Subject: RADON and BACKPROJECT
Posted by jcullom on Tue, 20 Jan 2004 07:22:17 GMT
View Forum Message <> Reply to Message

Does anyone have experience defining the rho and theta variables in
the BACKPROJECT operation ? It wants this to be a vector of
coordinates, but the RADON operation seems to define them. Any ideas
what it wants ? i.e, if I have 64 angles nad 64 pixels ina projection,
what values should these variables be ?

Thanks,
JCKC

Subject: Re: RADON and BACKPROJECT
Posted by rigby on Tue, 20 Jan 2004 16:02:51 GMT
View Forum Message <> Reply to Message

jcullom@cvit-online.com wrote in message
news:<34e9572f.0401192322.75ae54e8@posting.google.com>...
> Does anyone have experience defining the rho and theta variables in
> the BACKPROJECT operation ? It wants this to be a vector of
> coordinates, but the RADON operation seems to define them. Any ideas
> what it wants ? i.e, if I have 64 angles nad 64 pixels ina projection,
> what values should these variables be ?
>
> Thanks,
> JCKC

Assuming you generate the projections using Radon(), I think you need
to capture the RHO and THETA values Radon() generates and then pass
them back in for the backprojection call.

The code below is basically what I came up with to use non-default
values with Radon() -- perhaps it will be helpful.

Good luck!
Wayne

```
PRO Radon_Outline

;; example input array
nx = 64
ny = 32
ph = fltarr(nx,ny)
ph[*,10] = 1.0
ph[30,*] = 2.0
```

```
;; Radon() parameters
dx = 1.0   ;; unit of input array -- don't change these
dy = 1.0   ;; never tried (dy NE dx)

default_ntheta = ceil(!pi * sqrt(0.5*(nx^2+ ny^2)) )

default_drho = 0.5* sqrt(dx^2 + dx^2)
;; half the pixel diagonal -- this is Radon()'s default
;; documentation says DHRO can be smaller than the default,
;; shouldn't be larger.

;; origin is the center of the image (empirical)
if (Even(nx) and Even(ny)) then begin
   xmax = 0.5 * (nx-1) * dx  ;; empirical
   ymax = 0.5 * (ny-1) * dy
endif else stop    ;; never tried nx, ny odd

default_maxrho = sqrt(xmax^2 + ymax^2)  ;; corner of input
default_nrho = 1 + fix(2 * ceil( default_maxrho / default_drho ))
;; odd, but the documentation doesn't say it must be

; This seems to be the right way to do it -- specify DRHO, NRHO and
NTHETA.
; Increasing NRHO by itself just gives a larger *range* of RHO. I want
finer
; sampling in the range direction.
scale_rho = 2
drho = default_drho/scale_rho

nrho = fix(1.3*default_nrho)  ;; pad a bit over the default
if (Odd(nrho)) then nrho = nrho + 1  ;; I want NRHO even

scale_theta = 2
ntheta = default_ntheta * scale_theta

print, "input:"
help, ph
print, "nx = ", nx
print, "ny = ", ny

ph_radon = Radon(ph, linear=doLinearInterp, $
                 drho=drho, nrho = nrho, $
                 ntheta=ntheta, $
                 rho=rho, theta=theta)

print, ""
print, "radon transform:"
help, ph_radon
```

```
print, "ntheta = ", ntheta
print, "nrho = ", nrho

;; back-projection (not the inverse of the radon transform!)
ph_back = Radon(ph_radon, /backproject, $
                linear=doLinearInterp, $
                rho=rho, theta=theta, $
                nx=nx, ny=ny)

print, ""
print, "back-projected:"
help, ph_back

end ;; radon_outline
```