Exercises

Exercise 8.1. Select an image array (x_i, y_j), that is, for a set $x_i = i\Delta x$ and $y_j = j\Delta y$, we want to compute the image values $f_{i,j} \equiv f(x_i, y_j)$ at (x_i, y_j) (for a more precise definition, see (74)): the values for $f_{i,j}$ that we obtain by computation form the 'reconstructed' or 'output' image.

We suppose that the projected values $p_\phi(\xi)$ are available (can be measured) for a set of angles $\phi = \phi_k = k\Delta\phi$ and, for each ϕ in this set, for a set of points $\xi = \xi_\ell = \ell\Delta\xi$.

Alg. 1 uses this input information to reconstruct the image.

> *A. Initialize the output image to 0 ($f_{i,j} = 0$ for all i, j).*
>
> *B. For each angle ϕ in the collection $\{\phi_k\}$ do:*
>
> *1. 'Measure' the projections $p_\phi(\xi)$ at the points $\xi = \xi_\ell$.*
>
> *2. Use FFT to compute $\widehat{p}_\phi(\rho)$ (see (72)).*
>
> *3. Use inverse FFT to compute $\widetilde{p}_\phi(\xi)$ at the points $\xi = \xi_\ell$ (see (72)).*
>
> *4. Use the results for this one projection to update the output image (see (73)), that is, for each (x, y) in the set (x_i, y_j) do:*
>
> *a. Determine ξ from (68).*
>
> *b. Approximate the modified projection at ξ by interpolation of the $p_\phi(\xi_\ell)$ (linear interpolation is usually sufficient).*
>
> *c. Add this contribution to the ϕ integral (73) to the image.*
>
> *5. Display the image obtained thus far (optional).*

ALGORITHM 1. An outline of an image reconstruction algorithm based on the theory in §8.A.

In practice, of course, the projections in step 1 would be obtained from the CT scanning device. For our purposes, we compute them from an "input" image.[22] The reconstructed image is the output image.

Use the outline in Alg. 1 to write a computer program to reconstruct images.[23] You may assume that the images are contained in a circle of unit radius.[24]

Investigate the effect of changing the resolution (i.e., the size of Δx and Δy. You may take $\Delta x = \Delta y$. Consider, for instance $\Delta x = 2/100, 2/200, \ldots$). Do you need the same step sizes Δx and Δy for the reconstruction as for the projections? How do you relate the step size Δx and Δy for reconstruction to the step size $\Delta\xi$ and $\Delta\phi$?

Discuss the advantages of this approach with one dimensional inverse Fourier transforms followed by an integral over ϕ as compared to an approach based on one two dimensional inverse Fourier transform.

What do you expect from replacing linear interpolation in step 4b by cubic interpolation? (i.e., approximate the value $\widetilde{p}_\phi(\xi)$ by the value $P(\xi)$ of the cubic polynomial P for which $P(\xi_{\ell-1}) = \widetilde{p}_\phi(\xi_{\ell-1})$, $P(\xi_\ell) = \widetilde{p}_\phi(\xi_\ell)$, $P(\xi_{\ell+1}) = \widetilde{p}_\phi(\xi_{\ell+1})$, $P(\xi_{\ell+2}) = \widetilde{p}_\phi(\xi_{\ell+2})$, where ℓ is such that $\xi \in (\xi_\ell, \xi_{\ell+1})$. How do you define P near the boundary?).

For more details, we refer to [6, Ch.3].

---

[22] The code for computing the projections will be provided for.

[23] You can use existing codes for FFT and inverse FFT.

[24] The MATLAB file `Images.m` can be used for producing images. The file gives several ways of producing image files that can be used in testing the image reconstruction code.

Discretization. A picture of which each pixel is either black or white seems to be composed of patches of different shades of gray when viewed from a distance. When in an approximation with larger pixels the color is determined by sampling the shades of gray disappear and the black and white becomes visible. Discretization by averaging seems to be a more appropriate approach here: specifically in image processing, discretization by averaging leads to pictures that are more appealing to the eye, then when discretized by sampling.

If the real-valued function f that is defined on the square $[-a,a]\times[-a,a]$ represents an picture,[25] then f can be approximated as follows by a piece-wise constant function. Select a step size Δx in the x-direction and a step size Δy in the y direction ($\Delta x = 2a/N_x$ and $\Delta y = 2a/N_y$). Then, $(x_i, y_j) \equiv ((i-\frac{1}{2})\Delta x, (j-\frac{1}{2})\Delta y) - (a,a)$ is the center of the pixel area $I_{i,j} \equiv [x_i - \frac{1}{2}\Delta x, x_i + \frac{1}{2}\Delta x) \times [y_j - \frac{1}{2}\Delta y, y_j + \frac{1}{2}\Delta y)$ $(i = 1,\ldots,N_x, j = 1,\ldots,N_y)$, and we compute the image value $f_{i,j}$ for this pixel by averaging,

$$f_{i,j} \equiv \frac{1}{\Delta x\,\Delta y}\iint_{I_{i,j}} f(x,y)\;\mathrm{d}x\;\mathrm{d}y : \tag{74}$$

f is approximated by the function f_d that has the value $f_{i,j}$ on $I_{i,j}$ $(i = 1,\ldots,N_x, j = 1,\ldots,N_y)$.

This approach suggests the following discretization for the projection $p_\phi(\xi)$. Select a step size $\Delta\xi = 2/(N_\xi - 1)$. Define

$$p_{\ell,\phi} \equiv \frac{1}{\Delta\xi}\int_{\xi_\ell-\frac{1}{2}\Delta\xi}^{\xi_\ell+\frac{1}{2}\Delta\xi} p_\phi(\xi)\;\mathrm{d}\xi \qquad (\ell = 1,\ldots,N_\xi), \tag{75}$$

where $\xi_\ell \equiv -1 + (\ell-1)\Delta\xi$.

Computing the projections. Note that p_ϕ is the sum of all projected pixels. To be more precise, if $\chi_{i,j}$ is the function with value 1 on $I_{i,j}$ and zero elsewhere, then

$$p_\phi(\xi) = \sum_{i,j} f_{i,j}\int \chi_{i,j}(x(\eta), y(\eta))\;\mathrm{d}\eta. \tag{76}$$

Here, we sum over all $i = 1,\ldots,N_x$ and all $j = 1,\ldots,N_y$, and $x(\eta)$ and $y(\eta)$ are as defined in §8.1. Except for the color value $f_{i,j}$, the function

$$\pi_\phi(\xi) \equiv \int \chi_{i,j}(x(\eta), y(\eta))\;\mathrm{d}\eta$$

is the projected pixel. To simplify notation, we surpress the indices i and j here. For a graph of this function, see Fig. 28. Note that the integral of π_ϕ over ξ is precisely equal to $\Delta x\,\Delta y$ (why?). The contribution of the (i,j)th pixel to $p_{\ell,\phi}$ is equal to

$$f_{i,j}\,\frac{1}{\Delta\xi}\int_{\xi_\ell-\frac{1}{2}\Delta\xi}^{\xi_\ell+\frac{1}{2}\Delta\xi} \pi_\phi(\xi)\;\mathrm{d}\xi.$$

[25] We assume the picture to be defined on a square contained in the unit disk $\{(x,y)\,\big|\,|x|^2+|y|^2 \le 1\}$, that is, $a \equiv \frac{1}{2}\sqrt{2}$. Obviously, this is not a restriction, since each picture can be scaled to this situation. Working on the unit disk slightly simplifies the coding when using polar coordinates.

For a full color picture, we need three color functions f_r, f_g, and f_b, where, for instance, f_r defines the red intensities, f_g the green intensities, and f_b the blue intensities. However, in CT the colors are artificial anyway. We, therefore, assume the image to be defined by one color function f only. The function value $f(x,y)$ represent the intensity of the color at (x,y).

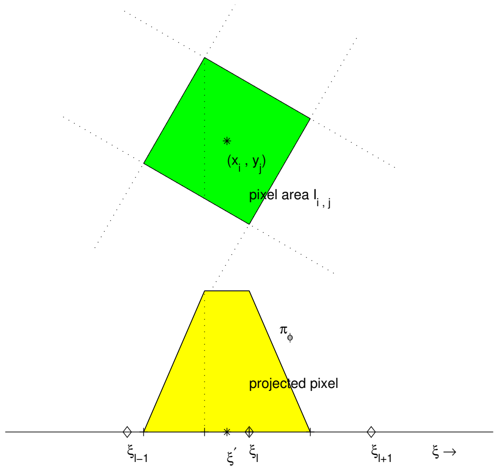

FIGURE 28. The picture represents a pixel (with center (x_i, y_j)) and its projection π_ϕ on a ξ-axis. The projection is on the line with angle ϕ with the x axis (ϕ is here $-\pi/3 = 2\pi/3$). Rather than rotating the projection line over an angle ϕ, the pixel area has been rotated over an angle $-\phi$ in this picture.

If π_ϕ is zero outside the interval $\xi_\ell + [-\frac{1}{2}\Delta\xi, \frac{1}{2}\Delta\xi]$, then this contribution is precisely equal to $f_{i,j}\,\frac{\Delta x\,\Delta y}{\Delta\xi}$. Of course, it depends on the diameter of the pixel (on Δx and Δy) and on $\Delta\xi$, but, in general, π_ϕ will not be non-zero on one interval only.[26] To speed up computations, we will not compute what part of π_ϕ is non-zero on each of the intervals $\xi_\ell + [-\frac{1}{2}\Delta\xi, \frac{1}{2}\Delta\xi]$, but, we will simply follow the following strategy.

If ξ' is the ξ-coordinate of the center (x_i, y_j) of the (i, j)th pixel (i.e., $x_i = \xi' c - \eta' s$, $y_j = \xi' s + \eta' c$ where $c \equiv \cos(\phi)$ and $s \equiv \sin(\phi)$), and ℓ is such that $\xi' \in [\xi_\ell, \xi_{\ell+1})$, then we add the fraction $f_{i,j}\,\frac{\Delta x\,\Delta y}{\Delta\xi}\,\frac{\xi'-\xi_\ell}{\xi_{\ell+1}-\xi_\ell}$ to $p_{\ell+1,\phi}$ and the remaining fraction $f_{i,j}\,\frac{\Delta x\,\Delta y}{\Delta\xi}\,\frac{\xi_{\ell+1}-\xi'}{\xi_{\ell+1}-\xi_\ell}$ to $p_{\ell,\phi}$: we follow a 'linear interpolation' type of strategy.

[26]If $\Delta x + \Delta y \ll \Delta\xi$ then most of the π_ϕ will be non-zero on only one interval. However, certainly in the reconstruction phase, the pixels will be selected such that their diameters are comparable to $\Delta\xi$. Then, π_ϕ will generally be non-zero on two intervals.

References

[1] Juan Arias de Reyna. *Pointwise convergence of Fourier series*, volume 1785 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2002.

[2] D. C. Champeney. *Fourier transforms and their physical applications*. Academic Press, London, New-York, 1973.

[3] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.

[4] Paul L. DeVries. *A first course in computational Physics*. Wiley & sons, 1994.

[5] R. E. Edwards. *Fourier Series, a modern introduction*. Holt, Rinehart and Winston, Inc., New York, etc., 1967.

[6] A. C. Kak and Malcolm Slaney. *Principles of Computerized Tomographic Imaging*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[7] A. Papoulis. *The Fourier Integral and its Applications*. McGraw-Hill Book Company, 1962.

[8] A. Papoulis. *Signal Analysis*. McGraw-Hill Book Company, 1977.

[9] J. Price, editor. *Fourier techniques and applications*, New York, 1985. Plenum Press.

[10] P. L. Walker. *The Theory of Fourier Series and Integrals*. Wiley, Chichester, 1986.