USING MATLAB FOR SELECT APPLIED PROBLEMS

Mark Ioffe, Ph.D.

Abstract

Matlab software is used for solving across a wide spectrum of applied problems and has a specific character. This article presents a basic mathematical formulas and Matlab functions for computer aided tomography, when the process of reconstructing an image from projection through the image is simulated.

Basic mathematical formulas.

The tomography problem is to reconstruct a two-dimensional image, given by function f(x,y), from a set of projections through the image at various angles in the interval [0,π]. A projection p(t,θ) is a function of 2 variables generated by calculating line integrals of f(x,y) along parallel lines that pass through the image. Mathematically, function p(t,θ) is known Radon transformation of function f(x,y), i.e.

$$p(t,\theta) = \int_{-\infty}^{\infty} f(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)ds$$

(1)

The image is assumed to lie inside the square two unit on side, that is
f(x,y)=0 for x<-1 OR x>1 OR y<-1 OR y>1.

Let function F(u,v) is 2D Fourier transform of f(x,y), i.e.

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{i(xu+yv)} f(x,y)dxdy$$

(2)

And

$$f(x,y) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-i(xu+yv)} F(u,v)dudv$$

(3)

Accordingly (1) we have

$$p(t,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - t)dxdy$$

(4)

Where δ(ω) is Dirac function which is inverse Fourier transform of function $e^{-\omega z}$, i.e.

$$\delta(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-\omega z} dz$$

(5)

Accordingly (4) and (5)

$$p(t,\theta) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{itz} dz \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-i(x\cos\theta + y\sin\theta)z} dxdy$$

(6)

Using property of 1D Fourier transform we receive from (6)

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-i(x\cos\theta + y\sin\theta)z} dxdy = \int_{-\infty}^{\infty} p(t,\theta) e^{itz} dt$$

(7)

Accordingly (2) and (7)

$$F(z\cos\theta, z\sin\theta) = \int_{-\infty}^{\infty} p(t,\theta) e^{itz} dt$$

(8)

Switch to quasi polar coordinates $-\infty < z < \infty$ and $0 \le \theta < \pi$ in formula (3)

$$u = z\cos\theta$$
$$v = z\sin\theta$$

(9)

Because Jacobian equals $|z|$ formula (3) gives

$$f(x,y) = \frac{1}{4\pi^2} \int_0^{\pi} \int_{-\infty}^{\infty} F(z\cos\theta, z\sin\theta) |z| e^{-iz(x\cos\theta + y\sin\theta)} dz d\theta$$

(10)

Accordingly (8) and (10)

$$f(x,y) = \frac{1}{4\pi^2} \int_0^{\pi} d\theta \int_{-\infty}^{\infty} |z| e^{-iz(x\cos\theta + y\sin\theta)} dz \int_{-\infty}^{\infty} p(t,\theta) e^{izt} dt$$

(11)

Using (11) we can theoretically calculate function f(x,y) for given $p(t,\theta)$

Calculations and Matlab functions.

A tomography numerical algorithm takes a set projection data {p(t(k),θ(j)}, j,k=1….n, (a “shadow”) as input and calculates a reconstructed image f(x(i),y(l)), i,l=1,…n. We assume that x, y ,t, θ are spread evenly over rectangular net, (grid)

$$x(i) = -1 + \frac{2(i-1)}{n-1}$$
$$y(l) = -1 + \frac{2(l-1)}{n-1}$$
$$t(k) = -\sqrt{2} + \frac{2\sqrt{2}(k-1)}{(n-1)}$$
$$t(j) = -\sqrt{2} + \frac{2\sqrt{2}(j-1)}{(n-1)}$$

(12)

One could say that tomography numerical algorithm transforms shadow matrix p={p(k,j), k,j=1,…n} into image matrix f={f(i,l), i,l=1…n}. Inversely, numerical implementation of Radon transformation, formula (1), is calculation of matrix p for given matrix f.
First, we consider Matlab implementation of Radon transformation.
Five Matlab functions calculates shadow matrix p for given image matrix f: numextrem.m, ExtremumS.m, radon0.m, radonfull.m and dblsrch.m.
Function numextrem.m calculates number of points i that line with parameters t and θ, $x = t\cos\theta - s\sin\theta, y = t\sin\theta + s\cos\theta$, intersects the square two unit on side $-1 \le x \le 1, -1 \le y \le 1$.
Obviously, i=0,1,2. If i=0 or 1 then p=0.
Function ExtremumS.m calculates intersection the square point’s coordinates x,y and parameters s.
Function radon0.m determines a set of square’s cell, i(k),l(k), k=1,..m through which line pass and calculates projection p by formula

$$p(t,\theta) = \sum_{k=1}^{m-1} f(i(k),l(k))(s(k+1) - s(k))$$

(13)

Function radonfull.m calculates matrix p(k,j), k,j=1,…n by using function radon0.m for any given t and θ.
Function dblsrch.m is auxiliary. It implemented a known algorithm of double search.
Second, we consider Matlab implementation of tomography numerical algorithm, based only on formula (11). The input information is matrix p(k,j), k,j=1,…n. We suppose that function p(t, θ) is step function, i.e.

$$p(t,\theta) = \sum_{k=1}^{n-1} p(k,\theta) 1(t_k, t_{k+1})$$

(14)

Function $1(t_k, t_{k+1})$ equals 1 if $t_k \le t < t_{k+1}$ and 0 else.

Then

EGAR technology
307 East 53rd Street, 6th Floor
New York, NY 10022
By Phone: 212-223-3552
By Fax: 212-421-6608
By Email: info@egartech.com
www.egartech.com

$$\int_{-\infty}^{\infty} p(t,\theta)e^{izt}dt = \sum_{k=1}^{n-1} p(k,\theta)\frac{e^{izt_{k+1}} - e^{izt_k}}{iz} \approx \tau\sum_{k=1}^{n-1} p(k,\theta)e^{izt_k}$$
$$\tau = t_{k+1} - t_k$$

(15)

We suppose that

$$-W < z < W$$
$$W = \frac{\pi}{\tau}$$

(16)

W=maximum value of parameters x.

From (11) and (15) we get

$$f(x,y) = \frac{1}{2\pi}\int_0^{\pi}\sum_{k=1}^{n-1} p(k,\theta)\left(\frac{\sin(\beta W)}{\beta} + \frac{\cos(\beta W) - 1}{\beta^2 W}\right)$$
$$\beta = x\cos\theta + y\sin\theta - t_k$$

(17)

Using for calculation of integral method of rectangles we have finally

$$f(x,y) = \frac{del\theta}{2\pi}\sum_{k=1}^{n-1}\sum_{j=1}^{n-1} p(k,j)\left(\frac{\sin(\beta(k,j)W)}{\beta(k,j)} + \frac{\cos(\beta(k,j)W) - 1}{\beta(k,j)^2 W}\right)$$
$$\beta(k,j) = x\cos\theta_j + y\sin\theta_j - t_k$$
$$del\theta = \frac{\pi}{n-1}$$

(18)

Matlab function invradon01.m implemented (18), i.e. calculates inverse Radon transform for one point (x,y). Matlab function invradon1.m calculates inverse Radon transform for square $-1 \le x \le 1, -1 \le y \le 1$.

Numerical illustration (Matlab algorithms tests).

A common method for testing tomography algorithms is to construct artificial images based on case when Radon transform can be calculated analytically, exactly. We consider two samples. First test image is generated by superimposing 10 ellipses of different brightness [1]. Shepp and Logan were the first to use this image for testing tomography algorithms. We use Shepp and Logan sample by two methods. In first method Matlab function SheppLoganDataRadon.m calculates analytically Radon transform of 10 ellipses image. This function uses ellipsradon.m. Function ellipsradon.m calculates integral (1) for any t and θ and any ellipse, specified by giving its center coordinates (x1,y1) , the length A of semi-major axis, the length B of semi-minor axis, the orientation α of the ellipse, and a grey-scale value λ for by the brightness of the ellipse. Calculation based on solution of quadratic equation for variable s. Vector-matrix form for this equation is

$$(t,s)T(\theta-\alpha)DT(\theta-\alpha)'(t,s)'-2(t,s)T(\theta-\alpha)D(\xi_0,\varsigma_0)'+(\xi_0,\varsigma_0)D(\xi_0,\varsigma_0)'-1=0$$

$$T(\theta-\alpha)=\begin{pmatrix}\cos(\theta-\alpha) & \sin(\theta-\alpha)\\ -\sin(\theta-\alpha & \cos(\theta-\alpha)\end{pmatrix}$$

$$D=\begin{pmatrix}\frac{1}{A^2} & 0\\ 0 & \frac{1}{B^2}\end{pmatrix}$$

$$(\xi_0,\varsigma_0)=(x_1,y_1)\begin{pmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{pmatrix}$$

(19)

In formulas (19) $'$ is transposition sign.
In second method for calculation Radon transform we use our Matlab function radonfull.m. Input for Radon transform of 10 ellipses image, matrix D, calculates Matlab function SheppLoganData.m. For both methods we calculate reconstructed image, inverse Radon transform, by using Matlab function invradon01.m.
Second test image is generated by the brightness described function f(x,y) with parameter σ:

$$f(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{1}{2\sigma^2}(x^2+y^2)}$$

(20)

Strictly speaking f(x,y)≠0 for x<-1 OR x>1 OR y<-1 OR y>1. But if $\sigma\leq 0.26115$ then
|f(x,y)|<0.000001 for
for x<-1 OR x>1 OR y<-1 OR y>1. We will analyze only.
Radon transformation of function f(x,y) function p(t,θ) equals:

$$p(t,\theta)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2\sigma^2}t^2}$$

(21)

Fourier transform of p(t,θ) is

$$\int_{-\infty}^{\infty}p(t,\theta)e^{izt}dt=e^{-\frac{1}{2}\sigma^2z^2}$$

(22)

Accordingly (11)

$$f(x,y)=\frac{1}{2\pi^2}\int_0^{\pi}d\theta\int_0^{\infty}z\cos(zt)e^{-\frac{1}{2}\sigma^2z^2}dz$$

(23)

Let

$$F(t)=\int_0^\infty z\cos(zt)e^{-\frac{1}{2}z^2}dz$$

(24)

Obviously

$$F(t)=1-t\int_0^\infty \sin(zt)e^{-\frac{1}{2}z^2}dz$$

(25)

It is easy to see that

$$\int_0^\infty \sin(zt)e^{-\frac{1}{2}z^2}dz=e^{-\frac{1}{2}t^2}\int_0^t e^{\frac{1}{2}u^2}du$$

(26)

Obviously

$$\int_0^\infty z\cos(zt)e^{-\frac{1}{2}\sigma^2 z^2}dz=\frac{1}{\sigma^2}F\left(\frac{t}{\sigma}\right)$$

(27)

By substituting (25), (26) and (27) into (23) we get

$$f(x,y)=\frac{1}{2\pi^2}\int_0^\pi (1-\frac{t}{\sigma^3}e^{-\frac{1}{2\sigma^2}t^2}\int_0^{\frac{t}{\sigma}} e^{\frac{1}{2}u^2}du)d\theta$$

$$t=x\cos\theta+y\sin\theta$$

(27)

Formula (27) can be used for modeling tomography algorithms for image with brightness described function (20).

For results's of calculation visualization we use Matlab function IMAGESC(C). This function displays matrix C as an image. Each element of C specifies the color of a rectilinear patch in the image. The results of calculation are shown in five Matlab graphic files:

D.fig-Shepp-Logan phantom, generated by superimposed ellipses;

PPP1.fig-Shepp-Logan reconstruction, generated by shadow, calculated analytically;

PPP.fig- Shepp-Logan reconstruction, generated by shadow, calculated by using numerical Radon transform;

G.fig-Gauss phantom, generated by function (20);

G1.fig- Gauss reconstruction, generated by shadow, calculated analytically.

References

[1] Hugh Murrel, 1996. Computer-Aided Tomography. The Mathematica Journal. Volume 6, Issue 2.