
Subject: Convolution Kernel

Posted by [Gray](#) on Thu, 02 Dec 2010 17:35:24 GMT

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

Hi all,

Maybe my calculus is screwy, but this doesn't make sense to me.
Here's my issue:

I have two astronomical images (of stars). I've fit an average PSF as a Moffat profile for each of the two images. I want to find the optimal convolution kernel to match the two psfs, so I call on my old friend Mr. Fourier. If MA is the Moffat profile for image A and MB is the Moffat profile for image B (both 2d), and K is my optimal kernel, then I can do this:

$MA ** K = MB$ --> $**$ is convolution in this scenario
 $F(MA ** K) = F(MB)$ --> $F()$ is the Fourier transform
 $F(MA) * F(K) = F(MB)$
 $K = F^{-1}(F(MB)/F(MA))$

With me so far? So I do this in IDL.

```
IDL> ma = moffat(params_a)
```

```
IDL> mb = moffat(params_b)
```

```
IDL> fma = fft(ma) & fmb = fft(mb)
```

```
IDL> k = fft(fma/fmb,inverse)
```

```
IDL> mc = convol(ma,k)
```

What I get, however, is that MC is a 2d delta function. Why...? It happens with 2d Gaussians, as well. Thanks for your help!

--Gray

Subject: Re: Convolution Kernel

Posted by [MC](#) on Fri, 03 Dec 2010 10:23:16 GMT

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

On Dec 3, 6:35 am, Gray <grayliketheco...@gmail.com> wrote:

> Hi all,

>

> Maybe my calculus is screwy, but this doesn't make sense to me.

> Here's my issue:

>

> I have two astronomical images (of stars). I've fit an average PSF as
> a Moffat profile for each of the two images. I want to find the
> optimal convolution kernel to match the two psfs, so I call on my old
> friend Mr. Fourier. If MA is the Moffat profile for image A and MB is

> the Moffat profile for image B (both 2d), and K is my optimal kernel,
 > then I can do this:
 >
 > $MA ** K = MB$ --> $**$ is convolution in this scenario
 > $F(MA ** K) = F(MB)$ --> $F()$ is the Fourier transform
 > $F(MA) * F(K) = F(MB)$
 > $K = F^{-1}(F(MB)/F(MA))$
 >
 > With me so far? So I do this in IDL.
 > IDL> ma = moffat(params_a)
 > IDL> mb = moffat(params_b)
 > IDL> fma = fft(ma) & fmb = fft(mb)
 > IDL> k = fft(fma/fmb,/inverse)
 > IDL> mc = convol(ma,k)
 >
 > What I get, however, is that MC is a 2d delta function. Why...? It
 > happens with 2d Gaussians, as well. Thanks for your help!
 >

Hmm, not sure what Moffat does, but maybe the division by fmb terms with small amplitudes could be the problem? What happens if you add a small constant to fmb? If you want to match the OTF of the two images could you just convolve a with the psf of b and convolve b with the psf of a?

Hope this helps
 MC

Subject: Re: Convolution Kernel
 Posted by [Bringfried Stecklum](#) on Fri, 03 Dec 2010 10:57:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Gray wrote:

> Hi all,
 >
 > Maybe my calculus is screwy, but this doesn't make sense to me.
 > Here's my issue:
 >
 > I have two astronomical images (of stars). I've fit an average PSF as
 > a Moffat profile for each of the two images. I want to find the
 > optimal convolution kernel to match the two psfs, so I call on my old
 > friend Mr. Fourier. If MA is the Moffat profile for image A and MB is
 > the Moffat profile for image B (both 2d), and K is my optimal kernel,
 > then I can do this:
 >
 > $MA ** K = MB$ --> $**$ is convolution in this scenario
 > $F(MA ** K) = F(MB)$ --> $F()$ is the Fourier transform

```
> F(MA) * F(K) = F(MB)
> K = F^-1(F(MB)/F(MA))
>
> With me so far? So I do this in IDL.
> IDL> ma = moffat(params_a)
> IDL> mb = moffat(params_b)
> IDL> fma = fft(ma) & fmb = fft(mb)
> IDL> k = fft(fma/fmb,/inverse)
> IDL> mc = convol(ma,k)
>
> What I get, however, is that MC is a 2d delta function. Why...? It
> happens with 2d Gaussians, as well. Thanks for your help!
>
> --Gray
```

In principle, your approach is right. First of all make sure the PSF MA is wider than MB otherwise deconvolution won't give anything meaningful. Furthermore, deconvolution of real images is tricky because of the presence of noise. I guess you want to match the PSFs to do image subtraction. If so this paper might provide some more information on the subject (in fact I think it describes the most sophisticated method at present)

http://esoads.eso.org/cgi-bin/nph-data_query?bibcode=2008MNRAS.386L..77B&db_key=AST&link_type=ABSTRACT&high=44a4d6a73417468

The author provides the IDL code (<http://www.danidl.co.uk/index.html>) provided you pay the subscription rate.

You may try the ISIS code for free (<http://www2.iap.fr/users/alard/package.html>).

Bringfried

Subject: Re: Convolution Kernel
Posted by [Jeremy Bailin](#) on Fri, 03 Dec 2010 18:06:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Dec 3, 5:23 am, MC <morefl...@gmail.com> wrote:

```
> On Dec 3, 6:35 am, Gray <grayliketheco...@gmail.com> wrote:
>
>
>
>
>
>
>
>
>
>> Hi all,
```

```

>
>> Maybe my calculus is screwy, but this doesn't make sense to me.
>> Here's my issue:
>
>> I have two astronomical images (of stars). I've fit an average PSF as
>> a Moffat profile for each of the two images. I want to find the
>> optimal convolution kernel to match the two psfs, so I call on my old
>> friend Mr. Fourier. If MA is the Moffat profile for image A and MB is
>> the Moffat profile for image B (both 2d), and K is my optimal kernel,
>> then I can do this:
>
>> MA ** K = MB --> ** is convolution in this scenario
>> F(MA**K) = F(MB) --> F() is the Fourier transform
>> F(MA) * F(K) = F(MB)
>> K = F^-1(F(MB)/F(MA))
>
>> With me so far? So I do this in IDL.
>> IDL> ma = moffat(params_a)
>> IDL> mb = moffat(params_b)
>> IDL> fma = fft(ma) & fmb = fft(mb)
>> IDL> k = fft(fma/fmb,/inverse)
>> IDL> mc = convol(ma,k)
>
>> What I get, however, is that MC is a 2d delta function. Why...? It
>> happens with 2d Gaussians, as well. Thanks for your help!
>
> Hmm, not sure what Moffat does, but maybe the division by fmb terms
> with small amplitudes could be the problem? What happens if you add a
> small constant to fmb? If you want to match the OTF of the two images
> could you just convolve a with the psf of b and convolve b with the
> psf of a?
>
> Hope this helps
> MC

```

(I thought I posted this this morning, but it never showed up)

This is certainly what's going on. I tried this with some Gaussians, but used

```
fma/(fmb > 1e-10)
```

instead inside the `k=fft(...)` line, and it changed from nonsense to working beautifully. Outside of the support of the Moffat fit to b, you're just getting amplified numerical noise. Try looking at the difference between those ratios (with TVIMAGE or SURFACE or something) and you'll see what's going on.

-Jeremy.
