- We measure the Radon transform of f,

$$\Re f(s,\vec{\theta}) = \log\left(\frac{I_0}{I_d}\right),$$

where I_0 is the intensity of the beam at the source, and I_d is the intensity of the beam at the detector.

- Reconstruct f from the Radon inversion formula (2.43).

For the filtered back-projection algorithm, we regard the radial integral in the Radon Inversion Formula as a filter. We denote the output of the filter by $\mathcal{G}\Re f(t,\vec{\theta})$, where

$$\mathcal{G}\Re f(t,\vec{\theta}) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \widetilde{\Re f}(r,\vec{\theta})e^{irt}|r|dr.$$

Then, with $t = x\cdot\vec{\theta}$,

$$f(x) = \frac{1}{2\pi}\int_0^{\pi} \mathcal{G}\Re f(x\cdot\vec{\theta},\vec{\theta})d\theta.$$

Note that one sees from this formula that low-frequency components are suppressed by $|r|$ and high-frequency components are amplified. Let's look at the filter a little more carefully. Recall that the Fourier transform of $g'(t)$ is

$$\mathcal{F}(\partial_t g)(\xi) = i\xi\hat{g}(\xi).$$

Thus if we had r instead of $|r|$ in the Radon inversion formula, we would have had the

$$\text{“inversion formula”} = \frac{1}{2\pi i}\int_0^{\pi} \partial_r \Re f(r,\theta)d\theta.$$

If f is real-valued, this quantity is purely imaginary! Thus, the $|r|$ is very important!

The MATLAB function `iradon.m` in the Image Processing Toolbox implements filtered back-projection. In the subsequent sections, we will be comparing the results of filtered back-projection implemented with `iradon.m` to other inversion techniques.

2.3.4 Discrete tomographic data

We model practical tomographic X-ray data by a bounded set $\Omega \subset \mathbb{R}^2$, a nonnegative attenuation coefficient f supported in $\overline{\Omega}$, and some finite collection $\{L_j\}_{j=1}^k$ of lines $L_j \subset \mathbb{R}^2$ intersecting Ω.

As a first example, we will use the following data set. It is an example of *parallel-beam geometry* illustrated in Figure 2.15. The angular variable is sampled with equidistant steps over the half circle:

$$\theta_j = \theta_1 + \left(\frac{j-1}{J}\right)\pi, \qquad 1 \le j \le J, \tag{2.44}$$

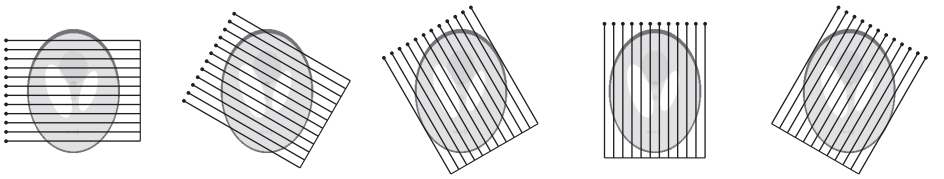


Figure 2.15. Parallel beam X-ray measurement geometry. Here $J = 5$ and $N = 11$. Black dots show the locations of the X-ray source at different times of measurement. The thick line represents the detector measuring the intensity of the X-rays after passing through the target. High attenuation is shown here as darker shades of gray and low attenuation as lighter shades.

where $\theta_1 \in \mathbb{R}$ is an appropriate constant, a reference angle. The linear parameter s is also sampled uniformly over a suitable interval:

$$s_\nu = -S + 2\left(\frac{\nu - 1}{N}\right) S, \qquad 1 \le \nu \le N, \tag{2.45}$$

where $S > 0$.

Defining $k = JN$, the measurement (1.1) then takes the form

$$\mathbf{m} = \mathcal{A}f + \varepsilon = \begin{bmatrix} \int_{L_1} f(x_1, x_2)\, ds_1 \\ \vdots \\ \int_{L_k} f(x_1, x_2)\, ds_k \end{bmatrix} + \varepsilon, \tag{2.46}$$

where ds_j denotes the one-dimensional Lebesgue measure along the line L_j. Each integral in (2.46) can be understood as a suitable rotation and scaling of formula (2.39).

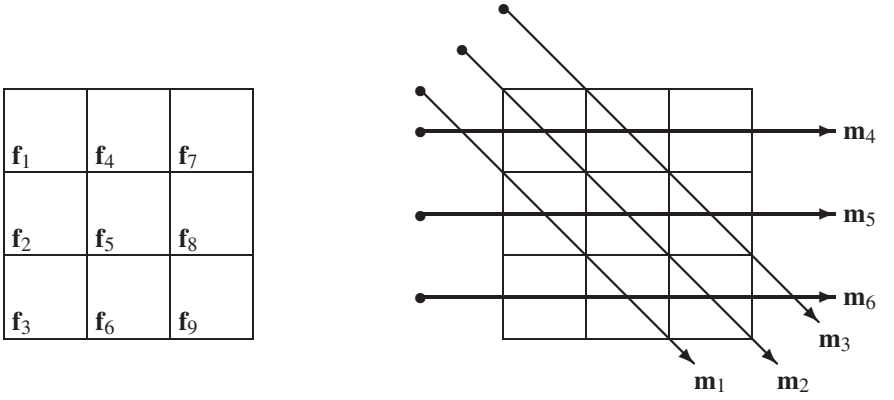
For the computational solution we need to build a finite-dimensional measurement model of the form (1.3). We discretize the tomographic problem by dividing the unknown area into n pixels and assume that attenuation values are constant within each pixel. We number the pixels from 1 to n and call the corresponding attenuation values $\mathbf{f}_j \ge 0$ for $j = 1, \ldots, n$.

The measurement $\mathbf{m}_i$ of the line integral of f over line L_i is approximated by

$$\mathbf{m}_i = \int_{L_i} f(x_1, x_2)\, ds \approx \sum_{j=1}^{n} a_{ij} \mathbf{f}_j, \tag{2.47}$$

where a_{ij} is the distance that L_i travels in the jth pixel. Note that only pixels that intersect the beam L_i are included in this sum. Further, if we have k measurements in the vector $\mathbf{m} \in \mathbb{R}^k$, then (2.47) yields a matrix equation $\mathbf{m} = A\mathbf{f}$, where the matrix is defined by $A = (a_{ij})$.

Consider the following discretization and measurements, where $J = 2$, $k = 6$, $N = 3$ and the total number of pixels is $N^2 = 9$:



Here we have divided the square-shaped domain $\Omega \subset \mathbb{R}^2$ into 9 pixels, denoted by thin lines. The length of the side of each pixel is 1. Inside the pixels there is a constant value $\mathbf{f}_j$ of attenuation. The six arrows are X-rays used for probing the inner structure of Ω. Measurement data is the vector $\mathbf{m} = [\mathbf{m}_1, \ldots, \mathbf{m}_6]^T$ modeled by (2.47). The resulting measurement model is

$$\begin{bmatrix} 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 \\ \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \end{bmatrix}. \tag{2.48}$$

The model (2.48) is low-dimensional and simple. However, it already demonstrates one feature typical for inverse problems: nonuniqueness of solution. Namely, as can be seen in Exercise 2.3.5, there are several targets that produce exactly the same data. Thus the inverse problem cannot be uniquely solved using the measurement information alone.

Let us build a more realistic (higher-dimensional) data simulation model. We work with the so-called Shepp–Logan phantom, which is a piecewise constant model of a cross-section of a human head. The phantom is defined using ellipses and can be realized at any desired discrete resolution. See Figure 2.16 for pictures of the Shepp–Logan phantom at discretizations with $16 \times 16 = 256$ pixels, $50 \times 50 = 2500$ pixels, and $512 \times 512 = 262144$ pixels.

Let us construct the measurement matrix A corresponding to the low-resolution case with 16×16 pixels and projection directions specified by taking $J = 16$ in (2.44). In this case the size of A is not too large and we can show a picture of the nonzero elements of A for observing its structure.

We use MATLAB's command `radon.m` to simulate parallel-beam X-ray projection data from 16×16 pixel images with zero entries except one pixel with value 1. The pixel value 1 is first located in pixel 1 in the numeration shown in Figure 2.17(b), then in pixel 2, and so on. This way, column by column, we construct a measurement matrix A for a computational tomography model of the form (1.3).

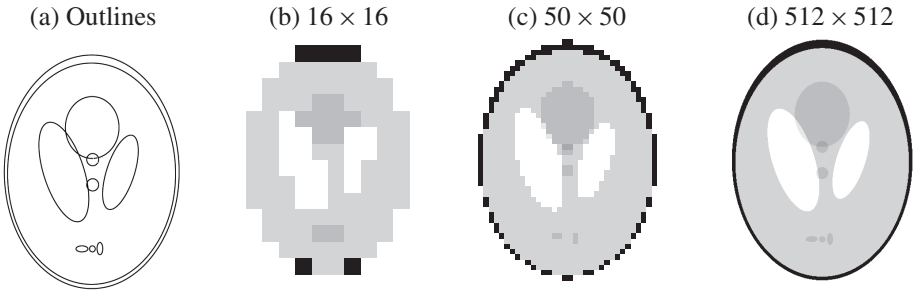


Figure 2.16. (a) Schematic illustration of the Shepp–Logan phantom. Areas of different attenuation values are bounded by various ellipses. (b)–(d) Plots of grayscale images of the Shepp–Logan phantom at different resolutions. High attenuation is shown here as darker shades of gray and low attenuation as lighter shades.

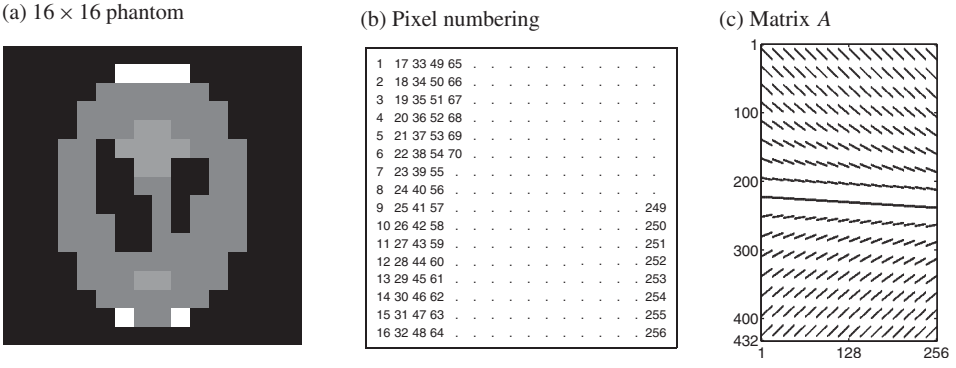


Figure 2.17. (a) Low-resolution Shepp–Logan phantom (16×16). Here black denotes zero attenuation, and white denotes maximum attenuation. (b) Numbering of the pixels in a 16×16 image when interpreted as a vector in $\mathbb{R}^{256}$. (c) Nonzero elements of the 432×256 tomographic measurement matrix.

What is the size of A? Obviously, the number of columns must be 256, the number of pixels in a 16×16 image. The `radon.m` algorithm picked automatically the value $N = 27$ in formula (2.45), so A has $JN = 16 \cdot 27 = 432$ rows. We observe that there are 11086 nonzero elements out of the total 110592; this means that roughly 90% of the elements in A are zero.

Three-dimensional X-ray tomography problems can be approached similarly to the above explanation using voxelization instead of pixelization and by tracing the paths of X-rays through the voxels in a three-dimensional manner. However, for illustration and simplicity purposes we stick to the two-dimensional case in this book. In principle there is no essential difference between the two- and three-dimensional cases, only the computations will be significantly more demanding in three dimensions.

2.3.5 Naïve reconstruction

The 16×16 Shepp–Logan phantom used in Section 2.3.4 has too low a resolution to really show the intended anatomic features properly. In the rest of the book we work with the 50×50 Shepp–Logan phantom when we need to construct the matrix A explicitly, and with the 512×512 phantom when we illustrate matrix-free large-scale methods in Section 9. See Figure 2.16.

Next we wish to experiment with naïve reconstructions of the 50×50 Shepp–Logan phantom shown in Figure 2.16(c). We choose the number of projection directions to be $J = 50$ in formula (2.44). We construct the measurement matrix A column by column as explained in Section 2.3.4. MATLAB's `radon.m` algorithm picked automatically the value $N = 75$ in formula (2.45), so A has $JN = 50 \cdot 75 = 3750$ rows. We arrive at the following measurement model:

$$A \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_{2500} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_{3750} \end{bmatrix}, \tag{2.49}$$

where the elements of the 50×50 pixel image $\mathbf{f}$ and the elements of the 75×50 sinogram $\mathbf{m}$ are numbered similarly to Figure 2.17(b). The phantom and sinogram are shown in Figure 2.18.

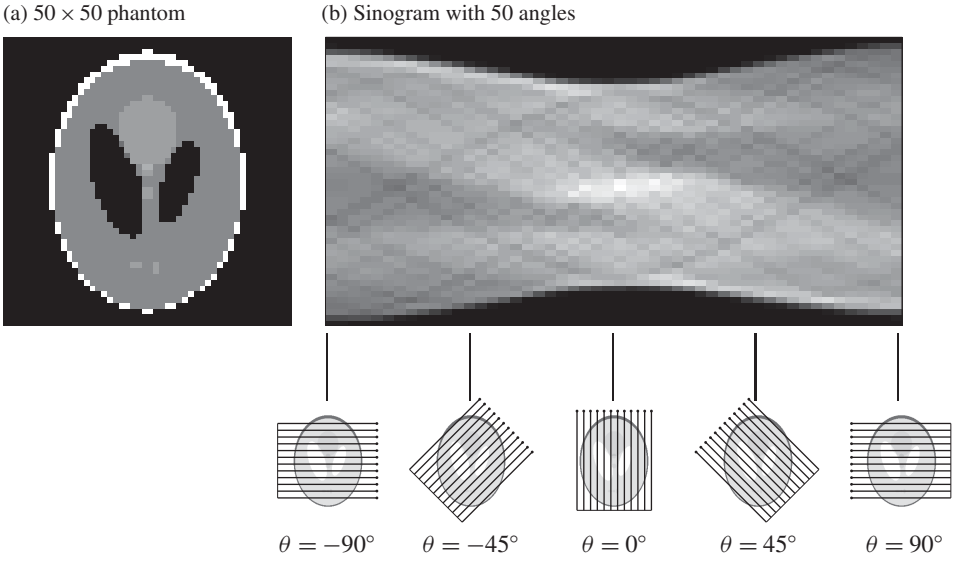


Figure 2.18. (a) Shepp–Logan phantom at resolution 50×50. Here black denotes zero attenuation, and white denotes maximum attenuation. (b) Measured data (involving inverse crime) in sinogram form, where the horizontal axis is the angle θ and the vertical axis the variable s in (2.40). We have removed some purely zero rows from the top and bottom of the sinogram for clarity. Underneath the sinogram we show some of the projection directions to illustrate the structure of the sinogram.

Once A is in place, we can try out naïve inversion, but not in the sense of (1.4) since A is not a square matrix. Instead we use least-squares naïve inversion defined as follows:

$$\mathbf{f} \approx (A^T A)^{-1} A^T \mathbf{m}. \tag{2.50}$$

Derivation and interpretation of formula (2.50) is postponed to Section 5.2. The result of applying (2.50) to ideal (noise-free) tomographic data is shown in Figure 2.19(b), and it looks very good indeed. The relative error of this reconstruction is very small. Perhaps we can conclude that we succeeded in reconstructing the 50×50 phantom from indirect tomographic measurements?

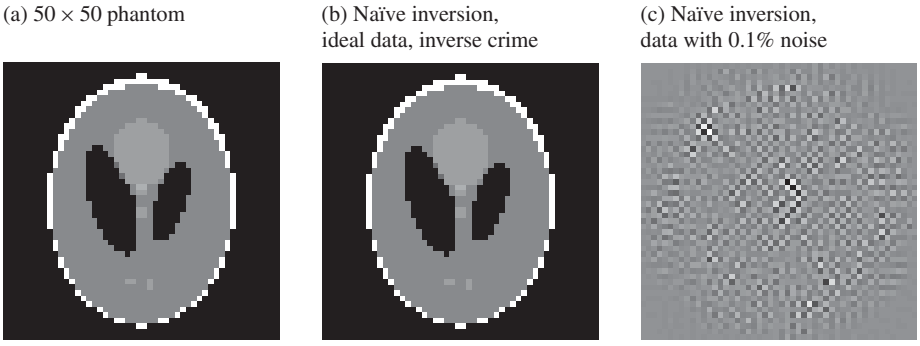


Figure 2.19. (a) Shepp–Logan phantom at resolution 50×50. (b) Result of naïve inversion (2.50) from noise-free data. The seemingly successful result is not to be trusted because an inverse crime was committed. (c) Result of naïve inversion (2.50) from data contaminated by 0.1% noise. The much worse performance of (c) compared to (b) indicates that naïve inversion is not stable with respect to noise.

Before jumping to such a (wrong) conclusion, let us see what a small amount of measurement noise does to the naïve reconstruction. We add white noise of relative amplitude 0.1% to the sinogram and try formula (2.50) again. The result is shown in Figure 2.19(c), and it consists merely of numerical garbage. This shows that formula (2.50) is not practically useful since real measurements always contain noise.

2.3.6 Naïve reconstruction without inverse crime

We wish to avoid the inverse crime evident in Figure 2.19. To this end, we interpolate our data from tomographic data simulated using the Shepp–Logan phantom on a twice finer grid (100×100) than the grid used in the naïve reconstruction. This can be done conveniently as the phantom is defined analytically using ellipses, so it can be evaluated with arbitrary resolution. The measurement angles are the same. See Figure 2.20 for plots of the data.

Figure 2.21 shows the result of applying naïve reconstruction to the crime-free data. Now the result has unacceptable quality even when there is no added noise.

Exercise 2.3.1. *Let* $\mathbf{f} \in \mathbb{R}^8$ *be a signal and* $\mathbf{p} = [\mathbf{p}_{-1} \quad \mathbf{p}_0 \quad \mathbf{p}_1]^T$ *a point spread function. Write down the* 8×8 *matrix* A *modeling the one-dimensional convolution* (2.11) *with the assumption that* $\mathbf{f}_{j-\ell} = 0$ *for the cases* $j - \ell < 1$ *and* $j - \ell > 8$.

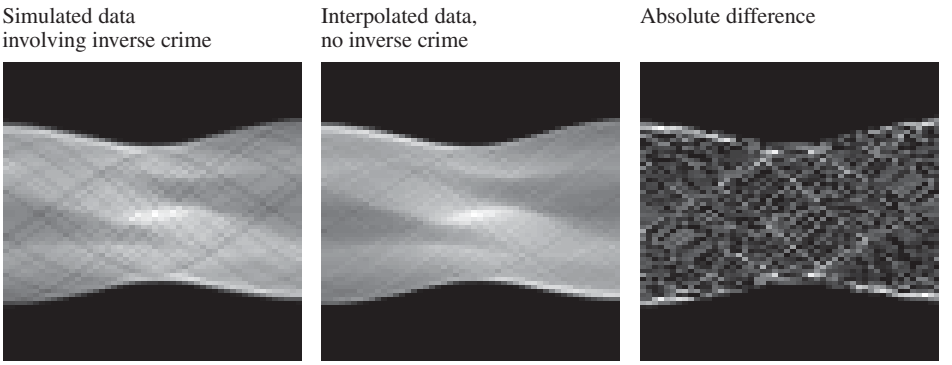


Figure 2.20. Tomographic data with and without inverse crime. Left: ideal data obtained by applying the measurement model matrix to the Shepp–Logan phantom at the final reconstruction resolution of 50×50 pixels. Middle: tomographic data computed from 100×100 Shepp–Logan phantom (at same measurement angles but finer arrangement of X-rays) and interpolated to lower resolution. Right: absolute difference between the two data sets.

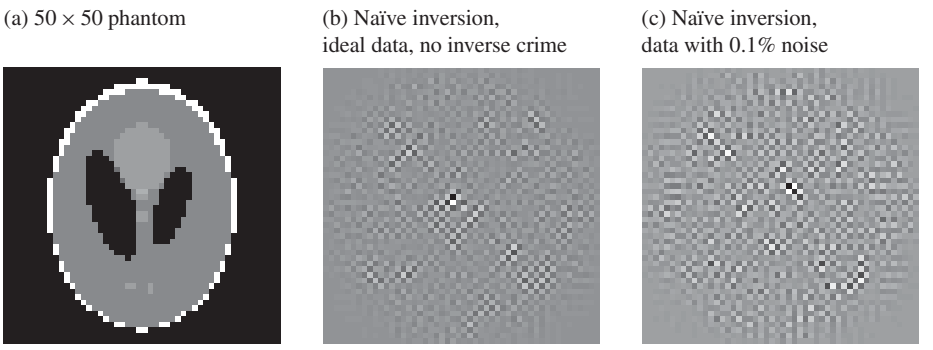


Figure 2.21. (a) Shepp–Logan phantom at resolution 50×50. (b) Result of naïve inversion (2.50) from noise-free data with no inverse crime. (c) Result of naïve inversion (2.50) from data contaminated by 0.1% noise. Compare to Figure 2.19.

Exercise 2.3.2. *Assume that an X-ray detector provides the proportional number Mc instead of the actual photon count c. Here $M > 0$ is a positive constant. Show that the calibration procedure described in Figure* 2.12 *works fine even if M is unknown.*

Exercise 2.3.3. *Define f and I_0 and I_1 appropriately in the context of the simple example shown in Figure* 2.12*. Furthermore, describe the measurement data in Figure* 2.12 *in terms of formula* (2.38)*.*

Exercise 2.3.4. *In Figure* 2.22*, thin lines depict pixels and thick lines X-rays. Give a numbering to the nine pixels ($\mathbf{f} \in \mathbb{R}^9$) and to the six X-rays ($\mathbf{m} \in \mathbb{R}^6$), and construct the matrix A for the measurement model $\mathbf{m} = A\mathbf{f}$. The length of the side of a pixel is one.*

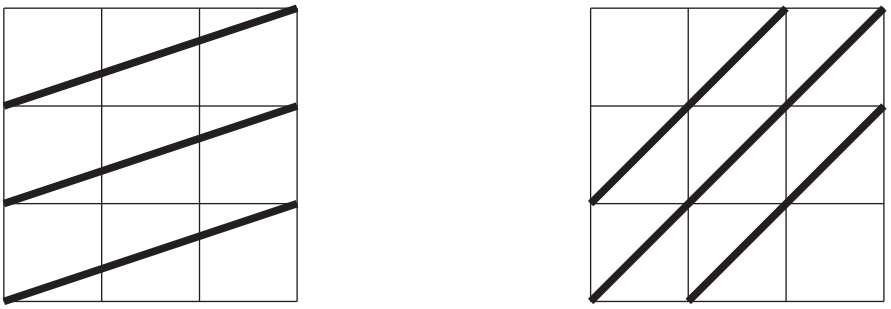


Figure 2.22. Tomographic X-ray measurement configuration related to Exercise 2.3.4. Thick lines depict X-rays, and the length of the side of a pixel is one.

Exercise 2.3.5. *Show that the following targets produce exactly the same data in the measurement model* (2.48):

4	4	5
1	3	4
1	0	2

5	6	2
1	5	2
4	0	−1

(a) *What's wrong with the negative value* -1 *above?* (b) *Can you find more examples that produce the same data but have only nonnegative entries?*