Lab 4: The Radon Transform, the filtered backprojection

The Radon transform is the mathematical foundation of Computerized Tomography. We recall the definition in $\mathbb{R}^2$:

The Radon transform of a given function on $\mathbb{R}^2$ is a function defined on the set of all lines of $\mathbb{R}^2$. Every line is parametrized by a normal vector to the line, $\theta \in \mathbb{T}$, and its (signed) distance from the origin $s \in \mathbb{R}$, so that it can be written as

$$\theta_s := \{\, x \in \mathbb{R}^2 \;:\; x \cdot \theta = s \,\}.$$

Let $f : \mathbb{R}^2 \to \mathbb{R}$ then the *Radon transform of f* is the function $\mathcal{R}f$, defined on the set of lines of $\mathbb{R}^2$, whose value at a line equals the integral of f on that line, i.e.

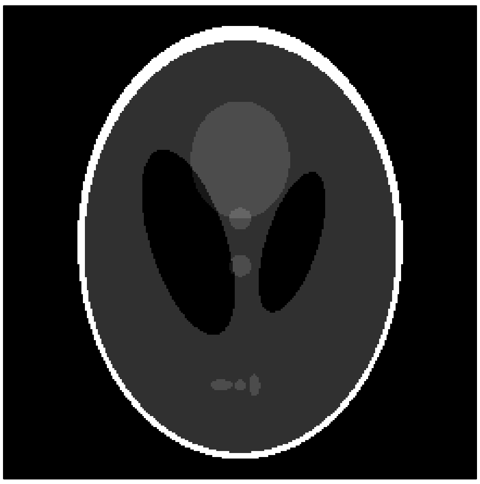
$$\mathcal{R}f(\theta, s) = \mathcal{R}_\theta f(s) := \int_{\theta_s} f(x)\, dx.$$

In Matlab/Octave this is implemented by the function

```
G = radon(F, 0:179);
```

Here F is a representation of an image given by a matrix $n \times m$ of real numbers that represent the graylevels of the image. One can try for example with

```
F= phantom(256);
```



the radon trasnform is an even function, R(theta,s) = R(-theta,-s) which loads in the matrix F an image of 256×256 pixels. You can see it with the instruction `imshow(F);` which is called the Logan-Shepp phantom test image and is used in the field as a standard yardstick.

As a second paramente in the function `radon` you can pass a parameter as in `radon(F,theta)` where `theta` is a vector of angles (by default is 0:179). The output is a matrix with as many columns as angles and each column is a vector with the Radon transform along that direction. The inverse radon transform is obtained with `FR = iradon(G, 0:179);` and you can see the reconstruction with `imshow(FR);`

Exercise 1. *Implement the function* `iradon`*. The program should takes as an input a matrix with columns corresponding to the Radon transform along the angles given in the vector that you pass a second parameter. The outpout of the program should be a 'matrix that correspond to an approximation to the original image.*

There are several algorithms to reconstruct f from its Radon transform. The first one that we will address in this lab is the filtered backprojection. The backprojection is the formal adjoint operator to the Radon transform.

The *backprojection* of a function g on $\mathbb{T} \times \mathbb{R}$ is the function

$$\mathcal{R}^{\#}g(x) := \int_{\mathbb{T}} g(\theta, x \cdot \theta)\, d\theta \qquad (x \in \mathbb{R}^2).$$

Observe that if $g = \mathcal{R}f$ then $g(\theta, x \cdot \theta)$ is the integral of f on the line passing through the point $x \in \mathbb{R}^2$ which is orthogonal to $\theta \in \mathbb{T}$, so $\mathcal{R}^{\#}g(x)$ is the "mean" of the integrals of f on the lines passing through x. One of the basic properties of the backprojection is that $\mathcal{R}^{\#}$ is the formal adjoint operator of $\mathcal{R}$:

$$\int_{\mathbb{T}} \int_{\mathbb{R}} \mathcal{R}f(\theta, s)\, g(\theta, s)\, ds\, d\theta = \int_{\mathbb{R}^2} f(x)\, \mathcal{R}^{\#}g(x)\, dx.$$

But more important for our porpouses:

$$\boxed{f * (\mathcal{R}^{\#}g) = \mathcal{R}^{\#}(\mathcal{R}f * g).}$$

Proof.

$$f * (\mathcal{R}^{\#}g)(x) = \int_{\mathbb{R}^2} \mathcal{R}^{\#}g(x-y) f(y) dy = \int_{\mathbb{R}^2} \int_{\mathbb{T}} g(\theta, (x-y)\cdot\theta)\, d\theta f(y)\, dy =$$
$$= \int_{\mathbb{T}} \int_{\mathbb{R}^2} g(\theta, (x-y)\cdot\theta) f(y)\, dy d\theta.$$

We make the change of variables in $\mathbb{R}^2$ $y = s\theta + z$ where $z \in \theta^\perp$, and we obtain:

$$\begin{aligned} f * (\mathcal{R}^\# g)(x) &= \int_{\mathbb{T}} \int_{\mathbb{R}} \int_{\theta^\perp} g(\theta, x\cdot\theta - s) f(s\theta + z)\, dz ds d\theta \\ &= \int_{\mathbb{T}} \int_{\mathbb{R}} g(\theta, x\cdot\theta - s)\mathcal{R}f(\theta, s)\, ds d\theta = \\ &= \int_{\mathbb{T}} (g * \mathcal{R}f)(\theta, x\cdot\theta) = \mathcal{R}^\#(g * \mathcal{R}f)(x). \end{aligned}$$

□

This is for arbitray f and g. Now we take $g = v$ and $V = \mathcal{R}^\# v$, we have $f * V = \mathcal{R}^\#(v * \mathcal{R}f)$ for all v and f. Finally if we denote by $g = \mathcal{R}f$, the previous identity is

$$(V * f)(x) = \mathcal{R}^\#(v * g)(x) = \int_{\mathbb{T}} (v * g)(\theta, x\cdot\theta) d\theta. \tag{1}$$

The key feature of the filtered backprojection algorithm is the choice of a so-called *point-spread function* V approximating the Dirac mass δ_0. Then the left-hand side of the identity above approximates $f(x)$.

Once v is determined, using that $\mathcal{R}^\# v = V$, the integral on the right-hand side of the identity has to be discretized.

Identity (1) explains the name of the algorithm: first the data g are filtered with v (this gives $v * g$) and then the backprojection $\mathcal{R}^\#$ is applied.

Usually V is chosen so that $V * f$ deletes or de-emphasizes high frequencies, which are mostly observation noise. Since f has essential bandwidth Ω (this means that it can be very well approximated by a function with bandwith Ω, one looks for V such that

$$(V * f)\widehat{\ }(\zeta) \simeq \begin{cases} \widehat{f}(\zeta), & \text{if } |\zeta| \le \Omega, \\ 0, & \text{if } |\zeta| > \Omega. \end{cases}$$

The relationship between V and v is explicit through the following distributional identity [**2**, Theorem 2.4]: if g is even then

$$(\mathcal{R}^\# g)\widehat{\ }(\zeta) = 2|\zeta|^{-1}\widehat{g}(\zeta/|\zeta|, |\zeta|).$$

In practice only radial symmetric functions $V(x) = V(|x|)$ are considered. Then v does not depend on θ and it is an even function of s. In this particular situation the identity above gives

$$\widehat{V}(\zeta) = 2|\zeta|^{-1}\widehat{v}(|\zeta|), \tag{2}$$

where $\widehat{V}$ indicates the 1-dimensional Fourier transform.

In the usual cases the point-spread function V can be computed explicitly from $\widehat{V}$.

In order to reconstruct accurately functions f with essential bandwidth Ω we can take, for instance $\widehat{V}(\zeta) = \mathcal{X}_{B(0,\Omega)}(\zeta)$. More generally, consider a filter $\widehat{\phi}(\sigma)$ close to 1 when $|\sigma| \le 1$ and with $\widehat{\phi}(\sigma) = 0$ for $|\sigma| > 1$, and define

$$\widehat{V}_\Omega(\zeta) = \widehat{\phi}\left(\frac{|\zeta|}{\Omega}\right).$$

According to (2), the corresponding function v_Ω (such that $\mathcal{R}^\# v_\Omega = V_\Omega$) is determined by the identity

$$\widehat{v}_\Omega(\sigma) = \frac{1}{2}|\sigma|^{-1}\widehat{\phi}\left(\frac{|\sigma|}{\Omega}\right). \tag{3}$$

In applications many different $\widehat{\phi}$'s have been proposed. It seems, however, that there is no justification for any specific choice other than the expiremental results. Next, we show three common filters.

(a) *Ram-Lak filter.* Introduced in this context by Ramachandran and LakshmiNarayanan (1971). It is associated to the standard low-pass filter $\widehat{\phi}(\sigma) = \mathcal{X}_{[0,1]}(\sigma)$. Here (3) yields $\widehat{v}_\Omega(\sigma) = 1/2|\sigma|\mathcal{X}_{[0,1]}(|\sigma|/\Omega)$, hence

$$v_\Omega(s) = \int_{\mathbb{R}} \widehat{v}_\Omega(\sigma)e^{2\pi i\sigma s}d\sigma = \frac{1}{2}\int_{-\Omega}^{\Omega} |\sigma|e^{2\pi i\sigma s}d\sigma.$$

Splitting the integral for $\sigma > 0$ and $\sigma < 0$, and integrating by parts we get

$$\begin{aligned}\int_{-\Omega}^{\Omega} |\sigma|e^{2\pi i\sigma s}d\sigma &= 2\Omega^2\frac{\sin(2\pi\Omega s)}{2\pi\Omega s} + 2\frac{\cos(2\pi\Omega s)-1}{(2\pi s)^2}\\ &= 2\Omega^2\Big(\operatorname{sinc}(2\pi\Omega s) - \frac{1}{2}\big(\operatorname{sinc}(\pi\Omega s)\big)^2\Big),\end{aligned}$$

where $\operatorname{sinc}(x) = \sin(x)/x$ is the cardinal sinus, and finally,

$$v_\Omega(s) = \Omega^2 u(2\pi\Omega s), \qquad \text{where} \quad u(s) = \operatorname{sinc}(s) - \frac{1}{2}\left(\operatorname{sinc}\left(\frac{s}{2}\right)\right)^2.$$

(b) *Cosine filter.* Here $\widehat{\phi}(\sigma) = \cos(\frac{\sigma\pi}{2})\mathcal{X}_{[0,1]}$ and the corresponding filter is

$$v_\Omega(s) = \frac{\Omega^2}{2}\left(u\left(2\pi\Omega s + \frac{\pi}{2}\right) + u\left(2\pi\Omega s - \frac{\pi}{2}\right)\right), \qquad \text{where } u \text{ is as in (a).}$$

(c) *Shepp-Logan filter.* Now $\widehat{\phi}(\sigma) = \operatorname{sinc}(\frac{\sigma\pi}{2})\mathcal{X}_{[0,1]}$ and

$$v_\Omega(s) = \frac{2\Omega^2}{\pi}u(2\pi\Omega s), \qquad \text{where} \quad u(s) = \begin{cases} \frac{\pi/2 - s\sin s}{(\pi/2)^2 - s^2}, & \text{if } s \ne \pm\pi/2,\\ 1/\pi, & \text{if } s = \pm\pi/2.\end{cases}$$

Discretization of (1). In a first instance the convolution integral of (1) has to be discretized:

$$(v_\Omega * g)(\theta, s) = \int_{\mathbb{R}} v_\Omega(s-t)g(\theta,t)\,dt = \int_{-1}^{1} v_\Omega(s-t)g(\theta,t)\,dt.$$

According to (2), v_Ω has bandwidth Ω, while g as a function of s is essentially bandlimited.

Thus, except for a negligible error (g is only essentially bandlimited), Shannon's Theorem [**2**, Theorem 4.2] can be applied to $f_1(t) = v_\Omega(s-t)$, $f_2(t) = g(\theta, t)$ and the grid $(\Delta s)\mathbb{Z}$, with $\Delta s \leq 1/(2\Omega)$. This yields

$$(v_\Omega * g)(\theta, s) = \Delta s \sum_{l=-q}^{q} v_\Omega(s - s_l)\, g(\theta, s_l). \tag{4}$$

Notice that with our normalization of the Fourier transform the critical density in Shannon's theorem is $1/(2\Omega)$. Next step consists of discretizing the backprojection

$$(V * f)(x) = \mathcal{R}^{\#}(v * g)(x) = \int_0^{2\pi} (v * g)(\theta, x \cdot \theta) d\varphi, \quad \text{where} \;\; \theta = e^{i\varphi}.$$

A computation shows that the π-periodic function $h(\varphi) = (v * g)(\theta, x \cdot \theta)$ has essential bandwidth $4\pi\Omega$, in the sense that

$$\widehat{h}(k) = \frac{1}{2\pi} \int_0^{2\pi} (v * g)(\theta, x \cdot \theta) e^{-ik\varphi} d\varphi$$

is negligible for $|k| > 4\pi\Omega$ [**2**, p.84-85]. Thus we can apply Shannon's theorem [**2**, Theorem 4.2], at the cost of only a negligible error: if $\Delta\varphi \leq 1/(2\Omega)$ we obtain the approximation

$$\begin{aligned}(V * f)(x) = \int_0^{2\pi} (v * g)(\theta, x \cdot \theta) d\varphi &= \frac{\pi}{p} \sum_{j=0}^{2p-1} (v * g)(\theta_j, x \cdot \theta_j) \\ &= \frac{2\pi}{p} \sum_{j=0}^{p-1} (v * g)(\theta_j, x \cdot \theta_j),\end{aligned}$$

where the last identity follows by π-periodicity.

This together with (4), and always taking $\max\{\Delta\varphi, \Delta s\} \leq 1/(2\Omega)$, yields

$$\begin{aligned}(V * f)(x) &= \frac{2\pi}{p} \sum_{j=0}^{p-1} \Delta s \sum_{l=-q}^{q} v_\Omega(x \cdot \theta_j - s_l) g(\theta_j, s_l) \\ &= \frac{2\pi}{p} \Delta s \sum_{j=0}^{p-1} \sum_{l=-q}^{q} v_\Omega(x \cdot \theta_j - s_l) g(\theta_j, s_l).\end{aligned} \tag{5}$$

The algorithm, as given by (5), is computationally too demanding. It requires $O(pq)$ operations for each $f(x)$, and since f has (essential) bandwidth Ω it is necessary to compute $f(x)$ in a lattice with stepsize $1/(2\Omega)$. This gives a total number of operations of order $O(\Omega^2 pq) \simeq O(\Omega^4)$. This complexity can be reduced with a linear interpolation.

Since $v_\Omega * g$ has bandwidth Ω it is determined by $(v_\Omega * g)(\theta_j, s_l)$, which can be computed with $O(pq^2)$ operations. Then the values $(v_\Omega * g)(\theta_j, x\cdot\theta_j)$ required to compute $V * f$ are obtained from the previous ones by linear interpolation. This reduces the number of operations to $O(\Omega^3)$.

Final algorithm

Step 1. For every direction θ_j, $j = 1, \ldots, p$ take the discrete convolution

$$h_{j,k} = \Delta s \sum_{l=-q}^{q} v_\Omega(s_k - s_l)\, g_{j,l} \qquad (k = -q, \ldots, q).$$

Step 2. For each x compute the discrete backprojection using a linear interpolation of the values obtained in Step 1:

$$f_A(x) = \frac{2\pi}{p} \sum_{j=0}^{p-1} (1-\eta) h_{j,k} + \eta h_{j,k+1},$$

where $k = k(j,x) = \left\lfloor \frac{x\cdot\theta_j}{\Delta s} \right\rfloor$, $\eta = \eta(j,x) = \frac{x\cdot\theta_j}{\Delta s} - \left\lfloor \frac{x\cdot\theta_j}{\Delta s} \right\rfloor$ and $\lfloor a \rfloor$ denotes the integer part of a.

Bibliography

[1] B. F. Logan and L. A. Shepp. Optimal reconstruction of a function from its projections. *Duke Math. J.*, 42(4):645–659, 1975.

[2] F. Natterer. *The Mathematics of Computerized Tomography.* Classics in Applied Mathematics 32, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2001.

[3] C. Cascante, X. Massaneda, J. Ortega-Cerdà and D. Pascuas Computerized Tomography and the Radon Transform. *Boletín de la SEMA*, 41, 2007.