Subject: Re: backprojection
Posted by David Fanning on Wed, 17 Jan 2007 15:11:00 GMT
View Forum Message <> Reply to Message

TimLS writes:

> I have some radar data which I want to process. I have range profiles
> (back scattered power is recorded at different ranges) that have been
> collected at 1 degree increments all the way round an object that I am
> trying to image. This must be very similar to a lot of medical imaging
> applications. I next want to use something like the inverse radon
> transform to transform the r, theta data into an x-y type image of the
> object. Is there anyone out there in the medical imaging world who has
> some code or who can give me a clue as to what I need to do?

Have you looked at the HOUGH and RADON functions and the
examples in the documentation for back projection?

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: backprojection
Posted by Wox on Wed, 17 Jan 2007 16:32:28 GMT
View Forum Message <> Reply to Message

On 17 Jan 2007 07:00:22 -0800, "TimLS" <tlsmith@qinetiq.com> wrote:

> Is there anyone out there in the medical imaging world who has
> some code or who can give me a clue as to what I need to do?

Try the code below. The function 'createsinogram' makes a sinogram.
testtomo uses the radon function for reconstruction as Mr. Fanning
suggested.

Some things you have to supply to radon:
1. projcen: the center of projection (i.e. the projection of the
center of rotation)
2. angleinc: rotation increment

The first might be tricky. For info on how to extract the center of
projection from a sinogram:
http://www-fp.mcs.anl.gov/xray-cmt/rivers/tutorial.html
They have also some IDL code there, including filters on the sinogram
to improve the reconstructed image.

Remark: Notice that in the example, the center of projection is not in
the middle. The reconstruction is not very good because the object
falls out of the detection area during rotation. Using filter (as
metioned above) may improve images like that.

Anyway, hope this helps.

```
function createsinogram
; 1. Create an image with rings plus random noise:
N=250
M=250
R0=(N<M)*0.1
R1=(N<M)*0.12
xc=N/2.+20
yc=M/2.
x = (LINDGEN(N,M) MOD N)
y = (LINDGEN(N,M)/N)
radius = SQRT((x-50)^2 + (y-80)^2)
realslice = (radius GT R0) AND (radius LT R1)
radius = SQRT((x-40)^2 + (y-100)^2)
realslice OR= (radius GT R0) AND (radius LT R1)
radius = SQRT((x-60)^2 + (y-100)^2)
R0=0
realslice OR= (radius GT R0) AND (radius LT R1)
realslice += RANDOMU(seed,N,M)

realslice[xc-1:xc+1,yc-1:yc+1] = 2*max(realslice)

tvscl,realslice,0, 0, /NORMAL

; 2. Create sinogram
sinogram = RADON(realslice,xmin=-xc,ymin=-yc,drho=1,NRHO=N,RMIN=-xc)
tvscl,sinogram,0, 0.5, /NORMAL

return,sinogram
end;function createsinogram


pro testtomo
```

```
sinogram=createsinogram()
s=size(sinogram)
N=s[2]
NAngles=s[1]

projcen=N/2.+20
angleinc=180./NAngles
anglestart=0.

angleinc*=!pi/180
anglestart*=!pi/180
angles=anglestart+angleinc*findgen(NAngles)
rho=findgen(N)-projcen
 tomogram=radon(sinogram,/BACKPROJECT,rho=rho,theta=angles,/L
INEAR,nx=N,ny=N,xmin=-projcen)

tvscl,tomogram,0.5, 0, /NORMAL
end
```

Subject: Re: backprojection
Posted by TimLS on Thu, 18 Jan 2007 08:40:04 GMT
View Forum Message <> Reply to Message

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

Thanks, Tim

Wox wrote:
> On 17 Jan 2007 07:00:22 -0800, "TimLS" <tlsmith@qinetiq.com> wrote:
>
>> Is there anyone out there in the medical imaging world who has
>> some code or who can give me a clue as to what I need to do?
>
>
> Try the code below. The function 'createsinogram' makes a sinogram.
> testtomo uses the radon function for reconstruction as Mr. Fanning
> suggested.
>
> Some things you have to supply to radon:
> 1. projcen: the center of projection (i.e. the projection of the
> center of rotation)
> 2. angleinc: rotation increment
>
> The first might be tricky. For info on how to extract the center of
> projection from a sinogram:

> http://www-fp.mcs.anl.gov/xray-cmt/rivers/tutorial.html
> They have also some IDL code there, including filters on the sinogram
> to improve the reconstructed image.
>
> Remark: Notice that in the example, the center of projection is not in
> the middle. The reconstruction is not very good because the object
> falls out of the detection area during rotation. Using filter (as
> metioned above) may improve images like that.
>
> Anyway, hope this helps.
>
>
>
>
> function createsinogram
> ; 1. Create an image with rings plus random noise:
> N=250
> M=250
> R0=(N<M)*0.1
> R1=(N<M)*0.12
> xc=N/2.+20
> yc=M/2.
> x = (LINDGEN(N,M) MOD N)
> y = (LINDGEN(N,M)/N)
> radius = SQRT((x-50)^2 + (y-80)^2)
> realslice = (radius GT R0) AND (radius LT R1)
> radius = SQRT((x-40)^2 + (y-100)^2)
> realslice OR= (radius GT R0) AND (radius LT R1)
> radius = SQRT((x-60)^2 + (y-100)^2)
> R0=0
> realslice OR= (radius GT R0) AND (radius LT R1)
> realslice += RANDOMU(seed,N,M)
>
> realslice[xc-1:xc+1,yc-1:yc+1] = 2*max(realslice)
>
> tvscl,realslice,0, 0, /NORMAL
>
> ; 2. Create sinogram
> sinogram = RADON(realslice,xmin=-xc,ymin=-yc,drho=1,NRHO=N,RMIN=-xc)
> tvscl,sinogram,0, 0.5, /NORMAL
>
> return,sinogram
> end;function createsinogram
>
>
> pro testtomo
>
> sinogram=createsinogram()

```
>  s=size(sinogram)
>  N=s[2]
>  NAngles=s[1]
>
>  projcen=N/2.+20
>  angleinc=180./NAngles
>  anglestart=0.
>
>  angleinc*=!pi/180
>  anglestart*=!pi/180
>  angles=anglestart+angleinc*findgen(NAngles)
>  rho=findgen(N)-projcen
>   tomogram=radon(sinogram,/BACKPROJECT,rho=rho,theta=angles,/L
INEAR,nx=N,ny=N,xmin=-projcen)
>
>  tvscl,tomogram,0.5, 0, /NORMAL
>  end
```

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/

Subject: Re: backprojection
Posted by Marc Reinig on Fri, 19 Jan 2007 01:53:12 GMT
View Forum Message <> Reply to Message

"David Fanning" <news@dfanning.com> wrote in message
news:MPG.2017ea6ce320271a989e7b@news.frii.com...
> TimLS writes:
>
>> I have some radar data which I want to process. I have range profiles
>> (back scattered power is recorded at different ranges) that have been
>> collected at 1 degree increments all the way round an object that I am
>> trying to image. This must be very similar to a lot of medical imaging
>> applications. I next want to use something like the inverse radon
>> transform to transform the r, theta data into an x-y type image of the
>> object. Is there anyone out there in the medical imaging world who has
>> some code or who can give me a clue as to what I need to do?
>
> Have you looked at the HOUGH and RADON functions and the
> examples in the documentation for back projection?
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Look up filtered backprojection and the Fourier slice theorem. Filtering
the back projected data compensates for the fact that the some of the data
is sampled more than others and needs to be de emphasized.

-Marco

_____ ____
Marc Reinig
Laboratory for Adaptive Optics
UCO/Lick Observatory

Subject: Re: backprojection
Posted by Wox on Fri, 19 Jan 2007 09:52:32 GMT
View Forum Message <> Reply to Message

On 18 Jan 2007 08:49:10 -0800, "Mike" <Michael.Miller5@gmail.com>
wrote:

> 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.

<snip>

> ;; 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

<snip>

To clarify: the filter shown is similar to the RamLak filtering in
real space (using convolution instead of multiplication) on Mark
River's web-site:
http://www-fp.mcs.anl.gov/xray-cmt/tomo_filter.htm