Subject: Re: backprojection
Posted by Mike[2] on Thu, 18 Jan 2007 16:49:10 GMT
View Forum Message <> Reply to Message

TimLS wrote:
> I got the radon transform to work for me. The link to Mark River's
> web-site was very interesting and useful.

Don't forget that the inverse radon transform is not the inverse of forward projection. The filtered backprojection is the inverse. Filtered backprojection generally proceeds by four steps: 1) calculate the Fourier transform of the projection data, 2) apply the appropriate filter, 3) calculate the inverse Fourier transform to get filtered projections, 4) backproject the filtered projections to get an image.

Here's a bit of code that I pulled out of our production code for PET reconstructions. Some of it is specific to our geometry, so you'd have to include your own sampling geometry. The input to this would be sgplane, an Nprojections x Nangles array of projection data.

```
Nangles = 768
Nprojections = 128
proj_spacing = 0.1
deltakx = 1.0 / (Nprojections * proj_spacing)

;; Nyquist frequency, in per cm, from default proj spacing of detector
;; banks
nyfreq = 1/(2.0*proj_spacing/10.0)

cutfreq    = cutperc/100. * nyfreq

;; ---- Make a ramp filter
; /10.0 since projspacing is in mm and we want kx in per cm
deltakx = 1.0 / (Nprojections * projspacing/10.0)

; array of kx values with kx=0 at center
kxvals = ( FINDGEN(Nprojections) - (Nprojections-1)/2 ) * deltakx

; shift to agree with IDL ordering, kx=0 at i=0, etc.
kxvals = SHIFT(kxvals,-(Nprojections-1)/2)
ft_filter = COMPLEXARR(Nprojections)
FOR i=0,(Nprojections-1) DO BEGIN
  kx = kxvals[i]
  ft_filter[i] = COMPLEX(!PI*ABS(kx),0.0)
ENDFOR
```

```
;; apply the filter to each row of the sinogram
FOR a = 0,Nangles-1 DO BEGIN
  row = sgplane[*,a]
  ftrow = FFT(row)
  ftfilteredrow = ft_filter * ftrow
  filteredrow = FFT(ftfilteredrow, /INVERSE)
  sgplane[*,a] = filteredrow
ENDFOR

projs  = ( FINDGEN(Nprojections) - proj_center ) * projspacing

;; parameters for backprojection
Npix = 128
angles = FINDGEN(Nangles) * !PI / FLOAT(Nangles) + rotation * !DTOR
pixsz  = fov / FLOAT(Npix)
xmin   = (-fov/2.0) - xoff
ymin   = (-fov/2.0) - yoff

image = RADON(TRANSPOSE(sgplane), /BACKPROJECT, /LINEAR, $
            RHO=projs, THETA=angles, $
            DX=pixsz, DY=pixsz, $
            NX=Npix, NY=Npix, $
            XMIN=xmin, YMIN=ymin)
```

Regards, Mike Miller, Imaging Sciences, IU School of Medicine

P.S. A good reference for tomography is Kak and Slaney's
"Principles of Computerized Tomographic Imaging," which is
available for download at http://www.slaney.org/pct/
