
Subject: Re: image-convolution

Posted by [Yngvar Larsen](#) on Fri, 08 Mar 2013 11:38:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Friday, 8 March 2013 08:27:11 UTC+1, TERRAX wrote:

> hello, idl-community. i (a IDL-beginner) have a problem with the creation of a window function for image processing. after many hours of tinkering, these problem-developed into a real nightmare for me. maybe you can help me.

>

>

>

>

>

> context is the processing of a multi-spectral image. for example 250 x 250 pixels and 5 bands. on this image, i want implement the following function for an image-convolution :

>

>

>

> $c[i] = 0.25 \cdot n-1[i] + 0.5 \cdot n[i] + 0.25 \cdot n+1[i]$

>

>

>

> c [i] forms the value at pixel i in the smoothed image band.

>

> n [i] is the original pixel values of the input image in band n.

>

> n-1[i] is the corresponding pixel in the input band n-1 and n+1[i] is the corresponding pixel in the input band n+1.

>

>

>

> the neighbors for smoothing are therefore the corresponding pixels of the adjacent bands.

>

>

>

> for the implementation i took the Convol function and loops, but I failed miserably. I can idl not teach, that for convolution the same pixel from the adjacent bands to be used.

You don't tell why looping over every pixel and convolving the spectral dimension with [0.25, 0.5, 0.25] fails. It should work fine.

```
;; Random test data
```

```
im = randomu(seed, Nbands, 250, 250)
```

```
for
```

But it is most likely much slower than the following trick that will work well, at least for short filters and moderately sized images:

```
Nf = 1  
filter = [0.25, 0.5, 0.25] ; 2*Nf-1 coefficients  
imsmooth = fltarr(Nbands, 250, 250)
```

```
for n=-Nf, Nf do imsmooth += filter[n+Nf]*im[shift(indgen(nbands), n), *, *]
```

This is equivalent to the /EDGE_WRAP keyword to CONVOL. Handling edge effects in other ways is left as an exercise to the reader :)

```
--  
Yngvar
```
