Subject: Re: Scale the psf on images.
Posted by anes.tziamtzis on Thu, 15 Jan 2015 07:13:16 GMT
View Forum Message <> Reply to Message

Well perhaps i will get the award for the dumbest person in the world for this.

On the following link there is an fft example
http://www.exelisvis.com/docs/fftreducebackgroundnoise.html

As you said Mats they do something to reduce the noise. The problem is by using this method, again the final image doesnt make sense. There is no structure actually and the pixel values vary from -1e7 to 1e7. This is the code

```
fits_read, 'Image1.fits',image1,header1
fits_read, 'Image1.fits.psf.1.fits',image2,header2
fits_read, 'Image2.fits',image3,header3
fits_read, 'Image2.fits.psf.1.fits',image4,header4


;Compute the difference kernel
p1=fft(image2)
p2=fft(image4)

; Compute the power spectrum of the transform and apply a log scale.

powerSpectrum = ABS(p1)^2
scaledPowerSpect = ALOG10(powerSpectrum)

powerSpectrum2 = ABS(p2)^2
scaledPowerSpect2 = ALOG10(powerSpectrum2)


; Scale the power spectrum to make its maximum value equal to 0.
scaledPS = scaledPowerSpect - MAX(scaledPowerSpect)
scaledPS2 = scaledPowerSpect2 - MAX(scaledPowerSpect2)

surface,powerSpectrum2

mask = REAL_PART(scaledPowerSpect) GT -5
psf1_trans = p1*mask
mask2 = REAL_PART(scaledPowerSpect2) GT -5
psf2_trans = p2*mask2
kernel = REAL_PART(FFT(psf1_trans/psf2_trans, /INVERSE, /CENTER))


image3_prime = convol(image3, kernel)
diff = image1 - image3_prime
```

```
WRITEFITS,'im_conv.fits',diff,header3
end
```