Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them
Posted by ben.bighair on Thu, 20 Jan 2011 19:31:52 GMT
View Forum Message <> Reply to Message

On 1/20/11 2:19 PM, Robin Wilson wrote:
> Hi,
>
> I've got a binary image and I want to clean it up a bit by removing all
> pixels (or small groups of pixels) which are surrounded by a lot of space.
>
> I can easily write code using CONVOL to tell me how many pixels were
> 'on' in that window, but I can't see an easy (and fast) way of blanking
> (as in, setting all the pixels to zero) in any windows where the CONVOL
> function has given a value greater than X.
>
> I can think of a way to do it in a loop (looping over all of the points
> that CONVOL found which were greater than X and then constructing 5x5
> windows around them), but there must be a proper 'IDL Way'. Any ideas?
>
> Robin

Hi,

Have you looked at MORPH_OPEN? I think it would be a good place to start.

Cheers,
Ben

Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them
Posted by Robin Wilson on Thu, 20 Jan 2011 19:35:35 GMT
View Forum Message <> Reply to Message

Hi Ben

> Have you looked at MORPH_OPEN? I think it would be a good place to start.

I have investigated this, but I don't think it'll do what I want as it
will affect the other pixels in the image too. I'm hoping just to
operate on pixels that have a lot of space around them (for example,
maybe with a 9x9 window, and only having three pixels turned on). Can
MORPH_OPEN do this?

Robin

Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them

Posted by David Fanning on Thu, 20 Jan 2011 19:52:30 GMT
View Forum Message <> Reply to Message

Robin Wilson writes:

> I have investigated this, but I don't think it'll do what I want as it
> will affect the other pixels in the image too. I'm hoping just to
> operate on pixels that have a lot of space around them (for example,
> maybe with a 9x9 window, and only having three pixels turned on). Can
> MORPH_OPEN do this?

I don't know too much about this, but I would investigate
Morph_HitOrMiss, too.

Cheers,

David

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

Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them
Posted by ben.bighair on Thu, 20 Jan 2011 20:03:45 GMT
View Forum Message <> Reply to Message

On 1/20/11 2:35 PM, Robin Wilson wrote:
> Hi Ben
>
>> Have you looked at MORPH_OPEN? I think it would be a good place to start.
>
> I have investigated this, but I don't think it'll do what I want as it
> will affect the other pixels in the image too. I'm hoping just to
> operate on pixels that have a lot of space around them (for example,
> maybe with a 9x9 window, and only having three pixels turned on). Can
> MORPH_OPEN do this?
>
> Robin
>

Hmmm. So, it isn't just a matter of the size of the speck but also the
amount of white space around it.

Filtering by size isn't such a big issue - you could even use histogram
operating on a labeled version of your image. You could also use
Morph_HitOrMiss for this.

The Morph_TopHat function might also be worth checking out. You can
control the amount of "brim-width" to allow when you specify the
structure.

Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them
Posted by Gray on Thu, 20 Jan 2011 23:24:39 GMT
View Forum Message <> Reply to Message

On Jan 20, 3:03 pm, Ben Tupper <ben.bigh...@gmail.com> wrote:
> On 1/20/11 2:35 PM, Robin Wilson wrote:
>
>> Hi Ben
>
>>> Have you looked at MORPH_OPEN? I think it would be a good place to start.
>
>> I have investigated this, but I don't think it'll do what I want as it
>> will affect the other pixels in the image too. I'm hoping just to
>> operate on pixels that have a lot of space around them (for example,
>> maybe with a 9x9 window, and only having three pixels turned on). Can
>> MORPH_OPEN do this?
>
>> Robin
>
> Hmmm. So, it isn't just a matter of the size of the speck but also the
> amount of white space around it.
>
> Filtering by size isn't such a big issue - you could even use histogram
> operating on a labeled version of your image. You could also use
> Morph_HitOrMiss for this.
>
> The Morph_TopHat function might also be worth checking out. You can
> control the amount of "brim-width" to allow when you specify the
> structure.

You have the indices where the pixels are, do you not? Or did I miss
a step?

IDL> array[x-2:x+2,y-2:y+2] = 0

Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them
Posted by rogass on Fri, 21 Jan 2011 08:07:04 GMT

View Forum Message <> Reply to Message

On 20 Jan., 20:19, Robin Wilson <ro...@rtwilson.com> wrote:
> Hi,
>
> I've got a binary image and I want to clean it up a bit by removing all
> pixels (or small groups of pixels) which are surrounded by a lot of space.
>
> I can easily write code using CONVOL to tell me how many pixels were
> 'on' in that window, but I can't see an easy (and fast) way of blanking
> (as in, setting all the pixels to zero) in any windows where the CONVOL
> function has given a value greater than X.
>
> I can think of a way to do it in a loop (looping over all of the points
> that CONVOL found which were greater than X and then constructing 5x5
> windows around them), but there must be a proper 'IDL Way'. Any ideas?
>
> 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

Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them
Posted by Robin Wilson on Fri, 21 Jan 2011 09:02:44 GMT
View Forum Message <> Reply to Message

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

Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them
Posted by on Fri, 21 Jan 2011 10:41:55 GMT
View Forum Message <> Reply to Message

On Jan 21, 10:02 am, 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

Hi Robin,

You described the problem in two different ways:
(1) clean the pixel neighborhood if the center pixel is below a
threshold
(2) clean the pixel if the number of neighbors on is below a threshold

I understand that you actually mean case (2). You can try that (I did
not test the code):
kernel_size = 9
threshold = 3.0 / (kernel_size*kernel_size) ;3 pixels on out of a 9x9
neighborhood
Image[WHERE(SMOOTH(image, [kernel_size,kernel_size], /EDGE_TRUNCATE)
LT threshold)] = 0

Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them
Posted by on Fri, 21 Jan 2011 10:52:15 GMT
View Forum Message <> Reply to Message

On Jan 21, 11:41 am, Axel M <axe...@gmail.com> wrote:
> On Jan 21, 10:02 am, 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
>
> Hi Robin,
>
> You described the problem in two different ways:
> (1) clean the pixel neighborhood if the center pixel is below a
> threshold
> (2) clean the pixel if the number of neighbors on is below a threshold
>
> I understand that you actually mean case (2). You can try that (I did
> not test the code):
> kernel_size = 9
> threshold = 3.0 / (kernel_size*kernel_size) ;3 pixels on out of a 9x9
> neighborhood
> Image[WHERE(SMOOTH(image, [kernel_size,kernel_size], /EDGE_TRUNCATE)
> LT threshold)] = 0

A correction: if image is not a float array, please change it like
that:
Image[WHERE(SMOOTH(FLOAT(image), [kernel_size,kernel_size], /
EDGE_TRUNCATE) LT threshold)] = 0

If you want the code to be robust, you should also check that WHERE is
not -1.

Subject: Re: Blanking all 5x5 windows with less than X 'on' pixels in them
Posted by rogass on Fri, 21 Jan 2011 14:23:11 GMT
View Forum Message <> Reply to Message

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

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