Subject: Re: perimeter measurement in idl
Posted by Martin Schultz on Tue, 19 Jan 1999 08:00:00 GMT
View Forum Message <> Reply to Message

L. Charles Burgwardt wrote:
>
> I'm looking for a perimeter measurement routine written in IDL. I have a
>
> binary image. The output could be in any units such as pixel perimeter,
> crack perimeter, or line segment. A boundary following routine would be
> good also, ie ordered list of boundary pixels.
>
> Charlie Burgwardt

I attached a little "boundary crawler" routine that I used to get the
outline polygons for individual countries after creating maps with
/continents. Subroutine getpoly does the actual job. It's nothing too
sophisticated and may not be the utmost efficient routine, but it worked
very stable (at least for reasonable choices of starting points). You
need
to give the routine one point that is within the region you want to
circumvent. It will then move north to find the boundary, then crawl
clockwise until it reaches a point that was already marked. Should be
fairly
straightforward to adapt for different threshold values or ranges.

Hope this helps,
Martin.

--
--- ---
Dr. Martin Schultz
Department for Engineering&Applied Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
Internet-homepage: http://www-as.harvard.edu/people/staff/mgs/
--- ---

; ======== HELPER ROUTINES FOR FILLIT ==========

```
function testpix,im,i,j

   test = intarr(4)
   test(0) = im(i+1,j) gt 0   ; east
   test(1) = im(i,j-1) gt 0   ; south
   test(2) = im(i-1,j) gt 0   ; west
   test(3) = im(i,j+1) gt 0   ; north

   return,test
end


function evalpix,im,i,j,direction


   test = testpix(im,i,j)  ; get neighbouring pixel values
   eval = (5 + direction - indgen(4)) mod 4 + 1   ; evaluation mask
   eval = eval*test   ; blank out zero values
   minval = min(eval(where(eval gt 0)))
   newdir = where(eval eq minval)


; print,direction,byte(eval),minval,newdir(0),format='(i2,":",6i4)'

return,newdir(0)
end


; GETPOLY : automatic generation of country boundaries

pro getpoly,im,i,j,x,y,XOFF=xoff,YOFF=yoff


   if (n_elements(XOFF) eq 0) then xoff = 0
   if (n_elements(YOFF) eq 0) then yoff = 0

   if (im(i-xoff,j-yoff) ge 1) then stop,'Already on boundary.'


   while (im(i-xoff,j-yoff) lt 1) do begin
      j = j+1   ; search for boundary
   endwhile

   ; save i,j pair
   i0 = i-xoff
   j0 = j-yoff

   ; initialize x and y and direction
```

```
  x = i0
  y = j0
  im(i-xoff,j-yoff) = 2
  dir = 0     ; 0=E, 1=S, 2=W, 3=N
  steps = 0L

  ; crawl along boundary
  repeat begin
    if (steps mod 200 eq 0) then tv,im
    ; test if pixel at right is boundary, if not, test pixel straight
    ; ahead, then pixel at left, finally move back
    res = evalpix(im,i-xoff,j-yoff,dir)
    case res of
     -1 :  print,'**'
      0 :  i = i+1     ; east
      1 :  j = j-1     ; south
      2 :  i = i-1     ; west
      3 :  j = j+1     ; north
    endcase
    x = [ x, i-xoff ]
    y = [ y, j-yoff ]
    steps = steps+1
    dir = res
    im(i-xoff,j-yoff) = 2
  endrep until ((i-xoff eq i0 AND j-yoff eq j0) OR steps ge 20000)

  tv,im   ; final display

print,steps ,' used for boundary definition.'
return
end




pro ind2xy,index,x,y,xsize=xsize

  if (n_elements(xsize) eq 0) then xsize = 720
  x = (index mod xsize)
  y = fix(index/xsize)

return
end


; ===========  PICKLIST ROUTINE  =========================
;
; interactive picking of country coordinates with mouse
```

```
pro  picklist,filename,SCALE=SCALE,start=start,XOFF=xoff,YOFF=yof f, $
            exclude=exclude

   if (n_elements(xoff) eq 0) then xoff = 0
   if (n_elements(yoff) eq 0) then yoff = 0

   if (n_elements(exclude) eq 0) then exclude = 0

   if (n_elements(start) eq 0) then start = 0L
   if (n_elements(SCALE) eq 0) then SCALE = 1
   ascale = float(SCALE)

   ; create two files interactively :
   ; a country file with starting points for each country
   ; a name file with country numbers and names

   openw,ilun,filename+'.dat',/get_lun
   printf,ilun,'COUNTRY   X   Y'

   openw,olun,filename+'.names',/get_lun


   ; use CURSOR to pick points
   print,'left click : the last point defining a country or an exclusion'
   print,'middle click : not the last point ... (e.g .for islands)'
   print,'right click : no more input (does not produce an entry !)'
   print

   mx=0  &  my=0
   status = -1
   count = long(start)  ; country counter

   cname = ''
   read,cname,prompt='COUNTRY : '
   printf,olun,count,cname,format='(i4,1X,A)'

   while (status ne 4) do begin
      print,count,cname,format='(i4,A)'
      cursor,mx,my,/down,/device
      status = !err
      if (status lt 4) then $
         printf,ilun,count,mx/ascale+xoff,my/ascale+yoff, $
               format='(i6,2f10.3)'

      if (status eq 1) then begin
         count = count + 1
         read,cname,prompt='COUNTRY : '
```

```
        if (cname eq '') then begin
          close,/all
          return
        endif
        printf,olun,count,cname,format='(i4,1X,A)'
      endif

    endwhile

    close,/all

return
end


; ==============  COUNTRIES : MAIN PROGRAM =====================
;
;   requires GIF files with "empty" country outlines as generated
;   by continents.pro
;
;   COUNTRIES operates in 3 steps:
;   - interactive determination of country coordinates and country names
;   - automatic determination of country boundaries
;   - merging of "filled" continent images and aggregating
;
;   typical calling sequence:
;       countries,continent="northamerica",start=1
;       countries,continent="southamerica",start=15
;       countries,continent="europe",start=28
;       countries,continent="asia",start=67
;       countries,continent="australia",start=89
;       countries,continent="africa",start=94
;
;       countries,continent="northamerica",/fillit
;       countries,continent="southamerica",/fillit
;       countries,continent="europe",/fillit
;       countries,continent="asia",/fillit
;       countries,continent="australia",/fillit
;       countries,continent="africa",/fillit
;
;       countries,/merge
;

pro  countries,continent=continent,start=start,fillit=fillit,excl ude=exclude, $
      merge=merge


   ; create region -> grid maps and country -> grid maps
```

```
; NEW version : uses pre-stored GIF images for country boundaries


SCALE = 8  ; 2 makes it a 0.5x0.5 grid, 4 gives better resolution
           ; SCALE must be identical to that used in CONTINENTS.PRO !!


; for simplicity, the window size is the same for all continents, only
; the offset shifts - always draw complete map into pixmap but then copy
; only continent region

XOFF = 0  &  YOFF = 0

if (n_elements(continent) eq 0) then continent = "northamerica"
continent = strlowcase(continent)

; for merge option: define continent names as array and create loop
; other calls willloop only once

if (keyword_set(merge)) then begin
   cnnames = [ 'northamerica', 'southamerica', 'europe', 'asia', $
               'australia', 'africa' ]
   ic0 = 0
   ic1 = 5
   grandim = intarr(SCALE*360,SCALE*180)  ; result image array
endif else begin
   ic0 = 0
   ic1 = 0
endelse


for ic = ic0,ic1 do begin
   ; IMPORTANT : the following offsets must be identical with those
   ; given in CONTINENTS.PRO !!!
   if (keyword_set(merge)) then continent = cnnames(ic)

   case continent of
     "northamerica" : begin
                      XOFF = 0
                      YOFF = 90
                      end
     "southamerica" : begin
                      XOFF = 45
                      YOFF = 20
                      end
     "africa"       : begin
                      XOFF = 120
                      YOFF = 50
```

```
                    end
        "europe"      : begin
                    XOFF = 90
                    YOFF = 90
                    end
        "asia"        : begin
                    XOFF = 225
                    YOFF = 90
                    end
        "australia"   : begin
                    XOFF = 225
                    YOFF = 30
                    end
       else          : message,'invalid continent !'
       endcase

       XSIZE = 135
       YSIZE = 90    ; plot size in degrees (multiply by SCALE)

      ; make sure that complete plotting area is used
      !p.position=[0,0,1,1]

      if (not keyword_set(merge)) then goto,no_merge


; ------   MERGING SECTION    -------------------------------------

      ; read in image with filled continents
      read_gif,continent+'.gif',im

      ; convert all non-white, non-country pixels to white
      ind = where(im lt 3)
      if (ind(0) ge 0) then im(ind) = 0

      print,ic,' : ',max(im)

      ; add partial image to total image
     grandim(SCALE*XOFF:SCALE*(XOFF+XSIZE)-1,SCALE*YOFF:SCALE*(YO FF+YSIZE)-1) =
$
     grandim(SCALE*XOFF:SCALE*(XOFF+XSIZE)-1,SCALE*YOFF:SCALE*(YO FF+YSIZE)-1) +
$
          fix(im)


   endfor  ; ic loop for merge


   print,'grandim:',max(grandim)
```

```
  ; load new color table
  loadct,27,bottom=1
  tvlct,r,g,b,/get

  if (!D.n_colors lt 170) then print,'WARNING : not enough colors !'

  ; write complete image as gif file
  write_gif,'world_countries_large.gif',byte(grandim),r,g,b

  ; condense image by aggregating 4x4 pixels
  ; this gives 0.5x0.5 degree resolution
  ; NOTE: tried rebin before, but this leads to unpleasant white lines
  ; therefore have to go through each 4x4 block

  tmp = intarr(720,360)
  rescale = fix(SCALE/2)

print,'Condensing image ... please be patient!'

  for i=0,719 do begin
     for j=0,359 do begin
        block =  grandim(i*rescale:(i+1)*rescale-1,j*rescale:(j+1)*rescale-1)

        code = 0
        if (max(block) gt 0) then begin   ; found a country
           ; determine country code with maximum portion
           u = block(uniq(block,sort(block)))
           u = u(where(u gt 0))
           for n=0,n_elements(u)-1 do begin
              testind = where(block eq u(n),count)
              if count gt code then code = u(n)
           endfor
        endif       ; else only water

        tmp(i,j) = code

     endfor
     if ((i mod 36) eq 0) then tv,tmp
  endfor

  ; shift by 10 degrees west in order to start at -180
  newim = [ tmp(700:719,*), tmp(0:699,*) ]

  ; move all colors down 3 so that first country starts at 1
  ind = where(newim gt 2)
  newim(ind) = newim(ind)-3
  print,max(newim)
  tv,newim
```

```
   write_gif,'world.countries.gif',byte(newim),r,g,b

   return


;  ---------------------------------------------------------- ------

no_merge:
   ; load continent
   read_gif,continent+'.empty.gif',im
   ind = where(im gt 0)
   if (ind(0) ge 0) then im(ind) = 1

   ; open a display window and show map
   window,xsize=XSIZE*SCALE,ysize=YSIZE*SCALE
   curwin = !d.window

   tv,im
   if (keyword_set(fillit)) then goto,fillit


   ; create picklist
   if (keyword_set(exclude)) then $
         picklist,strlowcase(continent)+'.exclude',scale=SCALE,   $
         XOFF=xoff,YOFF=yoff,/EXCLUDE   $
   else $
        picklist,strlowcase(continent)+'.countries',scale=SCALE,  $
         start=start,XOFF=xoff, YOFF=yoff
return

fillit:
   ; draw black frame around image
   s = size(im)
   im(0:1,*) = 1
   im(s(1)-2:s(1)-1,*)=1
   im(*,0:1) = 1
   im(*,s(2)-2:s(2)-1) = 1
    readdata,strlowcase(continent)+'.countries.dat',data,cols=3, $
           /noheader,skp1=1
   c_col = reform(data(0,*))
   iind = reform(data(1,*))
   jind = reform(data(2,*))

   if (n_elements(start) eq 0) then start = 0

   for i = 0,n_elements(iind)-1 do begin
      if (c_col(i) ge start) then begin
```

```
      getpoly,im,fix(iind(i)*SCALE),fix(jind(i)*SCALE),x,y, $
            xoff=xoff*SCALE,yoff=yoff*SCALE

      ; and display manipulated image
      tv,im
      polyfill,x,y,/device,/fill,color=c_col(i)+3
      im = tvrd()   ; read image with filled continents from display
    endif
  endfor


  ; save full resolution image as bytemap (use write_gif)
  write_gif,strlowcase(continent)+'.gif',im

  ; save map
;  openw,olun,'mapimage.txt',/get_lun
;  for i=180*SCALE-1,0,-1 do $
;     printf,olun,im(*,i),format='(360*SCALE(i1))'
;  close,/all


return
end
```

## File Attachments

1) countries.pro, downloaded 123 times