

# Report Computed Tomography

Royce Kraakman<sup>1</sup>

<sup>1</sup> Department of Mathematics, Utrecht University, Budapest 6, Utrecht, 3584 CD

\*r.kraakman@students.uu.nl

**Abstract:** This report will give a short overview of computed tomography: from sinograms to reconstruction methods. At last, we will consider some real world cases. © 2022 The Author(s)

## 1. Introduction

In 1895, after a serendipitous discovery, X-rays were born. It's creator – Wilhelm Röntgen – published an X-ray photo of his wife's hand and after that things started to blossom. Nowadays X-ray photography is widely adopted in healthcare. To capture an X-ray photograph a CT scanner is used. However, using a CT scanner does not directly give us a X-ray photograph of an object. Mathematics, developed by Johan Radon, Stefan Kaczmarz, Allan MacLeod Cormack and many others, is used to process the data generated by the CT scanner to reconstruct the object.

### 1.1. The Physics

The CT scanner shoots X-rays beams through an object from several angles. When the X-ray beam hits the object it loses it's intensity through absorption and scattering. The law of Beer-Lambert gives us an approximation of this process. Given an initial beam intensity  $I_0$ , material thickness  $h$  and the material attenuation  $\mu$ , the outgoing beam intensity  $I$  is given by

$$I = I_0 e^{-\mu h}. \quad (1)$$

Of course, in practice most objects have multiple materials. For example, if we want to scan a person's head multiple materials are present. Skin, bones, brain tissue, etc. We can express Beer-Lambert's law for multiple materials  $m = 1, 2, \dots, n$  as

$$I = I_0 \exp \left( - \sum_{m=1}^n \mu_m h_m \right). \quad (2)$$

As the amount of materials  $n$  increases to infinity we have

$$I = I_0 \exp \left( - \lim_{n \rightarrow \infty} \sum_{m=1}^n \mu_m h_m \right) = I_0 \exp \left( - \int \mu(t) dt \right). \quad (3)$$

Taking the logarithm of both sides gives

$$\log I = \log I_0 - \int \mu(t) dt \implies -\log \frac{I}{I_0} = \int \mu(t) dt. \quad (4)$$

We can observe the left side, since these are outgoing and incoming signals. However, the right side is unknown. This gives us a useful relation between observed data and unknown data.

### 1.2. Sinograms

The data collected by a CT scanner produces a sinogram. We can express a sinogram mathematically with the Radon transform. Given an image  $u(x, y)$  the sinogram is given by

$$f(s, \theta) = \int_{-\infty}^{\infty} u(x(s, t), y(s, t)) dt, \quad (5)$$

where  $x$  and  $y$  are given by

$$x(s, t) = s \cos \theta + t \sin \theta, \quad y(s, t) = s \sin \theta - t \cos \theta. \quad (6)$$

To give an example of the Radon transform we consider a uniform square

$$u(x, y) = \begin{cases} 1, & |x|, |y| \leq 1, \\ 0, & \text{other.} \end{cases} \quad (7)$$

We use another representation of the Radon transform

$$f(s, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x, y) \delta(x \cos \theta + y \sin \theta - s) dx dy, \quad (8)$$

where  $\delta$  is the Dirac delta function

$$\delta = \begin{cases} 1, & x \cos \theta + y \sin \theta = s, \\ 0, & \text{other.} \end{cases} \quad (9)$$

The alternative Radon transform representation can be linked to (5) by first substituting (6) and then integrating the Dirac delta function

$$\begin{aligned} f(s, \theta) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x, y) \delta(x \cos \theta + y \sin \theta - s) dx dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(s \cos \theta + t \sin \theta, s \sin \theta - t \cos \theta) \delta(s - t) dx dy \\ &= \int_{-\infty}^{\infty} u(x(s, t), y(s, t)) dt. \end{aligned} \quad (10)$$

Solving for  $x$  gives

$$x = \frac{s - y \sin \theta}{\cos \theta} \implies \delta \left( \frac{s - y \sin \theta}{\cos \theta} \cos \theta + y \sin \theta - s \right) = 1. \quad (11)$$

Substituting this value in the integral gives

$$\int_{-\infty}^{\infty} u \left( \frac{s - y \sin \theta}{\cos \theta}, y \right) dy. \quad (12)$$

We now need to solve for the following inequalities

$$\begin{cases} -1 \leq y \leq 1, \\ -1 \leq \frac{s - y \sin \theta}{\cos \theta} \leq 1. \end{cases} \quad (13)$$

Solving the second inequality for  $y$  gives

$$\begin{cases} -1 \leq y \leq 1, \\ \frac{s - \cos \theta}{\sin \theta} \leq y \leq \frac{s + \cos \theta}{\sin \theta}. \end{cases} \quad (14)$$

Now define the following boundaries

$$\begin{aligned} B_1(s, \theta) &:= \max \left\{ -1, \frac{s - \cos \theta}{\sin \theta} \right\}, \\ B_2(s, \theta) &:= \min \left\{ 1, \frac{s + \cos \theta}{\sin \theta} \right\}, \end{aligned} \quad (15)$$

so that the integral becomes

$$f(s, \theta) = \int_{B_1(s, \theta)}^{B_2(s, \theta)} dy = B_2(s, \theta) - B_1(s, \theta). \quad (16)$$

We see a visualisation of this sinogram in figure 1. As we can see in this figure the sinogram becomes clearer as we increase the amount of pixels used.

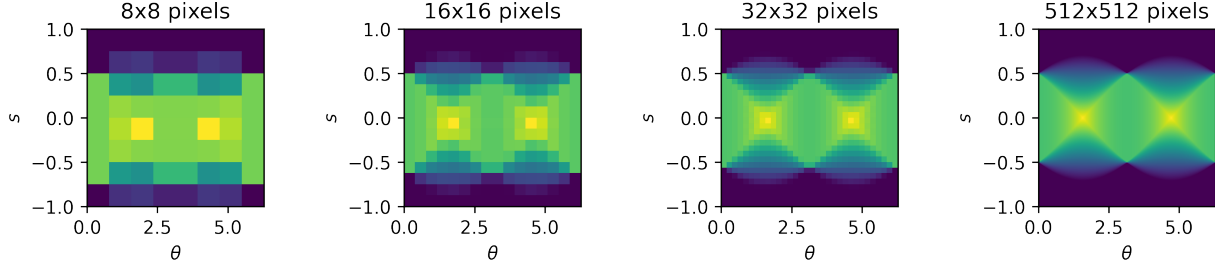


Fig. 1. Unit square sinograms with increasing amount of pixels

### 1.3. Computed Tomography

In Computed Tomography we try to reconstruct an image from a sinogram. In section 3 we discuss various methods to reconstruct an image from a sinogram. We see in the left side of figure 2 that the reconstruction errors increases, as the noise parameter  $\sigma$  increases. In the right side we can see that the error decreases, as we take more samples. Some questions arise:

1. is the Radon transform invertible?
2. what can we do about noisy or undersampled data?

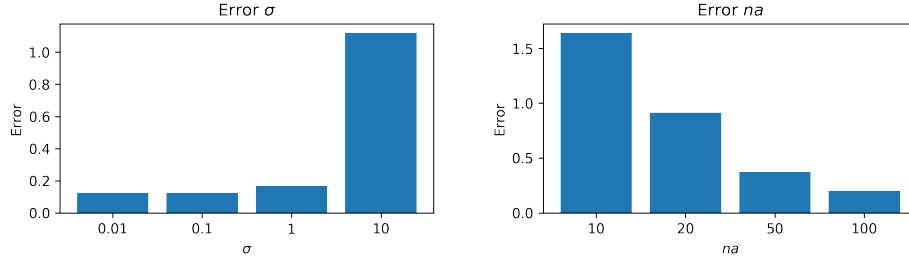


Fig. 2. Reconstruction errors for the Shepp-Logan phantom

## 2. Discrete Radon Transform

To actually numerically calculate the Radon transform, we need a discretised version. Firstly, we consider the Radon transform as defined in 5. Without loss of generality we assume that the detector and image are scaled, such that  $s \in [-1, 1]$  and  $(x, y) \in [-1, 1]^2$ . We make the following discretizations:

1. create a uniform 1D grid for the detector of size  $n_d$  and cell-size  $\frac{2}{n_d}$ ;
2. create a uniform 2D grid for the image of size  $n_x \cdot n_y$  and pixel-size  $\frac{2}{n_x} \times \frac{2}{n_y}$ .

For the  $j$ -th image pixel and  $i$ -th detector element, the contribution at angle  $\theta$  is given by

$$a_{ij}(\theta) = \int_{-\infty}^{\infty} \phi_j(x(s_i, t), y(s_i, t)) dt, \quad (17)$$

where  $\phi_j(x, y)$  is the  $j$ -th pixel of the image  $u(x, y)$ . For a single angle  $\theta$  we write

$$A(\theta) = \begin{pmatrix} a_{11}(\theta) & a_{12}(\theta) & \dots & a_{1n_x \cdot n_y}(\theta) \\ a_{21}(\theta) & a_{22}(\theta) & \dots & a_{2n_x \cdot n_y}(\theta) \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_d 1}(\theta) & a_{n_d 2}(\theta) & \dots & a_{n_d n_x \cdot n_y}(\theta) \end{pmatrix} \in \mathbb{R}^{n_d \times n_x \cdot n_y}. \quad (18)$$

Now given a discrete set of angles  $\{\theta_k\}_{k=1}^{n_\theta}$  we have

$$A = \begin{pmatrix} A(\theta_1) \\ A(\theta_2) \\ \vdots \\ A(\theta_{n_\theta}) \end{pmatrix}. \quad (19)$$

The relation between the discretised sinogram and the image is given by

$$f = Au. \quad (20)$$

We can use this relationship to retrieve the image  $u$ .

### 2.1. Sparse Matrices

Sparse matrices are very useful in calculating the discrete Radon transform. Since we are dealing with images, a lot of values are equal to zero. It would be a waste to use regular matrix multiplication. Working with sparse matrices is faster, since we are not explicitly processing the zero values, saving us computational time.

### 2.2. Matrix-free Implementation

In 3D applications matrices are very large. In case we use double precision, storage may take several Terabytes in practice. To reduce storage-costs we can use a matrix-free implementation, which in return can increase computational time.

## 3. Reconstruction Methods

### 3.1. Fourier Reconstruction

The first reconstruction method we will discuss is the Fourier reconstruction, which heavily relies on the Fourier slice theorem.

#### 3.1.1. Fourier Transform

For a  $n$ -dimensional function  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  the Fourier transform is given by

$$\hat{g}(\xi) = \int_{\mathbb{R}^n} g(\mathbf{x}) e^{-2\pi i \mathbf{x} \cdot \xi} d\mathbf{x}. \quad (21)$$

It's inverse is given by

$$g(\mathbf{x}) = \int_{\mathbb{R}^n} \hat{g}(\xi) e^{2\pi i \mathbf{x} \cdot \xi} d\xi. \quad (22)$$

#### 3.1.2. Fourier Slice Theorem

Using the Fourier slice theorem we can relate the 1D Fourier transform of the sinogram  $f$  to the 2D Fourier transform of the image  $u$

$$\begin{aligned} \hat{f}(\sigma, \theta) &= \int_{-\infty}^{\infty} f(s, \theta) e^{-2\pi i (\sigma s)} ds \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(\sigma \cos \theta, \sigma \sin \theta) e^{-2\pi i (\sigma (x \cos \theta + y \sin \theta))} dx dy \\ &= \hat{u}(\sigma \cos \theta, \sigma \sin \theta). \end{aligned} \quad (23)$$

#### 3.1.3. Fourier Reconstruction

Using the Fourier slice theorem we can reconstruct the image  $u$  as follows:

1. apply the Fourier transform to each row of the sinogram  $f$ ;
2. rearrange each Fourier transformed row as a diameter of a circle;
3. apply the 2D inverse Fourier transform to the circle to obtain the image.

An example of these steps can be seen in figure 3.

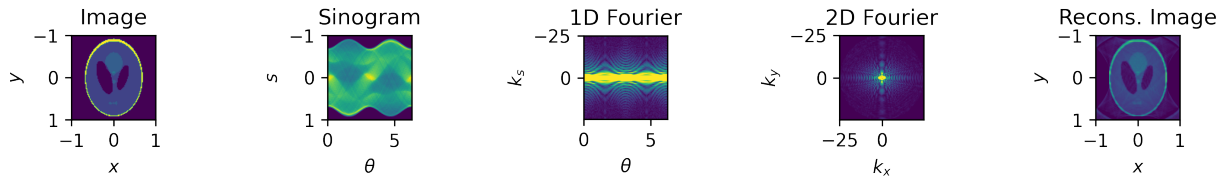


Fig. 3. Fourier reconstruction step-by-step

However, using this method two problems arise

1. after phase 2 in the Fourier slice theorem steps we have a polar representation, but the Fourier methods used require Cartesian coordinates;
2. since we wrap the Fourier transformed rows to a circle, lower frequencies have more data. Which means that those frequencies are mapped more accurately than high frequencies, possibly resulting in blurry images.

For the first problem we can apply a function to convert the data points from polar to Cartesian, see figure 4. To overcome the second problem we consider filtered back projection in section 3.2.

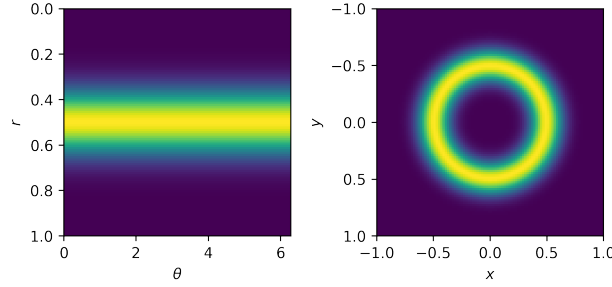


Fig. 4. From polar to Cartesian coordinates

#### 3.1.4. Discrete Fourier Transform

In practice we are dealing with discrete measurements. We saw earlier in chapter 2 how to construct a discrete sinogram. Denote each discrete measurement as  $f_{ij} = f(s_i, \theta_j)$ . From this we would like to represent the image  $u$  on a pixel-grid with intensities  $u_i = u(\mathbf{x}_i)$ . To use the Fourier transform on a discrete space we introduce the discrete Fourier transform (DFT). Given a sequence  $\{g_k\}_{k=0}^n$  the DFT is defined by

$$\hat{g}_k = \sum_{j=0}^{n-1} g_j \exp\left(-\frac{2\pi i}{n} kj\right), \quad k = 0, 1, \dots, n-1. \quad (24)$$

### 3.2. Filtered Back Projection

The second reconstruction method we consider is filtered back projection, which uses the adjoint of the Radon transform to project the sinogram back to the image.

#### 3.2.1. Adjoint of the Radon Transform

What the Radon transform basically does is integrating along a set of lines. This gives us a sinogram. The adjoint of the Radon transform reverses this process; it smears the Radon transform back on those set of lines. The continuous adjoint of the Radon transform is given by

$$u(\mathbf{x}) = \frac{1}{2\pi} \int_0^{2\pi} f(\mathbf{x} \cdot \mathbf{n}_\theta, \theta) d\theta, \quad (25)$$

where  $\mathbf{n}_\theta = (\cos \theta, \sin \theta)$ . In the discrete setting the adjoint Radon transform is equal to the transpose of the Radon transform matrix (19)

$$A^* = A'. \quad (26)$$

#### 3.2.2. Filtered Back Projection

Some useful properties of the Radon transform are given below

$$\begin{aligned} R\Delta_x u &= \partial_s^2 R u \\ R^* \partial_s^2 f &= \Delta_x R^* f \\ R^* R u &= 2(-\Delta_x)^{-1/2} u \\ R R^* f &= 2(-\partial_s^2)^{-1/2} f. \end{aligned} \quad (27)$$

Using these properties we obtain an expression for the image  $u$  using 2D-filtering after the back projection

$$\begin{aligned}
Ru &= f \\
\Rightarrow R^*Ru &= R^*f \\
\Rightarrow 2(-\Delta_x)^{-1/2}u &= R^*f \\
\Rightarrow u &= \frac{1}{2}(-\Delta_x)^{1/2}R^*f.
\end{aligned} \tag{28}$$

We can also apply 1D-filtering before the back projection

$$u = \frac{1}{2}R^*(-\partial_s^2)^{1/2}f. \tag{29}$$

In figure 5 we can see how an image is reconstructed using filtered back projection. Here we are using a ramp filter. Other filters also exist and in figure 6 we also see the cosine and Shepp-Logan filters. Note that the shape seems preserved, but the coloring is not accurate.

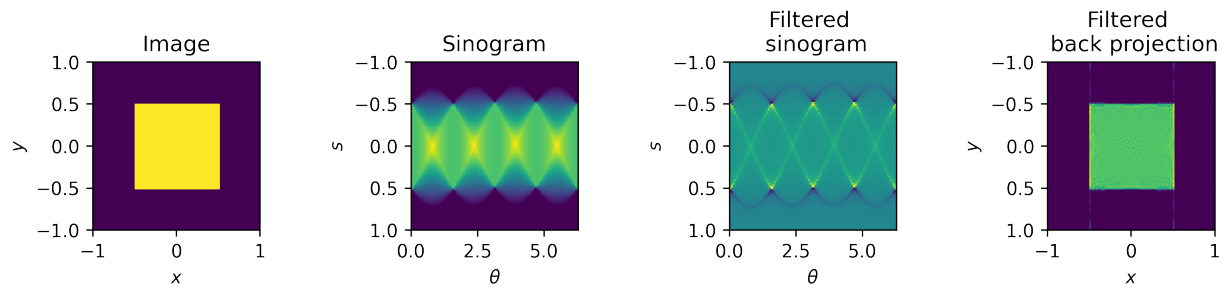


Fig. 5. Filtered back projection step-by-step

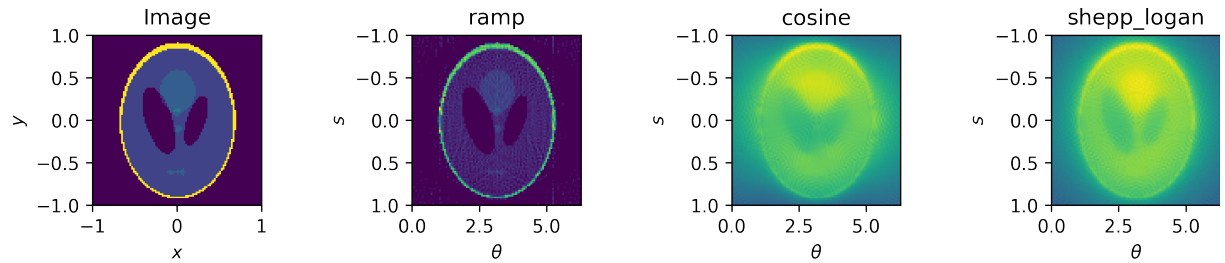


Fig. 6. Reconstructions using ramp, cosine and Shepp-Logan filters

In the tables below we compare the Fourier reconstruction and the filtered back projection for various  $\sigma$  and sample sizes. The  $\sigma$  parameter adds noise to the sinogram. The filtered back projection seems to be more robust for small distortions to the sinogram, while the Fourier reconstruction performs better for small sample sizes. For larger sample size the filtered back projection is more accurate.

Table 1. Errors for Fourier reconstruction and filtered back projection with added noise  $\sigma$

$\sigma$	Fourier reconstruction	Filtered Back Projection
0.01	0.684	0.1257
0.1	0.684	0.1262
1	0.6866	0.1681
10	0.9129	1.1205

Table 2. Errors for Fourier reconstruction and filtered back projection for various sample sizes

Amount of Samples	Fourier reconstruction	Filtered Back Projection
10	0.7311	1.6401
20	0.6991	0.9141
50	0.6854	0.3731
100	0.6839	0.2024

### 3.3. Algebraic Reconstruction

Lastly, we consider algebraic reconstruction methods.

#### 3.3.1. Richardson Iteration

Given a linear system of equations

$$Ku = f, \quad (30)$$

the Richardson iteration is given by

$$u^{(k+1)} = u^{(k)} - \alpha K^T (Ku^{(k)} - f). \quad (31)$$

For  $u^{(0)} = 0$  and  $0 < \alpha < 2/\|K\|^2$  the method converges to the minimum-norm solution of the normal equations

$$K^T Ku = K^T f. \quad (32)$$

To retrieve the image  $u$ , we can apply the Richardson iteration to the system given at (20).

#### 3.3.2. Row-action Methods

A con of using the Richardson iteration is that the computation cost is dominated by the Radon transform and the adjoint Radon transform. To decrease computational cost we can process one X-ray at a time with the Kaczmarz method, which is given by

$$u^{(k+1)} = u^{(k)} - \frac{f_i - k_i^T u^{(k)}}{\|k_i\|_2^2} k_i^T, \quad (33)$$

where  $k_i$  is the  $i$ -th row of  $K$ . In figure 7 we see that the Richardson iteration has an optimum in the first couple iterations, while the Kaczmarz method takes longer to reach an optimal point and could even continue to reach a lower point. Also, looking at the errors of both methods, the Richardson iteration has a better optimal point.

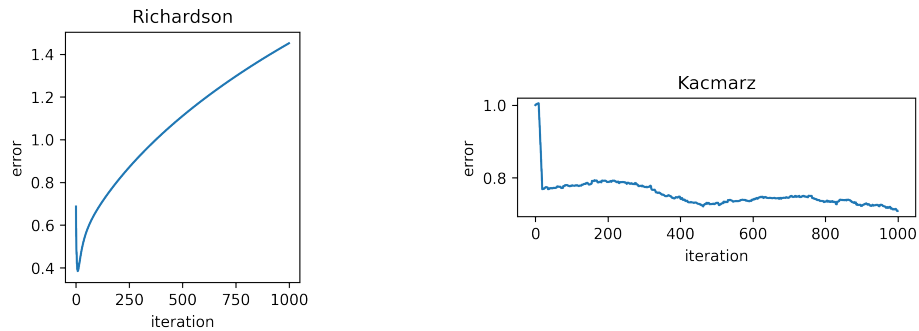


Fig. 7. Errors for the Richardson and Kaczmarz methods reconstructing a square

## 4. Real World Cases

In this section we consider three real-world sinograms.

### 4.1. Walnut

In figure 8 we see the sinogram of the walnut [1] on the left and the reconstructed image on the right. Filtered back projection was used with a ramp filter.

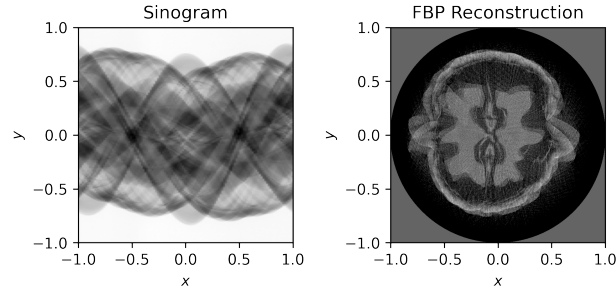


Fig. 8. Reconstruction of a walnut

### 4.2. Cheese

Figure 9 displays the sinogram of the cheese [2] on the left and the reconstructed image on the right. Although the shape was preserved, the reconstructed image lacks contrast. The image was reconstructed using filtered back projection with a ramp filter.

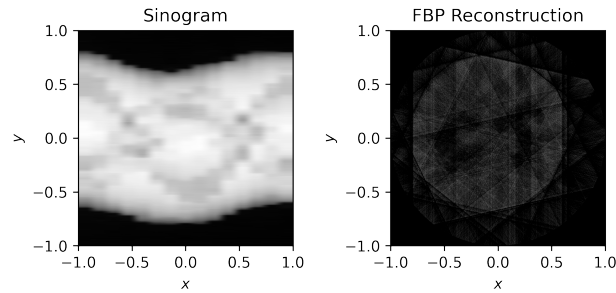


Fig. 9. Reconstruction of a slice of cheese

### 4.3. Lotus

Lastly, the lotus [3] can be seen in figure ?? . In this reconstruction we also applied filtered back projection with a ramp filter.

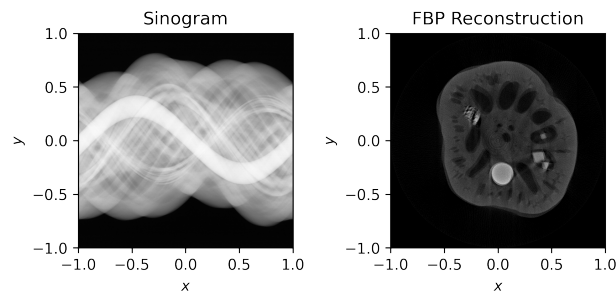


Fig. 10. Reconstruction of a lotus



## References

1. Keijo Hämäläinen, Lauri Harhanen, Aki Kallonen, Antti Kujanpää, Esa Niemi and Samuli Siltanen, “Tomographic X-ray data of a walnut“
2. Tatiana A. Bubba, Markus Juvonen, Jonatan Lehtonen, Maximilian März, Alexander Meaney, Zenith Purisha and Samuli Siltanen, “Tomographic X-ray data of a carved cheese“
3. Tatiana A. Bubba, Andreas Hauptmann, Simo Huotari, Juho Rimpeläinen and Samuli Siltanen, “Tomographic X-ray data of a lotus root filled with attenuating objects“