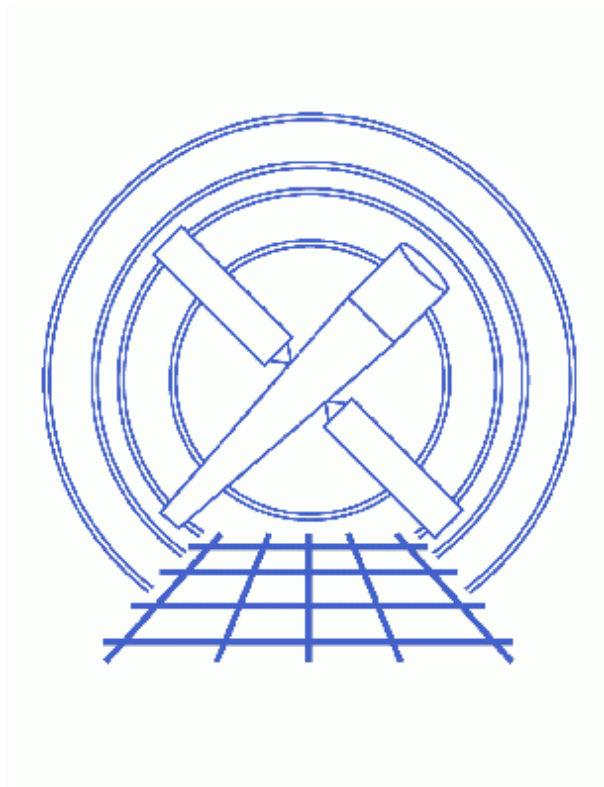


# Using a PSF Image as the Convolution Kernel



**Sherpa Threads (CIAO 3.4)**

# Table of Contents

- *Getting Started*
- *Reading & Filtering Image Data*
- *Defining the Instrument Model with a PSF Image*
  - ◆ FPSP model parameters
- *Defining a Multi-Component Source Model Expression*
- *Modifying Statistic Setting*
- *Fitting*
- *Examining Fit Results*
- *Summary*
- *History*
- *Images*
  - ◆ Input image data
  - ◆ Input image filter region
  - ◆ Filtered image data
  - ◆ PSF image
  - ◆ PSF sub-image for convolution
  - ◆ Data, best-fit model, and residuals
  - ◆ Radial profile of the best fit model
  - ◆ Surface plot of the residuals
  - ◆ Contour plot of the residuals

# Using a PSF Image as the Convolution Kernel

*Sherpa Threads*

## Introduction

In this thread, we fit 2-D image data using a Point-Spread-Function (PSF) image as the convolution kernel.

## Getting Started

Please follow the "[Sherpa Threads: Getting Started](#)" thread.

## Reading & Filtering Image Data

In this thread, we fit 2-D image data from the FITS datafile `center_box_0.25pix.fits`. This is the image created from an event file by binning over a region to 0.25 ACIS pixel size with `dmcopy`; for example:


```
unix% dmcopy "event.fits[sky=region(center_box.reg)][bin x>::0.25,y>::0.25]" \
  center_box_0.25pix.fits
```

This dataset is input into *Sherpa* with the `DATA` command:

```
sherpa> DATA center_box_0.25pix.fits
```

Now the dataset may be displayed:

```
sherpa> IMAGE DATA
```

The input data image looks [like this](#) .

There are several ways of filtering image data within *Sherpa*. Here, we illustrate two ways. Note that interactive filtering directly from ds9 (method 1 below) can be used during the session, while the command-line filters (method 2 below) can be used in scripts.

1. Use ds9 regions to filter the data. After the data have been displayed, go to the Region box in ds9 and choose a designed region shape. In this example we use the box shape. When the box is displayed, the size of the box can be changed with the cursor or within the Region Info box which is displayed from the Region box with the Get Info... button. After the desired region size is set, the region can be used to filter the data as follows:

```
sherpa> IGNORE ALL
sherpa> NOTICE IMAGE
```

## Using a PSF Image – Sherpa

Note that the `NOTICE IMAGE` command will notice all the pixels that are included in the current region displayed in ds9.

```
sherpa> SHOW

Optimization Method: Levenberg-Marquardt
Statistic:           Chi-Squared Gehrels

-----
Input data files:
-----

Data 1: center_box_0.25pix.fits fits.
Total Size: 24843 bins (or pixels)
Dimensions: 2
Size: 169 x 147
Coordinate setting: logical
Total counts (or values): 1831

Current filters for dataset 1:
ignore source 1 all
NOTICE source 1 FILTER "BOX(88.16875,75.8625,70.416667,68.508334)"
Noticed filter size: 4899 bins

Sum of data within filter: 1788
```

2. The same filter can be set on the command-line with the region definition. Note that *Sherpa* requires the Image coordinates (not the Physical coordinates) to be used in the region definitions:

```
sherpa> IGNORE ALL
sherpa> NOTICE FILTER "BOX(88.16875,75.8625,70.416667,68.508334)"
```

We can now display the filter region:

```
sherpa> IMAGE FILTER
```

The filter region looks like this .

In this case, the following command will display the filtered data:

```
sherpa> IMAGE DATA
```

The filtered data looks like this .

---

## Defining the Instrument Model with a PSF Image

The instrument model `FPSF`, which takes an image of the PSF, is established and named `psf0`:

```
sherpa> PARAMPROMPT OFF
Model parameter prompting is off
sherpa> FPSF[psf0]
sherpa> psf0.file=psf_f1_norm_0.25pix.fits
sherpa> SHOW psf0
fpsf2d[psf0]
  Param  Type      Value      Min      Max      Units
  ----  -
  1  file string: "psf_f1_norm_0.25pix.fits"
```


## Using a PSF Image – Sherpa

```
2  xsize frozen      32         1       1024
3  ysize frozen      32         1       1024
4  xoff frozen       0        -512       512
5  yoff frozen       0        -512       512
6  fft frozen        1         0         1
```

```
sherpa> INSTRUMENT = psf0
```

Note that the PSF is automatically renormalized to 1. Renormalization is done by summing over all image pixels, regardless of the setting of `xsize` and `ysize`.

The PSF image file was created using the CIAO tool `mkpsf` (see the CIAO [Create a PSF thread](#)). Note that in recent CIAO versions the best PSF image can be created with [The Chandra Ray Tracer \(ChaRT\)](#)). The file `center_box_0.25pix.fits` was used as input to `mkpsf` in order to match the binning of the resulting PSF image file (`psf_fl_norm_0.25pix.fits`). *Sherpa* requires that the binning of the PSF image file match the binning of the input data image.

To [view the PSF image](#) , load it into ds9 (from outside *Sherpa*).

## FPSF model parameters

- ***xsize* & *ysize*:**

The PSF image provided via the `file` parameter may be much larger than the FPSF size (in number of pixels), and therefore larger than needed. In order to speed the convolution process, a sub-image kernel (`xsize` by `ysize`) is specified. In this example, the input PSF image file has 255x255 pixels, but the portion that will be used for the convolution is only the center 32x32 pixels sub-image.


In some cases a larger sub-image will be needed: when the source is located off-axis, or when including the PSF wings is important for the analysis.

To see the sub-image of the PSF image file that will be used for the convolution:

```
sherpa> IMAGE psf0
NOTE: PSF fraction for (xsize,ysize): FRAC = 0.976061
```

This sub-image contains only a part of the input file defined by the size and offset parameters. Note that the sub-image will be empty if the PSF centroid is located outside the sub-image. When the `image` command is issued *Sherpa* prints out the information about the PSF fraction included in the sub-image. Updating `xsize` and `ysize` parameters increases the PSF fraction to about 99%.

```
sherpa> psf0.ysize=72
sherpa> psf0.xsize=72
sherpa> image psf0
NOTE: PSF fraction for (xsize,ysize): FRAC = 0.991294
```

The FPSF sub-images look [like this](#) . The left panel shows the default size sub-image of 32x32 pixel. The right panel shows an expanded to 72x72 pixels sub-image. Notice that the larger sub-image contains a significant amount of the PSF wings structure.

- ***xoff* & *yoff*:**

The FPSF (kernel) centroid must always be at the center of the extracted sub-image. The parameters `xoff` and `yoff` move the center of the extracted sub-image away from the center of the original `file` image. Here, `xoff` = `yoff` = 0 and so the kernel sub-image is extracted from the center of the

original file image.

- **fft:**

This parameter controls whether the convolution will be performed using Fast Fourier Transforms (`fft=1`) or the sliding cell technique (`fft=0`). For convolution with a large kernel, the best choice is FFT (the default).

## Defining a Multi-Component Source Model Expression

Now we will define a source model expression, using a two-dimensional Gaussian function called `GAUSS2D`, and a constant called `CONST2D`:

```
sherpa> SOURCE = CONST2D[cc1] + GAUSS2D[g2]
sherpa> g2 INTEGRATE ON

sherpa> SHOW SOURCE
Source 1: (cc1 + g2)
const2d[cc1] (integrate: on)
  Param  Type  Value      Min      Max      Units
  ----  -
  1      c0  thawed    27.5      0        55
gauss2d[g2] (integrate: on)
  Param  Type  Value      Min      Max      Units
  ----  -
  1      fwhm thawed    5.9377  5.9377e-02  593.7728
  2      xpos thawed    88.5     52.5     123.5
  3      ypos thawed    76.5     41.5     110.5
  4      ellip frozen    0        0        0.999
  5      theta frozen    0        0        6.2832
  6      ampl thawed    55      0.55     5500
```

The model component `cc1` is interpreted as the constant background contribution to the data.

## Modifying Statistic Setting

Since the data being fit has low counts, we wish to change the statistic to `CASH`:

```
sherpa> STATISTIC CASH
```

Note that truncation is turned on when the Cash statistic is used. This setting prevents negative model-predicted data values from affecting the convergence process.

Further details about the Cash statistic method are available by typing:

```
sherpa> ahelp cash
```

## Fitting

The data is first fit assuming a constant background (i.e. `cc1` is frozen):

```

sherpa> cc1.c0=1
sherpa> FREEZE cc1.c0
sherpa> FIT
NOTE: PSF fraction for (xsize,ysize): FRAC = 0.991294
WARNING: the Levenberg-Marquardt optimization method works
         less robustly when the Cash or cstat statistic is used.
         Consider using the Powell or Simplex method instead.
LVMQT: V2.0
LVMQT: initial statistic value = 4813.17
LVMQT: final statistic value = 4146.76 at iteration 43
         g2.fwhm  2.71486
         g2.xpos  88.6616
         g2.ypos  77.2268
         g2.ampl  175.802

```

Levenberg–Marquardt optimization method is the default *Sherpa* method, but it is not robust with our choice of statistics. We update the optimization method to Powell which is slower, but more robust and appropriate in this case and refit:

```

sherpa> method powell
sherpa> FIT
powll: v1.2
powll: initial statistic value =      4.14676E+03
powll: converged to minimum =      4.14676E+03 at iteration =      2
powll: final statistic value =      4.14676E+03
         g2.fwhm  2.71384
         g2.xpos  88.6616
         g2.ypos  77.2268
         g2.ampl  175.813

```

The fit is run again with the background component thawed.

```

sherpa> THAW cc1.c0
sherpa> FIT
powll: v1.2
powll: initial statistic value =      4.14676E+03
powll: converged to minimum =     -4.45758E+03 at iteration =     12
powll: final statistic value =     -4.45758E+03
         cc1.c0  0.00276129
         g2.fwhm  3.67578
         g2.xpos  88.5798
         g2.ypos  77.2409
         g2.ampl  115.973

```

---

## Examining Fit Results

The fit results may be examined with:


```
sherpa> IMAGE FIT
```

This command displays the data, best fit model, and residuals in three ds9 frames, as shown in this image .

## Using a PSF Image – Sherpa


The `IMAGE FIT` command displays a montage of the data, current model, and residuals that can be used to see how well the model describes the data. This can be hard to interpret. We can make use of the `plot_rprofr()` function provided by the `sherpa_plotfns.sl` script. See also [Fitting FITS Images](#) thread for more details.

```
sherpa> () = evalfile("sherpa_plotfns.sl");  
sherpa> plot_rprofr(0,80,5)  
sherpa> d 1 log y
```


This plots a radial profile of the data and model (points and solid line respectively in the image [like this](#) ) in the top plot and a radial profile of the residual image in the bottom plot. The arguments in the function call refer to the minimum (0) and maximum (80) radii of the profile and the third value (5) is the bin width (the coordinate system matches that set by the `COORD` command). The center is found from the source component which contains parameters called `xpos` and `ypos` (here it is the `g2` model). The radial profile indicates quite good fit parameters in this case.

To further examine the residuals using surface and/or contour plots:

```
sherpa> SPLOT RESIDUALS
```

The resulting surface plot looks [like this](#) .

```
sherpa> CPLOT RESIDUALS  
sherpa> cplot residuals  
=> Error bars computed using Chi Gehrels.  
Contour Levels: 12.7666 9.57494 6.38328 3.19163 -2.6213e-05  
Min: -12.2408, Max: 12.7666, Ave: -2.6213e-05
```

The resulting contour plot looks [like this](#) .

The residual data (in counts) may be written to an external file:

```
sherpa> WRITE RESIDUALS 2dpsf_resid_cnts.fits FITS  
Creating new filter for 2D visualization...done.  
Write X-Axes: (Bin,Bin) Y-Axis: Counts
```

This file may then be used in further analysis, such as smoothing the residuals with [aconvolve](#).

---

## Summary

This thread is complete, so we can exit the *Sherpa* session:

```
sherpa> EXIT
```

---

## History

14 Dec 2004 reviewed for CIAO 3.2: no changes

21 Dec 2005 reviewed for CIAO 3.3: minor changes to fit results



## Using a PSF Image – Sherpa

29 Jun 2006 added `() = evalfile("sherpa_plotfns.sl");` command in Examining Fit Results section

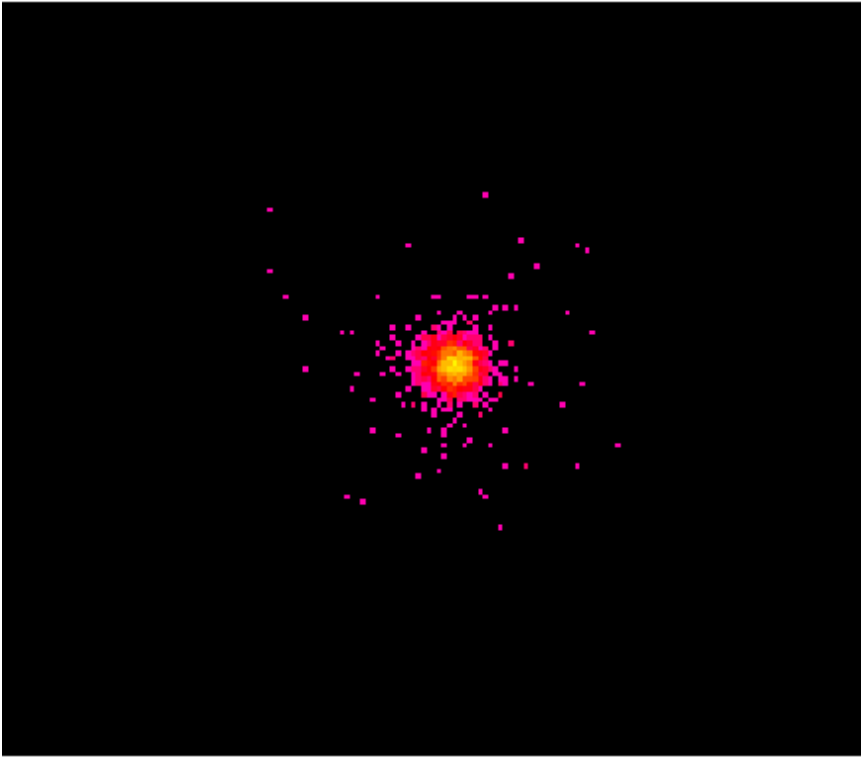
01 Dec 2006 reviewed for CIAO 3.4: no changes

---

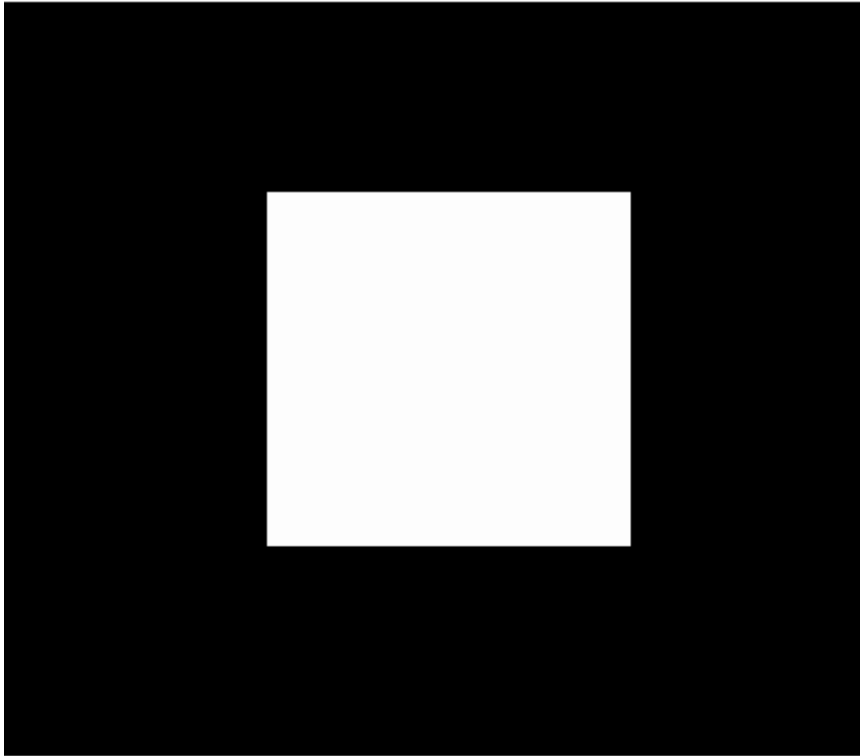
URL: <http://cxc.harvard.edu/sherpa/threads/2dpsf/>

Last modified: 1 Dec 2006

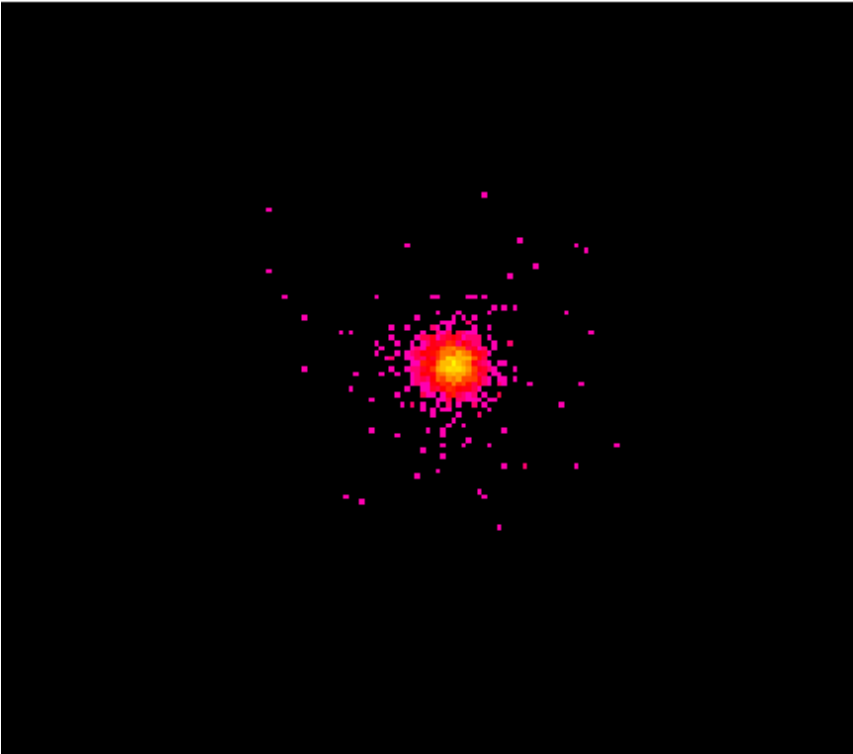
**Image 1: Input image data**



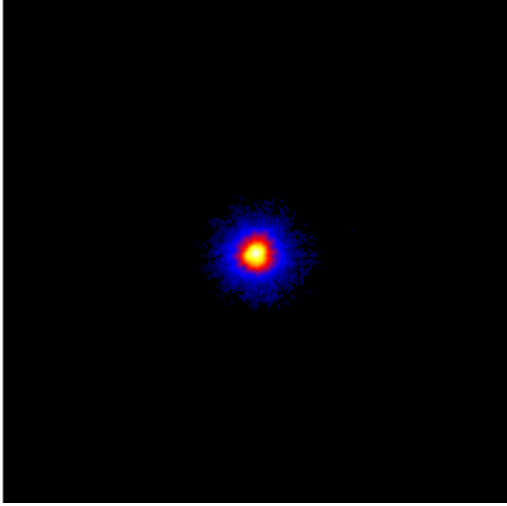
**Image 2: Input image filter region**



**Image 3: Filtered image data**



**Image 4: PSF image**



**Image 5: PSF sub-image for convolution**

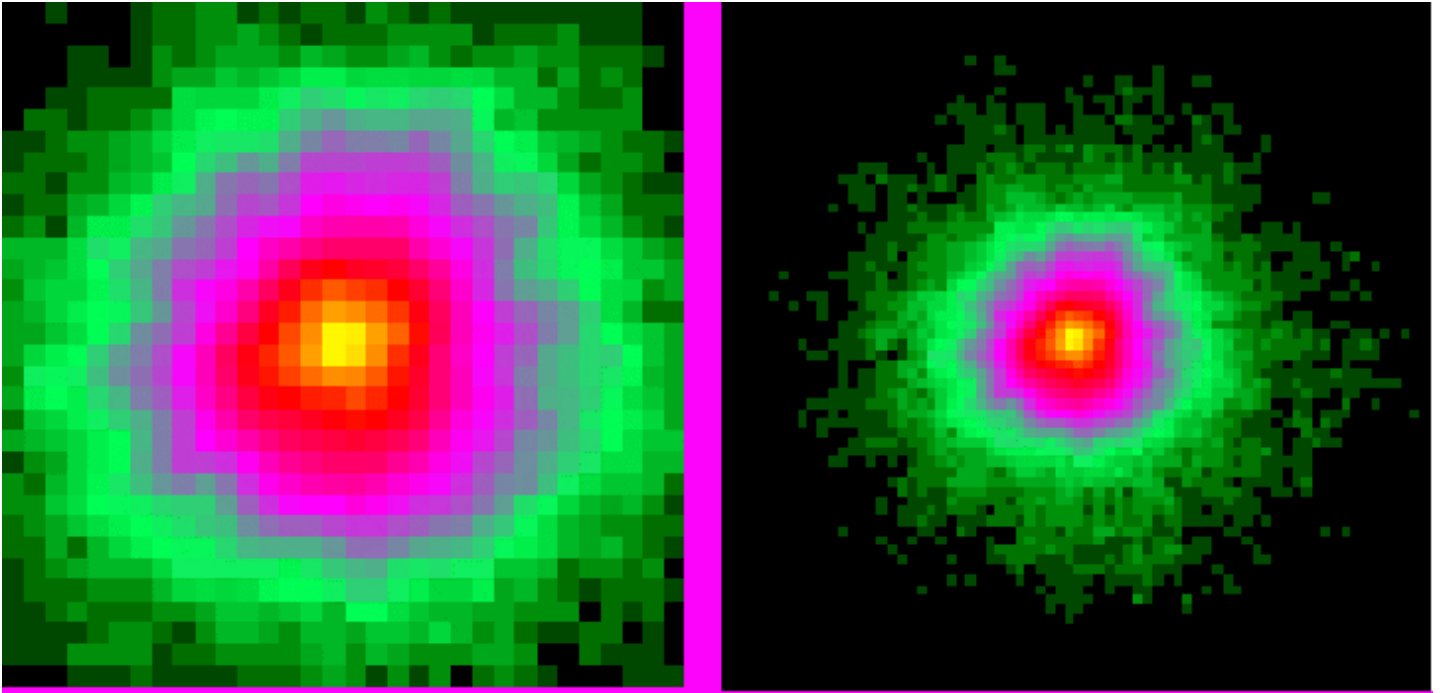
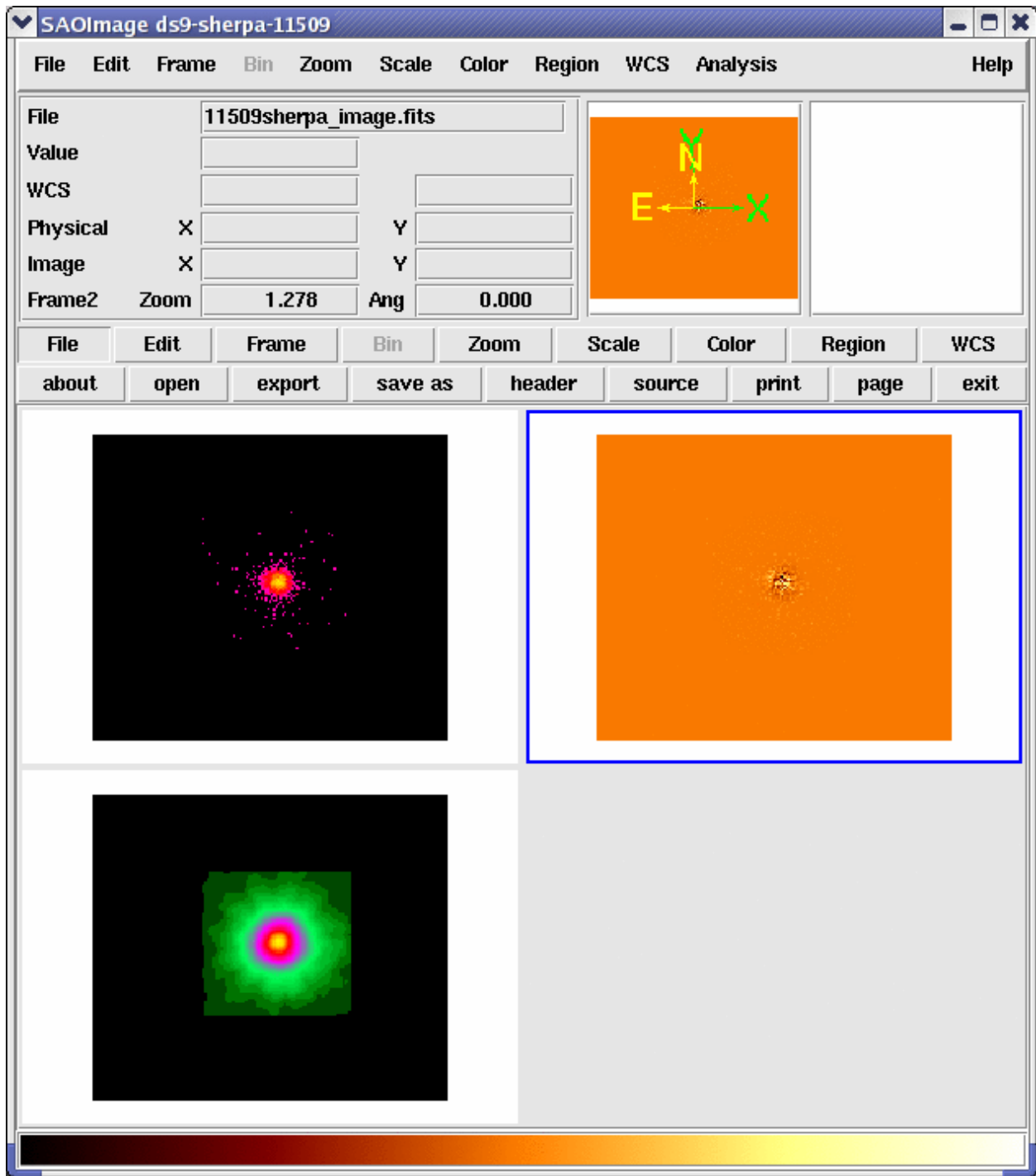
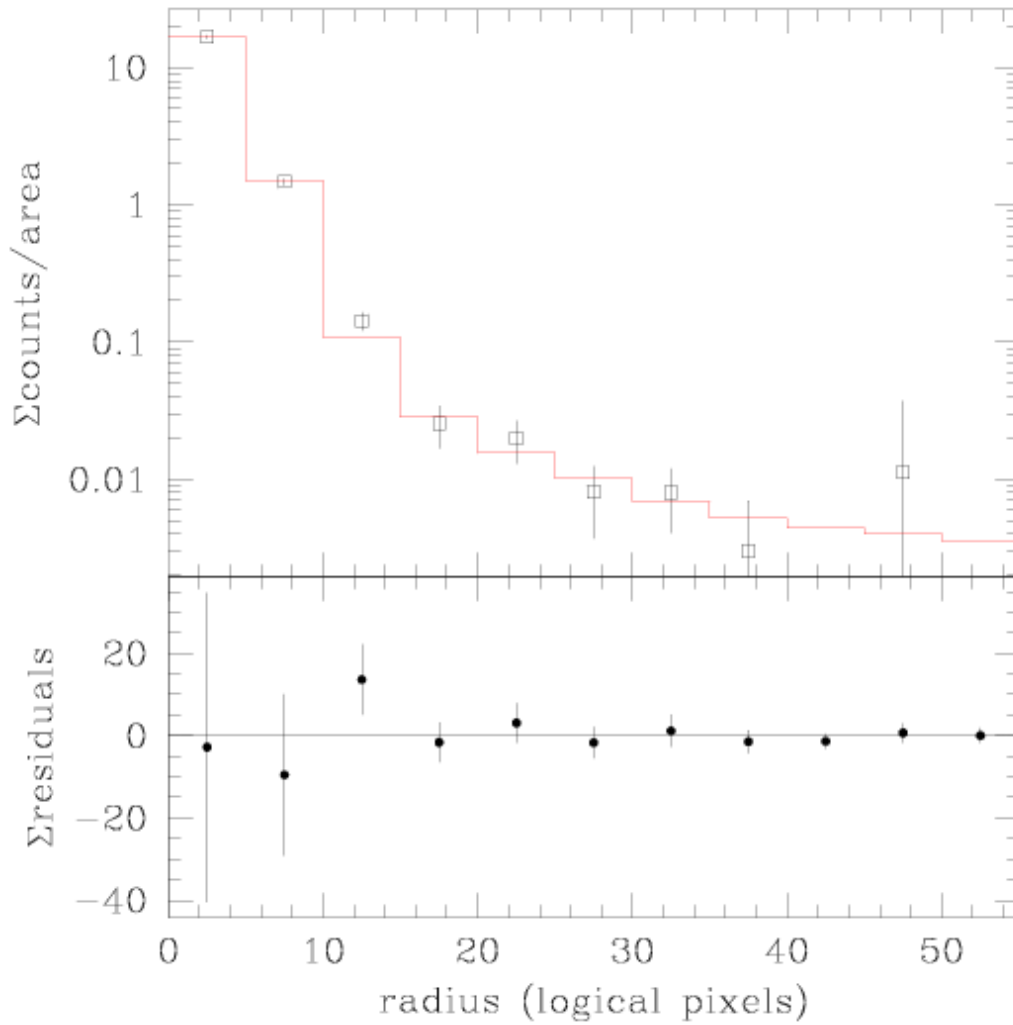


Image 6: Data, best-fit model, and residuals



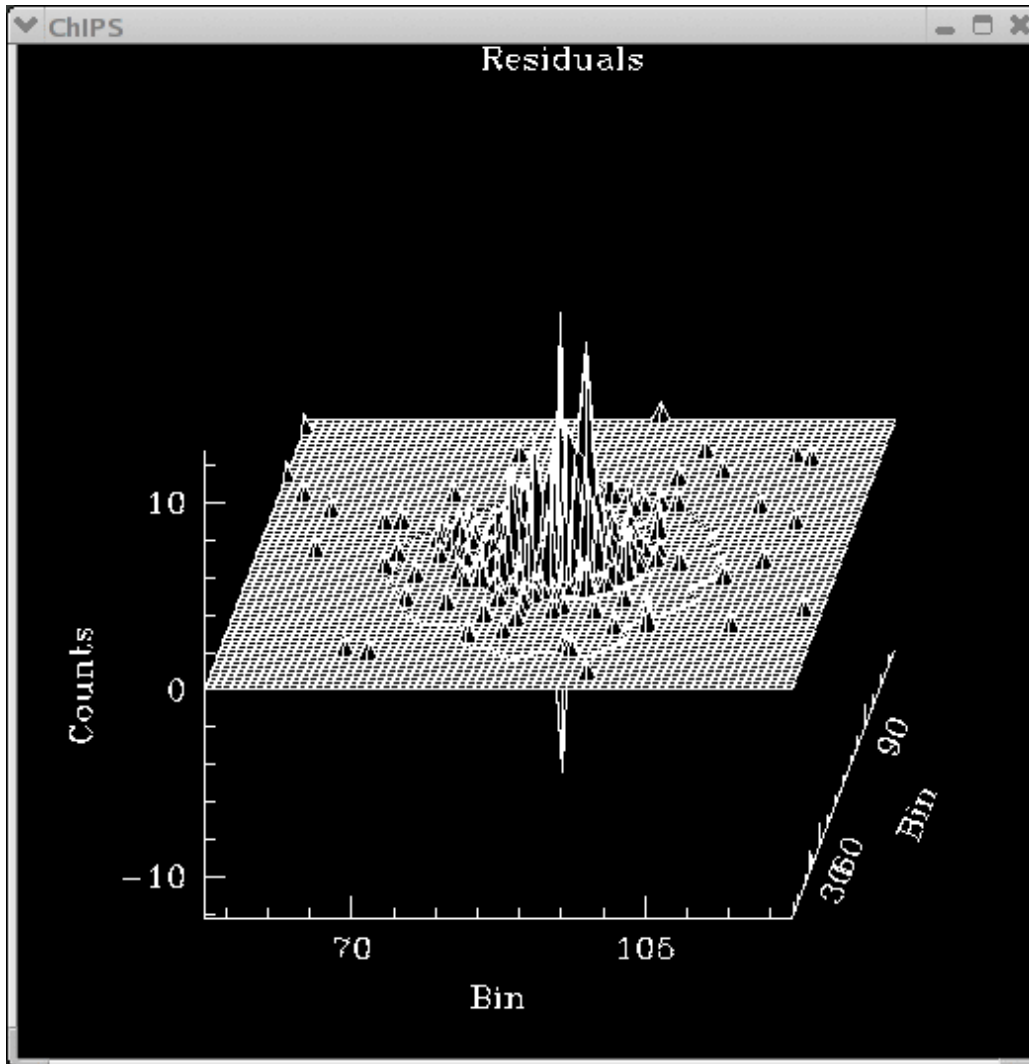
### Image 7: Radial profile of the best fit model

(cc1 + g2) @ 88.58,77.24





**Image 8: Surface plot of the residuals**



**Image 9: Contour plot of the residuals**

