Subject: Re: convolve mystery
Posted by wlandsman on Wed, 06 Nov 2013 16:54:34 GMT
View Forum Message <> Reply to Message

CONVOLVE() is user-written procedure in the IDL Astronomy Library. The version you are looking at is quite old -- the current version is in

http://idlastro.gsfc.nasa.gov/ftp/pro/image/convolve.pro

The documentation for the current version says that "the image is padded with zeros so that a large PSF does not overlap one edge of the image with the opposite edge of the image." So I'd say that CONVOLVE is giving the right answer -- or closer to what one would get with a true convolution. It also matches CONVOLVE_FFT() without the /NO_PADDING keyword.

-Wayne

P.S. Is it possible that you are using a newer version of CONVOLVE, and not the version on the Web page?

On Wednesday, November 6, 2013 10:13:49 AM UTC-5, Mats Löfdahl wrote:
> I found something surprising (to me) with the convolve() IDL function. There is something strange about how it does its Fourier wrap-around of an image from one side of the array to the other.
>
>
>
> Here is a simple example. First define a simple image where half is unity and half is zero:
>
>
>
> sz = 10
>
> im = [replicate(1., sz/2), replicate(0., sz/2)] # replicate(1., sz)
>
> print, 'Original:'
>
> print,im[*,sz/2], format = '(f5.2)'
>
>
>
> This gives the output:
>
>
>
> Original:
>
> 1.00
>

> 1.00
>
> 1.00
>
> 1.00
>
> 1.00
>
> 0.00
>
> 0.00
>
> 0.00
>
> 0.00
>
> 0.00
>
>
>
>
>
> Then define a point spread function and do the convolution:
>
>
>
> psf1 = [[1., 1., 1.], [1., 5., 1.], [1., 1., 1.]]
>
> psf1 = psf1/total(psf1)
>
> imc1 = convolve(im, psf1)
>
> print, 'With convolve:'
>
> print,imc1[*,sz/2], format = '(f5.2)'
>
>
>
> The output I get is:
>
>
>
> With convolve:
>
> 0.77
>
> 1.00
>

> 1.00
>
> 1.00
>
> 0.77
>
> 0.23
>
> -0.00
>
> 0.00
>
> -0.00
>
> -0.00
>
>
>
> See how the wrap-around reduced the 1.00 in the first pixel to 0.75 but the last pixel does not get the corresponding increase?
>
>
>
> Whereas if I do the equivalent operation explicitly with FFT, I do get the expected 0.23 in the last pixel:
>
>
>
> psf2 = fltarr(sz, sz)
>
> psf2[sz/2-1:sz/2+1, sz/2-1:sz/2+1] = psf1*sz*sz
>
> psf2 = shift(psf2, sz/2, sz/2)
>
> imc2 = float(fft(fft(im)*fft(psf2), /inv))
>
> print, 'With fft:'
>
> print,imc2[*,sz/2], format = '(f5.2)'
>
>
>
> With fft:
>
> 0.77
>
> 1.00
>

> 1.00
>
> 1.00
>
> 0.77
>
> 0.23
>
> -0.00
>
> -0.00
>
> -0.00
>
> 0.23
>
>
>
> I've looked at the code in http://www.astro.washington.edu/docs/idl/cgi-bin/getpro/library21.html?CONVOLVE and as far as I can see (due to various options the code is not entirely straight forward to read), the fft convolution has no reason to do give any different result from what I do explicitly with fft.
>
>
>
> Does anybody know what is going on?
