Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them
Posted by rogass on Sat, 22 Jan 2011 10:11:16 GMT
View Forum Message <> Reply to Message

On 21 Jan., 15:23, chris <rog...@googlemail.com> wrote:
> On 21 Jan., 10:02, Robin Wilson <ro...@rtwilson.com> wrote:
>
>
>
>
>
>
>
>
>
>
>> Hi Chris,
>
>> That's great. Yes, your description is correct - that's exactly what I
>> want to do.
>
>> Cheers,
>
>> Robin
>
>>> Hi Robin,
>>> i have such a vectorised function. Please give more details. Do you
>>> want to clean all pixels including the center pixel in a moving
>>> window, if the center pixel is below a threshold?
>
>>> Cheers
>
>>> CR
>
> ; conv is the 2D convolution result
>
> IDL> conv=randomn(seed,100,100)
> IDL> x=mean(conv);threshold
> IDL> wx=3 & wy=3;moving window size
> IDL> sz=size(conv,/dimensions)
> IDL> ind=cr_get_window(sz,wx,wy);get indices of moving window as index
> vector of size [wx,wy,n_elements(conv)]
> IDL> wh = where((conv[ind])[wx/2,wy/2,*] gt X);evaluate center pixel
> due to threshold
> IDL> ind[*,*,wh]=-1;set all indices in matched windows to -1
> IDL> ind2=uniq(((wi=ind[where(ind ne -1)])),sort(wi));extract uniq
> indices due to same indices in adjacent windows
> IDL> conv[ind2]=0
> IDL> tvscl,conv,0

```
>  IDL> tvscl,conv eq 0,1
>
>  function cr_get_window,sz,wx,wy,mode=mode,ind=ind
>  on_error,2
>
>  sx      =       long(sz[0])
>  sy      =       long(sz[1])
>  wx      =       long(n_elements(wx) gt 0                        ?       wx
   :       3)
>  wy      =       long(n_elements(wy) gt 0                        ?       wy
   :       3)
>  mode=   keyword_set(mode)               ?       -1>mode<3 :       0
>  ind     =       keyword_set(ind)                ?       ind     :       0
>  a       =       n_elements(ind) le 1    ?       ulindgen(sx,sy) :       ind
>  h       =       a[0:wx-1,0:wy-1] mod wx
>  ind     =       rebin(h + rebin(transpose(h[0:wx-1]*sx),wx,wy,/sample),wx,wy,
>  ((ss=sx*sy)),/sample)+$
>                  rebin(reform(a[*],1,1,ss,/over),wx,wy,ss,/sample)
>  if n_elements(sz) eq 3 then begin
>          mode=0
>          ind2    =       sz[2] le ss?    reform(a[0:sz[2]-1],1,1,1,sz[2],/over)  :
$
>                                  (ss le sz[2]?
reform((a[*])[0:sz[2]-1l],1,1,1,sz[2],/over) :
>  ulindgen(1,1,1,sz[2]))
>  endif
>  undefine,h,a
>  case mode of
>  0       :       result= n_elements(sz) lt 3?    ind     :       rebin(ind,
wx,wy,ss,sz[2],/
>  sample) + $
>
                          rebin(ind2*ss,wx,wy,ss,sz[2],/sample)
>  1       :       result= reform(ind,wx*wy,sx*sy,/over)
>  2       :       result= ind[*]
>  else:   result=-1
>  endcase
>  undefine, sx,sy,mode,ind
>  undefine, ind2
>  return, result
>  end
>
>  Hope it helps
>
>  Cheers
>
>  CR
```

Ah, I forgot - you need D. Fannings undefine to clear the memory.

Cheers

CR