
Subject: Filter image with moving window and averaging pixels
Posted by rpertaub@gmail.com on Wed, 31 Jan 2007 18:58:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I need to filter some noise out of an image and was pointed towards making a moving/sliding window (size, say 7x7) and compare the center pixel with surrounding pixels right next to border (ignoring middle pixels). And compare the value of center pixel with the border average pixels and thus filter out noise. However, I am not sure how to do this.

Anyone has any idea?

Did I say I was new to IDL...?

Thanks for all your help,

my last query was met by very helpful replies, and I am grateful!

rp

Subject: Re: Filter image with moving window and averaging pixels
Posted by [JD Smith](#) on Wed, 31 Jan 2007 23:52:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 31 Jan 2007 10:58:39 -0800, rpertaub@gmail.com wrote:

> Hello,
> I need to filter some noise out of an image and was pointed towards
> making a moving/sliding window (size, say 7x7) and compare the center
> pixel with surrounding pixels right next to border (ignoring middle
> pixels). And compare the value of center pixel with the border average
> pixels and thus filter out noise. However, I am not sure how to do
> this.
> Anyone has any idea?
> Did I say I was new to IDL...?
> Thanks for all your help,
> my last query was met by very helpful replies, and I am grateful!

I'd use CONVOL, e.g.:

```
kernel=fltarr(7,7)
kernel[0,*]=1.
kernel[6,*]=1.
kernel[* ,0]=1.
kernel[* ,6]=1.
```

```
imconv=convol(image,kernel,/CENTER,/EDGE_TRUNCATE)/(4*7-4)
bad=where(abs(image-imconv) gt threshold)
```

Then again, a box median might be just as good of a statistic, especially

if coupled to a box standard deviation. The former is easy, the latter requires some SMOOTH trickery, discussed several times here.

JD

Subject: Re: Filter image with moving window and averaging pixels
Posted by [Michael Galloy](#) on Wed, 31 Jan 2007 23:55:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jan 31, 11:58 am, "rpert...@gmail.com" <rpert...@gmail.com> wrote:
> I need to filter some noise out of an image and was pointed towards
> making a moving/sliding window (size, say 7x7) and compare the center
> pixel with surrounding pixels right next to border (ignoring middle
> pixels). And compare the value of center pixel with the border average
> pixels and thus filter out noise. However, I am not sure how to do
> this.
> Anyone has any idea?

I'm not sure I'm following you, but how about this:

```
; read/generate your image
im = byte(randomu(seed, 200, 200) * 255)

; create a kernel with 1's on the outside edges
kernel = bytarr(7, 7) + 1B ; set everything to 1's then...
kernel[1:5, 1:5] = 0B ; set the inside to zero
print, kernel
```

```
borderAverage = convol(im, kernel, total(kernel), /edge_truncate)
```

You say "compare the value of center pixel with the border average pixels" and I'm not sure what you want to do as a result of this comparison, but the border average pixels are in borderAverage. So you could compare borderAverage[10, 20] to im[10, 20] to compare the border average to the image value at pixel (10, 20).

Mike

--

www.michaelgalloy.com

Subject: Re: Filter image with moving window and averaging pixels
Posted by rpertaub@gmail.com on Thu, 01 Feb 2007 16:13:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thank you for the responses!! Really great to know I am not alone as I

learn my way thru this language!

Anyway, I ended up doing a some arithmetic to find neighboring pixels and used a couple of for loops to go through all the pixels in the image.(still need to add a border to image to get all these pixels)!! It is not very elegant, but once I have everything working, I might go back and make it more elegant with the kernel and convol function which I am still trying to grasp does exactly!

```
w=2           ;width of buffer pixels
for j=2,(imagesize[0]-1-w) DO BEGIN
    for i=2,(imagesize[1]-1-w) DO BEGIN

        centervalue=N[i,j]
        print,"Center Value is ",centervalue

        topH=N[(i-w):(i+w),(j-w):(j-w)]
        sum1=total(topH)
        ele1=N_elements(topH)

        rghtside=N[(i+w):(i+w),(j-w+1):(j+w)]
        sum2=total(rghtside)
        ele2=N_elements(rghtside)

        BtmH=N[(i-w):(i+w-1),(j+w):(j+w)]
        sum3=total(BtmH)
        ele3=N_elements(BtmH)

        Lftside=N[(i-w):(i-w),(j-w+1):(j+w-1)]
        sum4=total(Lftside)
        ele4=N_elements(Lftside)

        Average=float((sum1+sum2+sum3+sum4)/(ele1+ele2+ele3+ele4))
        print, "Total Average is ",average
        ratio = float(average/centervalue)
        print,"Ratio is average of border divided bycenter",ratio

    endfor
endfor
```
