Subject: Re: Filter image with moving window and averaging pixels
Posted by Jean H. on Wed, 31 Jan 2007 19:48:58 GMT
View Forum Message <> Reply to Message

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!
> rp
>

A way to do this would be to define your neighborhood, based on 0;0 and
then to use it to compute whatever you want.

dimX = ... (of your image)
dimY = ...
computeForPixel = indgen(dimX*dimY) ; <== consider the border effect...
you might want to leave every pixels within 3 pixels of the borders..)

nbValidPixel = n_elements(computeForPixel)

Neighborhood = [-2*dimX-3,-2*dimX-2, -2*dimX-1,-2*dimX,-2*dimX+1,
-2*dimX+2,-2*dimX+3,-2,2,2*dimX-3,2*dimX-2, 2*dimX-1,2*dimX,2*dimX+1,
2*dimX+2,2*dimX+3]   ;if you want to change the neighborhood size, write
a function that would compute this indices!

NbPixelsInNeighborhood = n_elements(Neighborhood)

Now, if your image is rather small, you can do something like:

pixelIndices =
 transpose(rebin(computeForPixel,nbValidPixel,NbPixelsInNeigh borhood)) +
rebin(neighborhood,NbPixelsInNeighborhood,nbValidPixel)

meanNeighborhoodValue = total(image[pixelIndices],1) /
float(NbPixelsInNeighborhood)

==> you now have the mean value for every pixel you have specified and
you can easily compare it witht the value of the pixel.

If your image is quite huge and you run out of memory, you can do a simple loop...

```
for i=0l, nbValidPixel do begin
    meanValue = mean(image[i+Neighborhood])
   ;save or compare this value.
endfor
```

Jean