Subject: Re: Looking for Hough and/or Radon transform code
Posted by Bobstrosity on Mon, 17 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

John McFee wrote in message <7heo6bmkb1@coyote.dres.dnd.ca>...
>
> Would anyone have or know the whereabouts of IDL code to do either the
Hough
> transform or any variants of the Radon transform?
>
> Thanks
>
> John McFee

Included below is a function that calculates the radon
transform using linearly interpolation.
Copy it to a file called radon.pro

Below the function is a small piece of test code.
Run radon.pro for an example of how it can be used.
(it will produce three plots, and 3 lines of text output.
This is a preliminary piece of code so, as the
wise one (mike brady) says ``caveat emptor''.

Cheers,
bob stockwell

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;;;;;
; radon transform
;
; Copyright (c) 1999-5000, R.G. Stockwell
;       Unauthorized modification prohibited.
;
; DISCLAIMER:  this is from RGS's collection of hackware. It
;       is not documented or optimized for speed. Use at your own risk.
;
;
; NAME:
;       RADON
;
; PURPOSE:
;   this incorporates the linear interpolation scheme
;   takes an image h(x,y) and returns  the radon transform
```

```
;   r(p, tau), where  p is the slope from -1 to 1
;   and tau is the y offset along a y axis at x= slength/2
;   ie that is halfway horizontally across the image.
;   if you want to find slopes > 1 or < -1 then rotate the
;   image 90 degrees.
;
; CATEGORY:
;       image pro.
;
; CALLING SEQUENCE:
;       g_radon = radon(image,p,tau,all_rays=all_rays,raylen=raylen)
;
; INPUTS:
;       image:    An 2D image
;
; OUTPUTS:
;       p:     1 D vector of the sampled slopes
;       tau:     1 D vector of the sampled y offsets
;
;
; KEYWORD PARAMETERS:
;      RETURNS EXTENDED INFORMATION, does not affect program execution.
;     all_rays:  a 3 dimensional array (p,tau,ray) holding the ray for each
slope and y offset
;   raylen:    the length of the ray for each ray in the above 3D array.
;      NOTE these parameters are not needed in normal use of the radon
transform.
;     They return the individual rays used in the radon calcuation in
;     you (or I) want to perform a different operation than averaging.
;
; EXAMPLE:  follows the function
;
; REFERENCE:
;       The Analysis of Time Series (Fourth Edition)
;       C. Chatfield
;       ISBN 0-412-31820-2
;
; MODIFICATION HISTORY:
;       Written by:  R.G. Stockwell
;     Research Scientist
;    Colorado Research Associates
;    3380 Mitchell Lane
;    Boulder, Colorado, 80301
;    stockwell@co-ra.com
;    early 1999
;-
```

```
function radon,image,p,tau,all_rays=all_rays,raylen = raylen

if  n_params() lt 1 then return,-1
sz = size(image)
if sz(0) ne 2 then return,-1
xl = sz(1)
yl = sz(2)
x = findgen(xl)-xl/2
y = findgen(yl)
; max size of inray is max of indexes



kl=xl
hl=yl
ml=xl
nl=yl
delta_x =(delta_y=1)
p = (findgen(kl)/(kl-1)*kl-(kl)/2)*delta_y/max(abs(x))
tau = findgen(hl)*delta_y
x_min = min(x)
y_min = min(y)
alpha = p * delta_x/delta_y

g_radon  =  fltarr(kl,hl)
raylen   =  fltarr(kl,hl)
all_rays =  fltarr(kl,hl,xl > yl)

for k = 0,kl-1 do begin
 for h = 0,hl-1 do begin
  beta = (p(k)*x_min+tau(h) - y_min)/delta_y
  if alpha(k) gt 0 then begin
    m_min = 0 > ceil((-beta)/alpha(k))
    m_max = ml - 1 < floor((nl-1-beta)/alpha(k))
  endif else begin
    m_min = 0 > ceil((nl-1-beta)/alpha(k))
    m_max = ml - 1 < floor((-beta)/alpha(k))
  endelse
  sum=0
  mv = findgen(m_max-m_min+1)+m_min
  nfloatv = (alpha(k)*mv+beta)
  wneg = where(nfloatv lt 0.0,wnegcount)
  wre = where(abs(nfloatv) lt 0.01,wrecount)
  if wrecount gt 0 then nfloatv(wre) = 0
  nv = floor(nfloatv)
  wv = nfloatv-nv
  sum = 0
```

```
  if max(nfloatv) gt nl-1 then stop ;this is an error in indexing
  wno = where(wv eq 0,wnocount)
  inray = fltarr(n_elements(mv))
  if wnocount gt 0 then begin
   inray(wno) = image(mv(wno),nv(wno))
   wint = where(wv ne 0,wintcount)
   if wintcount gt 0 then begin

 inray(wint)=image(mv(wint),nv(wint))*(1-wv)+image(mv(wint),n v(wint)+1)*wv
   endif
  endif else inray = image(mv,nv)*(1-wv)+image(mv,nv)*wv
  g_radon(k,h) = delta_x*(mean(inray))
  raylen(k,h) =n_elements(inray)
  all_rays(k,h,0:raylen(k,h)-1) = inray
 endfor
endfor


return,g_radon

end

;_____________________________________________________________
;_____________________________________________________________
; test code here


xl=(yl=100)
image = fltarr(xl,yl)

;slope=-0.8
;offset = 0
slope= (randomu(seed,1))(0)*2-1
offset = (randomu(seed,1))(0)*xl

line = 10
for i = 0,yl-1 do begin
  yind = slope*(i-xl/2)+offset
  if yind ge 0 and yind lt yl then $
   image(i,yind) = image(i,yind) + line
endfor

image=smooth(image,5)
image = image+randomn(seed,xl,yl)

g_radon = radon(image,p,tau,all_rays=all_rays,raylen=raylen)

wm = wheremax(g_radon)
```

```
print
print,'Actual Values for slope and y offset:..... P=',slope,  '    Tau =
: ',offset
print,'Measured slope and offset from radon transform: P=',p(wm(0)),'    Tau
= : ',tau(wm(1))
print,'value at peak of radon transform= ',g_radon(wm(0),wm(1))


!P.multi=[0,1,3]
!P.charsize=2
fill=1
contour,image,fill=fill,tit='Image (line + noise)',xtit='X',ytit='Y'
axis,xl/2,/data,yaxis=1
contour,g_radon,p,tau,tit='Radon
Transform',fill=fill,nlevels=21,xtit='Slope',ytit='Y Offset'
surface,g_radon,p,tau,tit='Radon Transform',xtit='Slope',ytit='Y Offset'


end
```