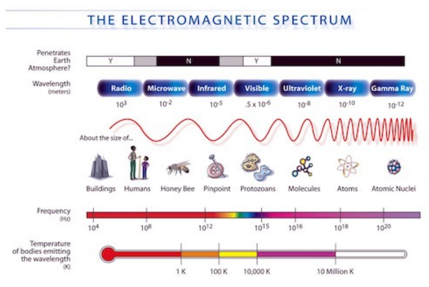


Fig. 8.1 Overview of the electromagnetic spectrum.

Fig. 8.2 X-ray photograph of the hand of Röntgen's wife.

8.2. The mathematics of X-rays

Now we will describe the attenuation of the intensity of a monochromatic ray of photons traveling through a material by the above processes mathematically. For this, we first assume that the photons are emitted at $x = 0$ with intensity I_0 and detected at $x = +\infty$ and ignore the y and z coordinate. The Beer-Lambert law states that the attenuation in a small interval of length δx is proportional to the intensity $I(x)$ of the ray, the attenuation coefficient $u(x)$ (which summarizes the probabilities for the physical processes mentioned above) in this interval and its length. This can be expressed a

$$I(x+\delta x) = I(x) - u(x)I(x)\delta x \iff \frac{I(x+\delta x)-I(x)}{\delta x} = -u(x)I(x)$$

In the limit $\delta x \to 0$, the expression on the right gives the ordinary differential equation (ODE)

$$\frac{d}{dx}I = -uI,\ I(0) = I_0,$$

which is solved by

$$I(x) = I_0 \exp\left(-\int_0^x u(x')dx'\right),$$

If we now assume that the X-rays travel along an arbitrary ray ℓ connecting source and sensor element, call the measured intensity I_1 and rearrange the above terms we can write

$$P_\ell = -\log\left(\frac{I_1}{I_0}\right) = \int_\ell u(x)dx. \tag{8.1}$$

So we see that in this formulation, we measure an integral of the unknown attenuation coefficient $u(x)$ along a line ℓ for each source-sensor pair.

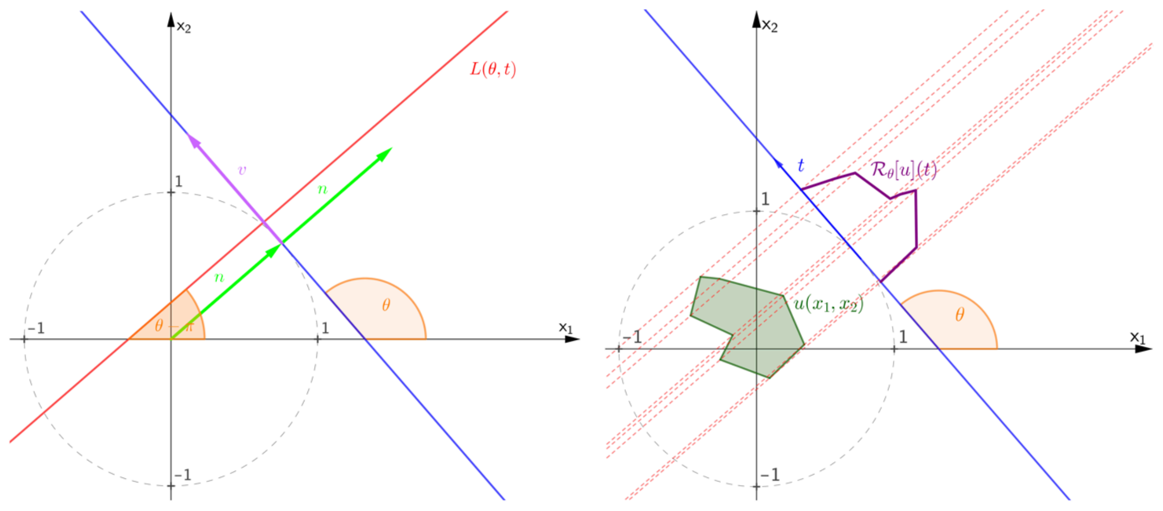


Fig. 8.3 Left: sketch of the auxiliary variables used to describe the two dimensional Radon transform. Right: Sketch of $R_\theta u(t)$, the projection of a function $u(x_1, x_2)$ onto a detector at angle θ

8.3. Computed tomography

In projectional radiography, only a particular two dimensional projection of a three dimensional object $u(x), x \in \mathbb{R}^3$ is obtained, namely its average X-ray attenuation along the projection rays. While this can reveal useful information about the interior properties of an object if patient, source and detector are arranged in a suitable way, it falls short of a complete reconstruction of u. The idea of computed tomography (CT) is to achieve this by combing projections from different directions by a computational algorithm. We start with the mathematical description of CT in two spatial dimensions, i.e., $x = (x_y, x_2)$. Let's assume that the support of $u(x_1, x_2)$ is constrained to the unit sphere, $\{x_1^2 + x_2^2 \leqslant 1\}$ and that we place a flat detector along the tangent of the unit sphere whose angle to the positive x_1-axis is given by $\theta \in [0, \pi)$ (cf. Fig. 8.3). Both the point of contact of this detection line with the unit sphere and its normal in this point are given as $n = (\cos(\theta - \frac{\pi}{2}), \sin(\theta - \frac{\pi}{2})) = (\sin(\theta), -\cos(\theta))$, while $v = (\cos(\theta), \sin(\theta))$ is a unit vector along the tangent (such that for $\theta = 0$, it points into positive x direction). Now, any point on the detector line can be described as $n + tv$, i.e., parameterized by $t \in \mathbb{R}$. A line ℓ that is orthogonal to the detector and crosses it at the point corresponding to $t \in [-1, 1]$ can be parameterized by $s \in \mathbb{R}$ as

$$\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} = sn + tv = s\begin{bmatrix} \sin(\theta) \\ -\cos(\theta) \end{bmatrix} + t\begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

We can now eliminate s by solving the equation for $x_2(s)$ for s and inserting this expression into the equation for $x_1(s)$ to obtain a handy implicit equation for all points lying on ℓ:

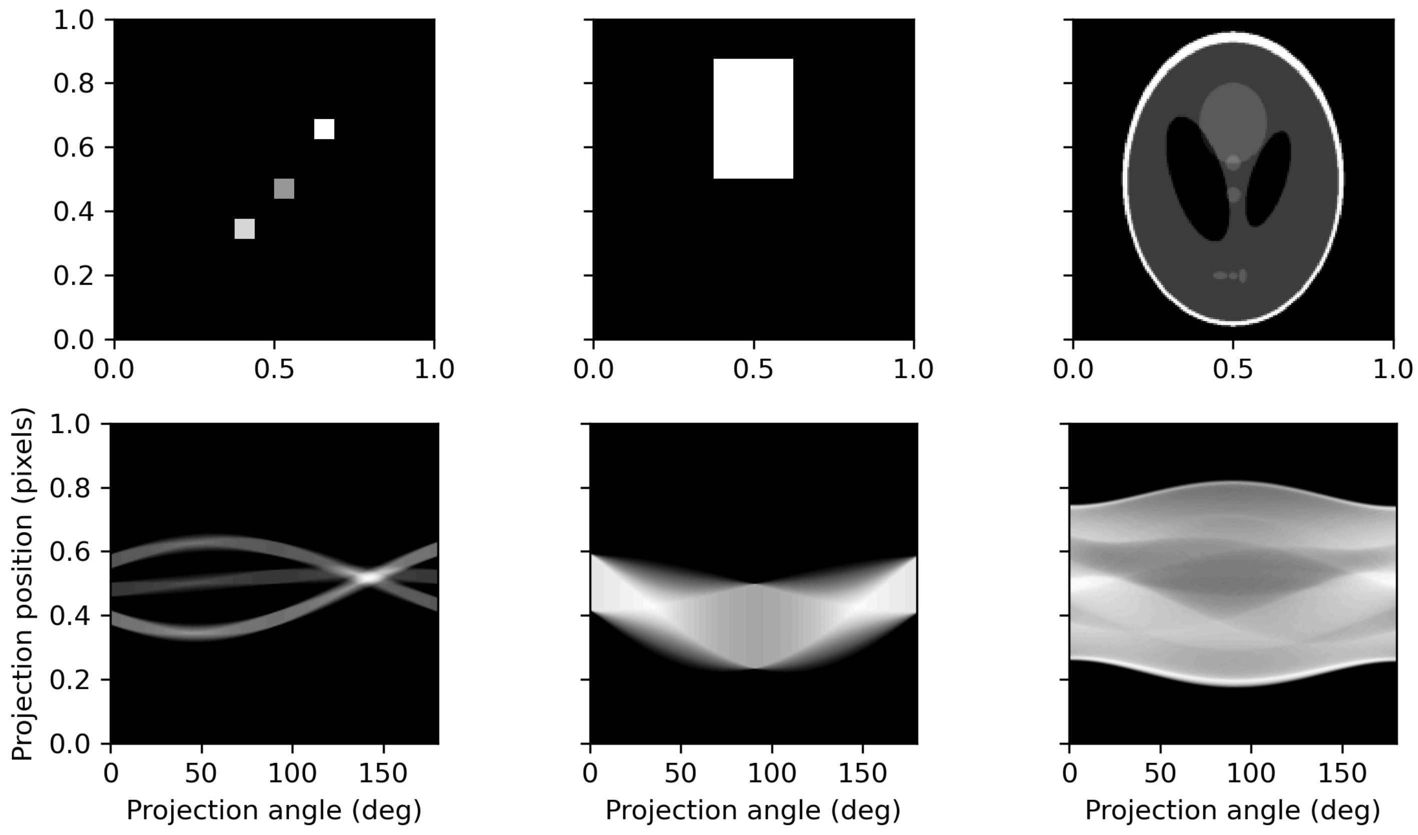
$$\ell(\theta, t) = \{x \in \mathbb{R} \,|\, x_1 \cos(\theta) + x_2 \sin(\theta) = t\}, \qquad (8.2)$$

where we stressed that ℓ is function of $\theta \in [0, \pi)$ and $t \in [-1, 1]$. In fact, all lines crossing the unit sphere can be parameterized in this way and inserting this parameterization into (8.1) yields a linear mapping R from $u(x_1, x_2)$ to a function $f(\theta, t)$:

$$f(\theta, t) = Ru(\theta, t) := \int_{\ell(\theta,t)} u(x)dx.$$

8.4. The Radon transform

The transform above describes all possible X-ray measurements of $u(x)$ and is called the *Radon transform* after the Austrian mathematician Johann Radon (1887-1956). Although Radon examined the problem from a purely mathematical perspective, namely as a problem of *integral geometry*, his work lay the mathematical foundations of CT. To get an intuition of how $Ru(\theta, t)$ looks like, consider $u = \delta_{(a,b)}$, i.e., a delta function located at $x_1 = a, x_2 = b$. If we call $R_\theta u(t) := Ru(\theta, t)$, i.e., the measurement on the detector for a fixed angle θ, $R_\theta u(t)$ will only be zero except for $t = a\cos(\theta) + b\sin(\theta)$ (cf. (8.2)), where it is one. That means that $Ru(\theta, t)$ is an indicator function on the set $\{(\theta, t) \,|\, t = a\cos(\theta) + b\sin(\theta)\}$, i.e., on a set where t can be described by the linear combination of a sine and a cosine function of θ. Therefore $f(\theta, t)$ typically looks like a superposition of sine and cosine waves and is therefore called *sinogram*. Some examples of functions and their sinograms are shown below.



Now that we can describe all possible X-ray measurements $f(\theta, t)$ of an image $u(x_1, x_2)$ via the Radon transform R, the interesting question is whether we can also recover u from f, i.e., whether R is invertible. This question is answered by the *Fourier slice theorem*, which relates the one dimensional Fourier transforms of $R_\theta u(t)$ to the two dimensional Fourier transform of u:

Theorem: Fourier Slice Theorem

The one-dimensional Fourier transform of $f(\theta, \cdot) = R_\theta u$ is related to the two-dimensional Fourier transform of u as

$$\widehat{f}(\theta, \omega) = \widehat{u}(\omega v), \qquad (8.3)$$

with $v = (\cos\theta, \sin\theta)$.

This implies that the projection data for a particular angle θ fully determines the radial slice of the Fourier transform of u with the same angle. Thereby, the complete Fourier transform of u can be computed from $f(\theta, t)$ and as the Fourier transform is invertible, we can recover u from f.

Proof

8.5. Fourier reconstruction

The Fourier slice theorem suggests a simple strategy to invert the Radon transform: First, we compute the one dimensional Fourier transform of each angular projection $R_\theta u$ and arrange them in two dimensional Fourier space to obtain the Fourier transform of u and then we compute the inverse Fourier transform to obtain u itself. The main problem with this strategy is that in practice, the amount of measurements is finite. If we assume that we sample $R_\theta u(t)$ on a regular t-grid, we also sample the radial slice $\hat{u}(\omega v)$ on a regular ω-grid. Due to the radial arrangement of the slices, we sample $\hat{u}$ on circles and not on a regular grid, which would be required to use efficient inverse Fourier transform algorithms such as the *fast Fourier transform* (*FFT*). As illustrated in Fig. 8.4, this means that the density of the frequency samples decreases towards the high frequencies, which contain information about the small scale image features.

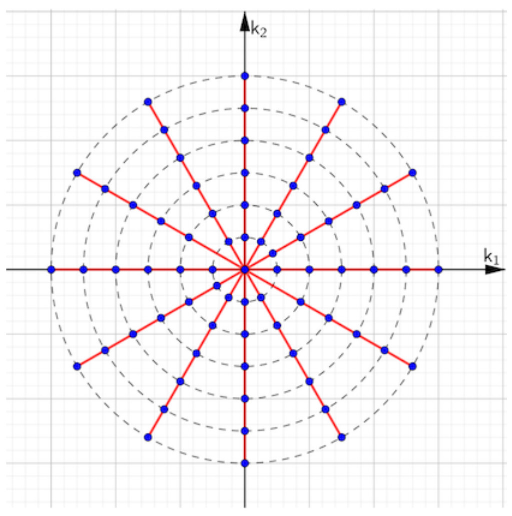


Fig. 8.4 Sampling in Fourier space: The Fourier slice theorem (8.3) states that every angular projection $R_\theta[u](t)$ contains the full information about $\hat{u}(k_1, k_2)$ along one radial slice (red lines). A finite, regular sampling of t translates into a finite, regular sampling along these lines (blue dots). It becomes apparent that lower frequencies ($\sqrt{k_1^2 + k_2^2}$ small) are sampled more densely than high frequencies.

8.6. Filtered back projection

The Fourier slice theorem is the basis the most widely used CT reconstruction technique, the *filtered back projection* (*FBP*). As the name suggests, FBP consists of two steps, a filtering step and a back projection step. We start with the back projection, which is a simple operation to map a sinogram $f(\theta, t)$ back into the image space: For every angle θ, we smear the one dimensional function $f_\theta(t) := f(\theta, t)$ (which corresponds to the projection $R_\theta u$) over the image by adding the value $f_\theta(t)$ to each point on the ray $\ell(\theta, t)$, namely to all (x_1, x_2) that satisfy $t = x_1 \cos\theta + x_2 \sin\theta$:

$$u_{BP}(x_1, x_2) = \int_0^\pi f(\theta, x_1 \cos\theta + x_2 \sin\theta) d\theta. \tag{8.4}$$

It turns out that the back projection operator is the adjoint R^* of the Radon transform. Unfortunately, it is not the inverse R^{-1}. Rather, one can show that

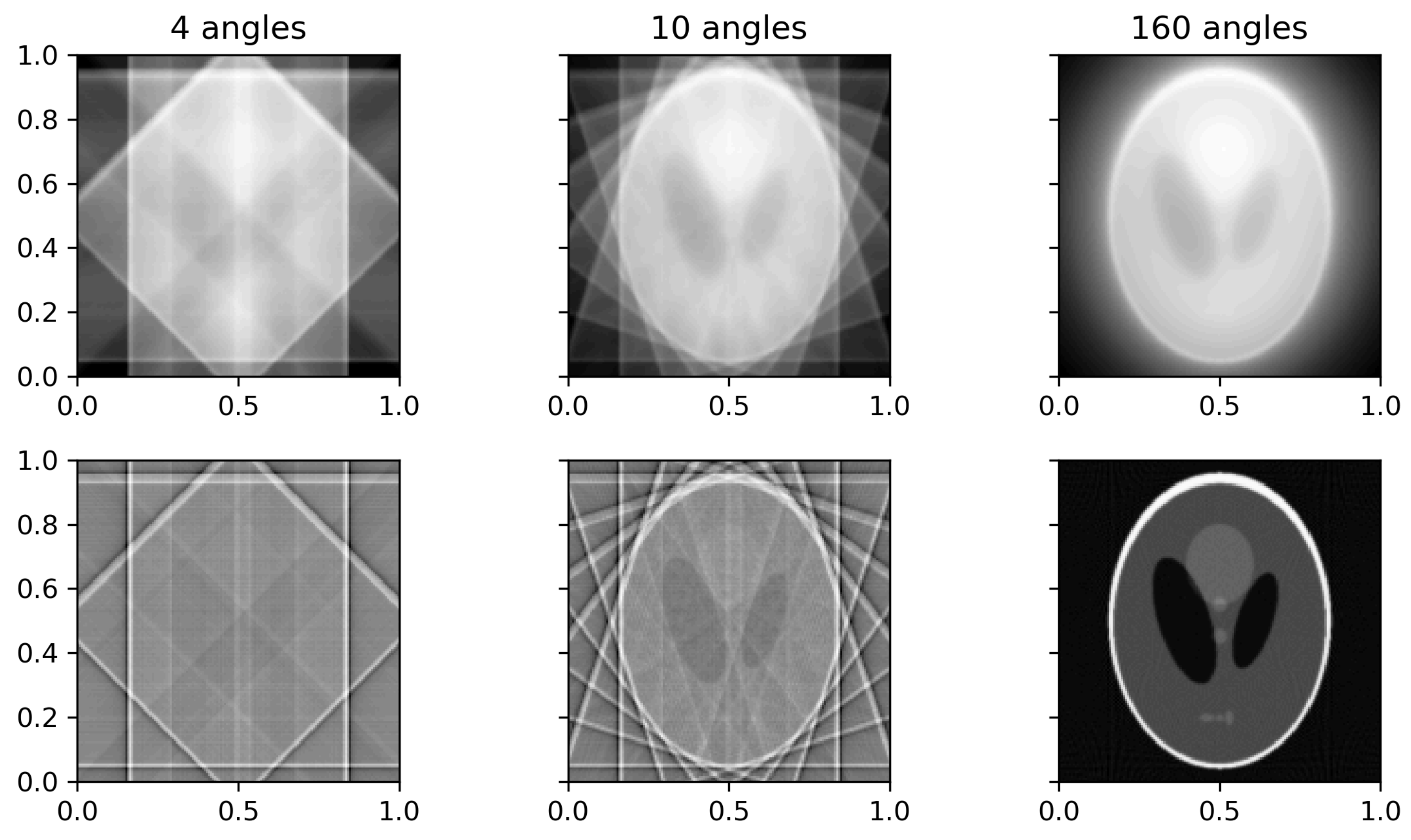
$$R^* R u(x) = \int_{\mathbb{R}^2} \frac{u(x)}{\|x - y\|_2} dy,$$

i.e., it only recovers $u(x)$ convoluted with a smoothing kernel, as will be illustrated below. The reason for this is the non-uniform sampling of $\hat{u}(k)$, the Fourier transform of $u(x)$ (cf. Fig. 8.4): In proportion, too many low frequencies are projected back, and low frequencies represent the smooth features of the image. A reasonable idea to improve upon this result is therefore to introduce weights that counter-act this imbalance in favour of the high-frequencies before the back projecting step. This corresponds to a *high-pass filtering* of the sinogram data $f(\theta, t)$ in t-direction, i.e., we convolute it with a suitable filter function $h(t)$. It turns out that the correct filter we need is a *ramp*-filter, whose Fourier transform is given by $\hat{h}(\omega) = |\omega|$. Taken together, the filtered sinogram q is given as

$$q(\theta, t) = Hf(\theta, t) := F_t^{-1}\left(\hat{h}(\omega) \cdot \widehat{f}(\theta, \omega)\right).$$

Below, we illustrate that if the filtered sinogram is now inserted into the back projection (8.4), we converge to the true solution if the number of angles increases, and one can indeed show that $R^* H R = I$.

```
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/2696548906.py:18:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  BP1 = iradon(sinogram1, theta=theta1, filter=None, circle = False)
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/2696548906.py:19:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  BP2 = iradon(sinogram2, theta=theta2, filter=None, circle = False)
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/2696548906.py:20:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  BP3 = iradon(sinogram3, theta=theta3, filter=None, circle = False)
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/2696548906.py:22:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  FBP1 = iradon(sinogram1, theta=theta1, filter='ramp', circle = False)
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/2696548906.py:23:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  FBP2 = iradon(sinogram2, theta=theta2, filter='ramp', circle = False)
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/2696548906.py:24:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  FBP3 = iradon(sinogram3, theta=theta3, filter='ramp', circle = False)
```

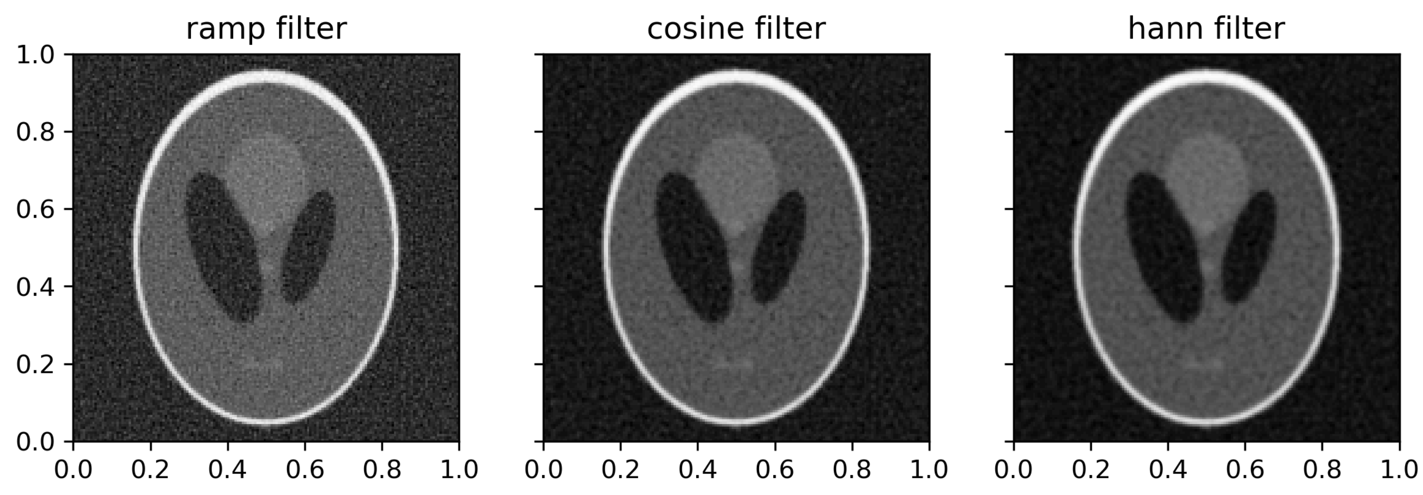


Alternatively, the filter may be applied *after* back projection. In this case the filter multiplies the two-dimensional Fourier transform of the back projected image by $\|\xi\|_2$.

8.7. Ill-posedness and regularisation

The previous section showed that in order to reconstruct the image from the sinogram we have to multiply either by $|\omega|$ or by $\|\xi\|$ in the Fourier domain of the sinogram or the image. This suggests that the inverse of the Radon transform is unbounded. To regularise the FBP method, various modifications to the filters have been proposed, as illustrated in the example below.

```
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/4062371152.py:16:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  FBP_ramp = iradon(f_delta, theta=theta, filter='ramp', circle = False)
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/4062371152.py:17:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  FBP_cosine = iradon(f_delta, theta=theta, filter='cosine', circle = False)
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/4062371152.py:18:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  FBP_hann = iradon(f_delta, theta=theta, filter='hann', circle = False)
```



8.8. Algebraic reconstruction

As we’ve seen in the previous sections, the Fourier Slice Theorem only applies when sufficient angular samples are available. An alternative to FBP are so-called *algebraic reconstruction techniques* which discretize the Radon transform using numerical quadrature and sets up a system of equations

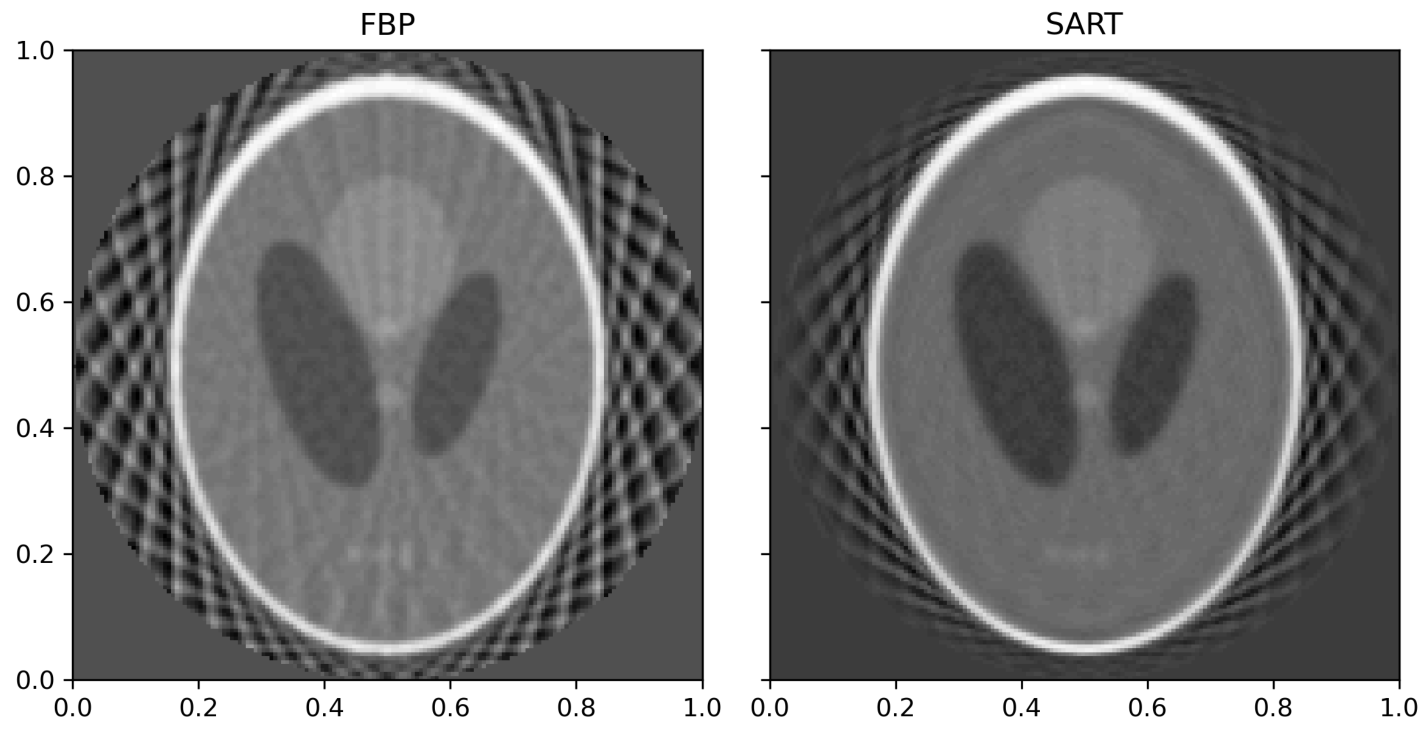
$$f_i = \sum_i r_{ij} u_j,$$

where $f_i = f(\theta_i, t_i)$, $u_j = u(x_j)$ denotes the intensity of u in the j^{th} pixel and r_{ij} indicates the contribution of pixel j to ray i. This typically leads to an underdetermined system of equations $Ru = f^\delta$ which we can attempt to solve using an iterative method. Popular method in this application is *SART* [Andersen and Kak, 1984]:

$$u^{(k+1)} = u^{(k)} + D_1 R^* D_2 \left(f - Ru^{(k)}\right),$$

where D_1 and D_2 are diagonal matrices containing the row and column sums of R respectively. An example is shown below.

```
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/57260500.py:17:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  u_fbp = iradon(f_delta, theta=theta, filter='hann')
```



8.9. Beyond 2D

Up to now, both the function $u(x_1, x_2)$ and its sinogram $f(\theta, t)$ where functions of two variables and we saw that this was enough to reconstruct u from f. In three spatial dimensions, we need four variables to parameterise all lines going through the unit sphere (two for the orientation and two for the spatial offset), but we do not need to measure all the corresponding line integrals. From the 2D results, we know that all directions lying in a plane are sufficient to fully determine the function in that plane. This is the idea behind the classical CT scanning strategy: The patient is lying on a bed that is traversed through the scanner step-by-step and all projections in a plane orthogonal to this direction are acquired, cf. Fig. 1.2. Then, the whole three dimensional attenuation map can be reconstructed slice-by-slice, which also explains the word *tomography*, which is derived from the ancient Greek *tomos*, which means *slice* or *section*.

8.10. Exercises

8.10.1. The Radon transform and derivatives

Given the Radon transform

$$Ru(\theta,t) = \int_{\mathbb{R}} u(t\cos\theta + s\sin\theta, t\sin\theta - s\cos\theta)\mathrm{d}s,$$

Show that (for sufficiently regular functions)

$$R\nabla^2 u(\theta,t) = \partial_t^2 Ru(\theta,t).$$

8.10.2. SART

Given a discretisation of the Radon transform $\in \mathbb{R}^{m\times n}$, we apply the SART method to reconstruct the image.

- Show that a stationary point of the SART iteration solves

$$\min_u \|Ru - f\|_W^2,$$

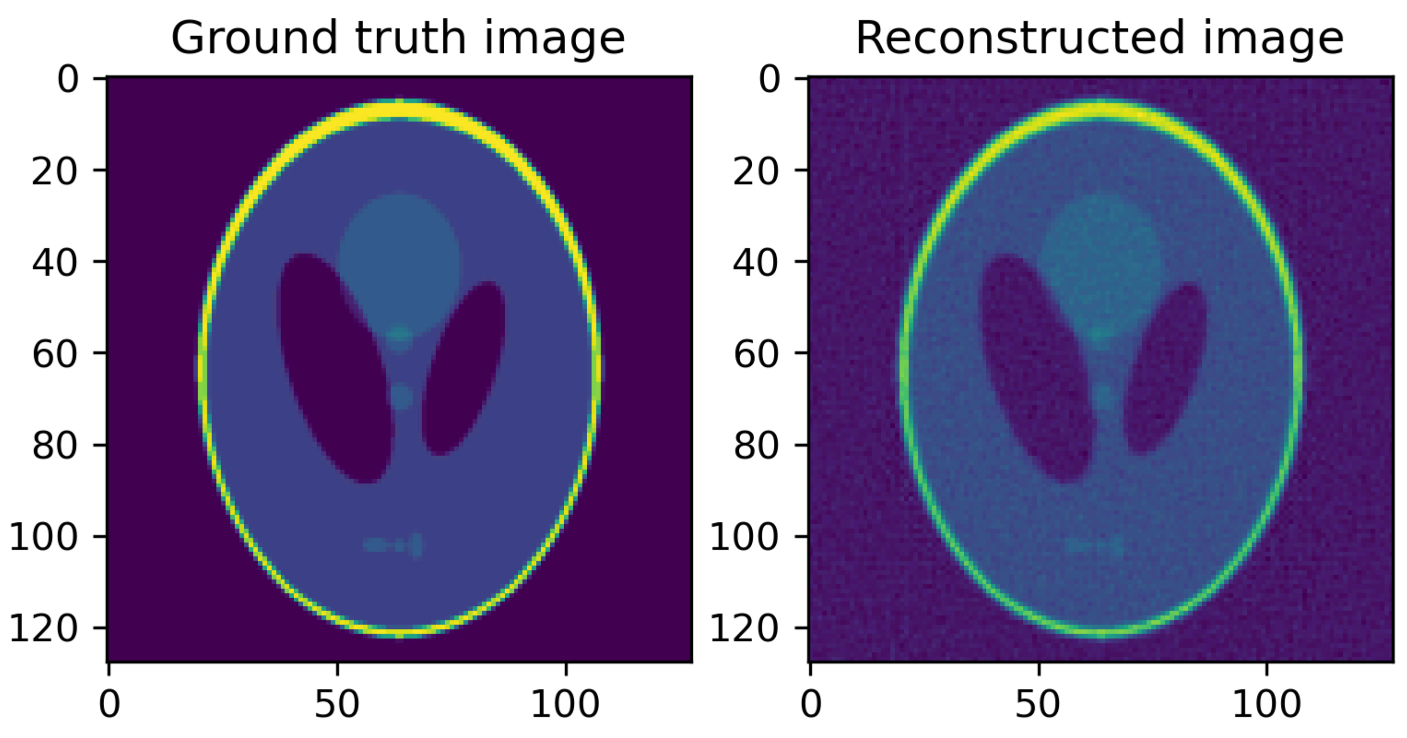
with W a diagonal elements with elements $w_{ii} = \sum_{j=1}^n r_{ij}$. Here $\|f\|_W = \sqrt{\langle f, Wf\rangle}$. You may assume that R has positive row and column sums.

8.10.3. FBP in practice

One implementation of the Radon transform and the FBP can be found in the `scikit-image` toolbox. An example of how to load a simple image, compute its Radon transform to simulate sinogram data and how to run a FBP to obtain a reconstruction from this data is shown below.

```python
# import libraries
import numpy as np
import matplotlib.pyplot as plt
from skimage import data
from skimage.transform import radon,iradon,resize
from skimage.util import random_noise

# settings
n = 128
theta = np.linspace(0., 180.,n, endpoint=False)
sigma = 1e-1

# ground truth image
image = resize(data.shepp_logan_phantom(),(n,n))

# transform and add noise
sinogram = radon(image, theta=theta, circle = False)
sinogram_noisy = random_noise(sinogram,mode='gaussian',var=sigma,clip=False)

# reconstruction, type help(iradon) for more options regarding the filter
image_reconstructed = iradon(sinogram_noisy, theta=theta, filter='ramp', circle = False)

# plot
fig, ax = plt.subplots(1,2)

ax[0].imshow(image)
ax[0].set_title('Ground truth image')

ax[1].imshow(image_reconstructed)
ax[1].set_title('Reconstructed image')

plt.show()
```

```
/var/folders/g2/z5xkdjy513j76nkltmmxlkp00000gn/T/ipykernel_24854/593708864.py:21:
FutureWarning: 'filter' is a deprecated argument name for `iradon`. It will be removed in
version 0.19. Please use 'filter_name' instead.
  image_reconstructed = iradon(sinogram_noisy, theta=theta, filter='ramp', circle = False)
```



- Now we examine what happens if we add noise to the sinogram data. Let $f = Ku$ be the clean sinogram data arranged as a $m \times 1$ vector and generate noisy data by setting $f^\delta = f + \sigma\varepsilon$, where ε is a $m \times 1$ vector of standard normal distributed random variables and σ determines the noise level. Examine how the FBP reacts to different noise levels and how changing the frequency filter affects the results.
- In many applications, the range of available projection angles is restricted $\theta \in [\theta_{min}, \theta_{max}]$, $\theta_{min} > 0$, $\theta_{max} < \pi$ or the angular resolution is very coarse. Examine the effects of these restrictions on FBP reconstructions. In limited angle tomography, which parts of the image are lost?

8.11. Assignments

8.11.1. The Radon transform as a compact operator

Define the Radon transform $R : L^2(\Omega) \rightarrow L^2([0,\pi) \times [-1,1])$ on the unit circle $\Omega = \{x \in \mathbb{R}^2 | \|x\|_2 \leq 1\}$ by:

$$Ru(\theta,t) = \int_{-\sqrt{1-t^2}}^{\sqrt{1-t^2}} u(t\xi_\theta + s\eta_\theta)\mathrm{d}s,$$

with $\xi_\theta = (\cos\theta, \sin\theta)$ and $\eta_\theta = (\sin\theta, -\cos\theta)$.

- Introduce

$$\|f\|_{\mathcal{F}}^2 = \int_0^\pi \int_{-1}^1 (1-t)^{-1/2}|f(\theta,t)|^2\mathrm{d}t,$$

and

$$\|u\|_{\mathcal{U}}^2 = \int_\Omega |u(x)|^2\mathrm{d}x.$$

Show that $\|Ru\|_{\mathcal{F}} \leq 2\sqrt{\pi}\|u\|_{\mathcal{U}}$.

- Derive the adjoint of R (with respect to the weighted inner product defined above).
- Show that the operator RR^* has eigenfunctions

$$u_{k,l}(\theta,t) = (1-t)^{-1/2}U_k(t)e^{-\imath(k-2l)\theta},$$

with eigenvalues

$$\sigma_k^2 = \frac{2\pi}{k+1}.$$

Here, U_k denote the Chebyshev polynomials of the second kind.

- Now compute the eigenfunctions of R^*R.

8.11.2. Discretisation of the Radon transform

Discretise the Radon transform by approximating $u(x)$ as a piece-wise bi-linear function on a regular grid on the domain $[-1,1]^n$:

$$u(x,y) = \sum_{i=0}^n\sum_{j=0}^n u_{ij}\phi\left(\frac{x-x_i}{h}\right)\phi_h\left(\frac{y-y_i}{h}\right),$$

with $h = 2/n$, $x_i = -1 + i\cdot h$, $y_j = -1 + j\cdot h$ and $\phi(x) = \max(1-|x|,0)$.