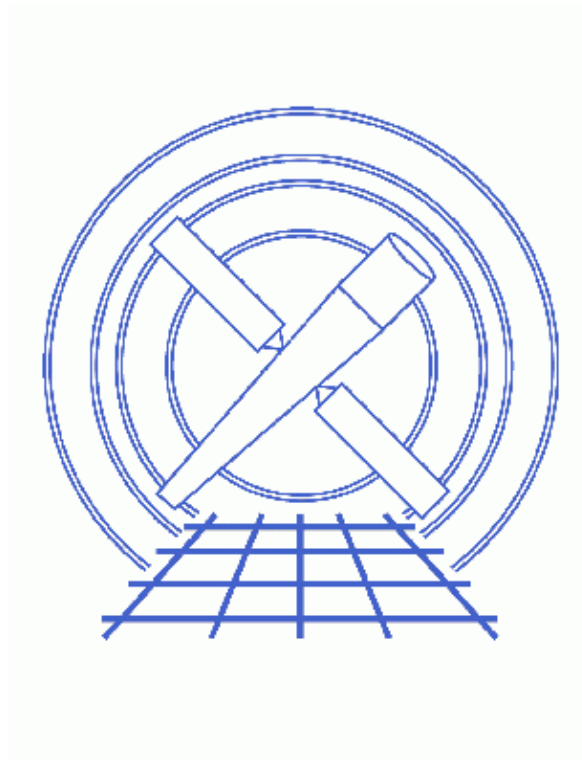


Create a PSF – CIAO 3.4

Create a PSF



CIAO 3.4 Science Threads

Table of Contents

- *The PSF Libraries*
- *Get Started*
- *Characterizing the Source*
 - ◆ What is the energy of the source? (dmextract)
 - ◆ How far off axis is my source? (dmstat)
 - ◆ Number of photons in the source (dmstat)
- *Create a PSF Image File (mkpsf)*
- *Normalize PSF to total counts in source (dmstat, dmimgcalc)*
- *Caveats*
 - ◆ Energy
 - ◆ Position
 - ◆ Common Errors
- *Parameter files:*
 - ◆ dmextract
 - ◆ dmstat
 - ◆ mkpsf
 - ◆ dmimgcalc
- *History*
- *Images*
 - ◆ Source region defined on the image
 - ◆ Source Spectrum
 - ◆ Image of the PSF at an energy of 3 keV
 - ◆ PSFs extracted at 3 keV, 1.4967 keV, and 4.51 keV
 - ◆ Image of off-axis PSF

Create a PSF

CIAO 3.4 Science Threads

Overview

Last Update: 1 Dec 2006 – updated for CIAO 3.4: ChIPS version

Synopsis:

The Chandra Ray Tracer (ChaRT) gives the best available HRMA Point Spread Function (PSF) for a point source at any off-axis angle and for any energy or spectrum. The PSF libraries are used in this thread to create a PSF for a specific observation. A comparison of these methods is available from the ChaRT webpages.

Purpose:

To create an image of the PSF for a source, and normalize it to the source flux. The PSF changes with source position and photon energy, and is created by interpolation of a library of pre-launch calibration files (the PSF hypercube library).

Read this thread if:

you would like to create a PSF for an HRC or ACIS imaging observation.

Related Links:

- Analysis Guide: Extended Sources
- Data Caveats for Responses
- The PSF Library page
- PSF HRMA calibration

Proceed to the HTML or hardcopy (PDF: A4 / letter) version of the thread.

The PSF Libraries

To create a PSF for a source, the PSF library for the instrument of interest (i.e. ACIS or HRC) must be installed. These libraries are supplied as part of the CALDB and can be downloaded if they are not already available. The ACIS and HRC libraries are stored in `$CALDB/data/chandra/acis/cpf/2dpsf/` and `$CALDB/data/chandra/hrc/cpf/2dpsf/` respectively:

```
unix% ls -l $CALDB/data/chandra/acis/cpf/2dpsf/  
README.txt  
acisi1998-11-052dpsf1N0002.fits  
acisi1998-11-052dpsf2N0002.fits
```

```

acisi1998-11-052dpsf3N0002.fits
acisi1998-11-052dpsf4N0002.fits
aciss1998-11-052dpsf1N0002.fits
aciss1998-11-052dpsf2N0002.fits
aciss1998-11-052dpsf3N0002.fits
aciss1998-11-052dpsf4N0002.fits
cxcpsflib.manual.ps@

unix% ls -l $CALDB/data/chandra/hrc/cpf/2dpsf/
README.txt
cxcpsflib.manual.ps@
hrci1998-11-052dpsf1N0002.fits
hrci1998-11-052dpsf2N0002.fits
hrci1998-11-052dpsf3N0002.fits
hrci1998-11-052dpsf4N0002.fits
hrcls1998-11-052dpsf1N0002.fits
hrcls1998-11-052dpsf2N0002.fits
hrcls1998-11-052dpsf3N0002.fits
hrcls1998-11-052dpsf4N0002.fits

```

The PSF changes with both the energy of the incoming photons and their location on the focal plane of the instrument, as described in the [Chandra Proposers' Observatory Guide](#). Each library contains a grid of PSF images covering a range of energies and locations on the detector plane; linear interpolation of these images is used to create the requested PSF.

Currently the PSF library is evaluated at five energies – 0.277 keV, 1.4967 KeV, 4.51 keV, 6.4 keV, and 8.6 keV – for a range of positions that depend on which library is used. A summary of the available libraries is available in the [PSF Library](#) section of the CIAO dictionary. The recommended libraries are the *f1* and *f2* files for each detector; use the *f1* library if the off-axis angle of the source is within the library's field of view, otherwise use the *f2* library (see the [PSF library manual](#) (PS, 16pp) for the area covered by each library).

Get Started

Sample ObsID used: 1838 (ACIS-S, G21.5-09)

File types needed: evt2; asol1


In this thread we only use data in the [energy range 0.3 to 8 keV](#); an image of the source region is also created:

```

unix% dmcop "acisf01838N001_evt2.fits[energy=300:8000]" acis_1838_evt2.fits
unix% dmcop "acis_1838_evt2.fits[bin x=3821:4320:1,y=4000:4499:1]" img_src_0.3-8keV.fits

```

Characterizing the Source

Several of the following steps require that you have a source region defined for your observation. [Figure 1](#)  shows the event file display in ds9 with the source region overlaid. The "Region -> Get Info..." dialog box, shown in the image, can be used to view and adjust the size of the region you created.

What is the energy of the source? (dmextract)

For the dataset we are using, the source has a maximum at sky coordinates of (4069.5,4250.5). We use the [dmextract](#) tool to create an energy histogram of all photons that fall within 20 pixels of this location. Since we are not going to use this for spectral analysis, we set the output type to [generic](#), and use a bin size of 0.1 keV to improve the signal-to-noise ratio:

Create a PSF – CIAO 3.4

```
unix% punlearn dmextract
unix% pset dmextract infile="acis_1838_evt2.fits[sky=circle(4069.5,4250.5,20)][bin energy=300:8000:100]"
unix% pset dmextract outfile=energy_histogram.fits
unix% pset dmextract opt=generic
unix% dmextract
Input event file (acis_1838_evt2.fits[sky=circle(4069.5,4250.5,20)][bin energy=300:8000:100]):
Enter output file name (energy_histogram.fits):
```

You can check the `dmextract` parameter file that was used with `plis` `dmextract`.

There are a number of ways to examine the energy histogram, as shown in the [Single Chip ACIS Exposure Map](#) thread. Here we use *ChIPS* to create a plot of energy vs. counts:

```
unix% chips

Welcome to ChIPS, version CIAO 3.4
Copyright (C) 1999–2003, Smithsonian Astrophysical Observatory

chips> curve "energy_histogram.fits[cols energy,counts]"
chips> simpleline
chips> xlabel "Energy (eV)"
chips> ylabel "Counts"
```

The resulting plot is shown in [Figure 2](#). We choose to evaluate the PSF at an energy of 3 keV. Exit *ChIPS* before continuing:

```
chips> exit
```

How far off axis is my source? (dmstat)

`mkpsf` uses the input SKY coordinates to determine how far off-axis the source is (see the [Create a PSF Image File](#) section), but it may be of interest to determine the off-axis angle for yourself as well. The location of the source in the focal plane of the detector can be found by using `dmcoords`:

```
unix% punlearn dmcoords
unix% pset dmcoords asolfile=pcadf084244404N001_asol1.fits
unix% dmcoords acis_1838_evt2.fits option=sky x=4069.5 y=4250.5
unix% pget dmcoords theta
1.282061482327381
```

The off-axis angle of the source is 1.282 arcminutes.

Number of photons in the source (dmstat)

For this example, we use a simple method for estimating the number of source photons; namely `dmstat` with a circular aperture (radius of 80 pixels) to define the source counts and an annulus of width 10 pixels outside this for the background:

```
unix% punlearn dmstat
unix% pset dmstat centroid=no sigma=no
unix% dmstat "img_src_0.3-8keV.fits[sky=circle(4069.5,4250.5,80)]"
EVENTS_IMAGE
  min:      0          @:      ( 4069.5 4170.5 )
  max:      81          @:      ( 4069.5 4250.5 )
  mean:     1.1467556397
  sum:      23028
  good:     20081
  null:     5840

unix% dmstat "img_src_0.3-8keV.fits[sky=annulus(4069.5,4250.5,80,90)]"
EVENTS_IMAGE
```

Create a PSF – CIAO 3.4

```
min:      0          @:      ( 4069.5 4160.5 )
max:      2          @:      ( 4022.5 4179.5 )
mean:     0.05896577381
sum:      317
good:     5376
null:     27385
```

Notice that it is no longer necessary to add the [opt null=-999] filter to the `dmstat` command; the tool now only checks values inside the region when computing statistics. You can check the `dmstat` parameter file that was used with `plist dmstat`.

The number of source photons is therefore $23028 - 3.7647 * 317 = 21835$. Note that the factor 3.7647 is used to scale the background counts to match the source count area, and is therefore the ratio of the source area to the background area; *if you change your source or background area, then you'll have to change this number*:


```
pi * r_src^2 / ( pi * ( r_out^2 - r_in^2 ) )
= 80^2 / ( 90^2 - 80^2 ) = 3.76471
```

The regions used for calculating the source and background signals are accurate for this example case; they should be changed to match the particulars of each dataset. For example, the background region above would be incorrect for a large, extended source.

Create a PSF Image File (mkpsf)

The CIAO tool `mkpsf` creates a PSF image. If the requested coordinates and energy do not match those in the PSF library, then the output image is constructed by linearly interpolating the library data. We shall use the *fl* ACIS-S library for the sky coordinates (4069.5,4250.5) and evaluate it at an energy of 3.0 keV. The pixel size and roll angle of the output image are taken from the `infile` parameter.

```
unix% punlearn mkpsf
unix% pset mkpsf coord=SKY x=4069.5 y=4250.5 energy=3.0
unix% pset mkpsf psflibfile=$CALDB/data/chandra/acis/cpf/2dpsf/aciss1998-11-052dpsf1N0002.fits
unix% pset mkpsf infile=img_src_0.3-8keV.fits
unix% pset mkpsf outfile=psf_3keV.fits
unix% pset mkpsf rotpts=9
unix% mkpsf
input coordinate system (SKY|DET) (SKY):
PSF binning in x direction (0.25:256.0) (INDEF):
PSF binning in y direction (0.25:256.0) (INDEF):
PSF size in x direction (2:2048) (INDEF):
PSF size in y direction (2:2048) (INDEF):
input file (img_src_0.3-8keV.fits):
energy in keV (0) (3):
x (4069.5):
y (4250.5):
PSF library file (/soft/ciao/CALDB/data/chandra/acis/cpf/2dpsf/aciss1998-11-052dpsf1N0002.fits):
output file (psf_3keV.fits):
psflib data output basename ( '.' to use output file ) ():
File psf_3keV.fits was created
```

The output image (`psf_3keV.fits`) is shown in [Figure 3](#) . You can check the parameter file that was used with `plist mkpsf`.

Normalize PSF to total counts in source (dmstat, dmimgcalc)

First, use `dmstat` to find the "signal" in the PSF image:

```
unix% dmstat psf_3keV.fits
AXAF_2DPSF
  min:      0                @:      ( 3942 4123 )
  max:      0.33929860592    @:      ( 4069 4250 )
  mean:     2.7752512972e-05
  sum:    1.8187886902
  good:     65536
  null:     0
```

and then `dmimgcalc` to normalize this image to the source counts (here 21835). The PSF image is multiplied by 12005 (= 21835/1.818789):

```
unix% punlearn dmimgcalc
unix% pset dmimgcalc infile=psf_3keV.fits infile2=none
unix% pset dmimgcalc weight=12005
unix% pset dmimgcalc operation=add
unix% pset dmimgcalc out=psf_3keV_norm.fits
unix% dmimgcalc
Input file #1 (psf_3keV.fits):
Input file #2 (none):
output file (psf_3keV_norm.fits):
arithmetic operation (add):
```

You can check the parameter file that was used with plist dmimgcalc.

To check the signal in the normalized PSF model:

```
unix% dmstat psf_3keV_norm.fits
psf_3keV_norm.fits
  min:      0                @:      ( 3942 4123 )
  max:      4073.2797641    @:      ( 4069 4250 )
  mean:     0.33316891823
  sum:      21834.558225
  good:     65536
  null:     0
```


If the absolute normalization of the PSF is important, make sure that you read about the limitations of the PSF library.

Caveats

As discussed in the PSF library manual (PS, 16pp), the libraries contain weight information to account for the finite domain of the data (i.e. the fact that not all the simulated photons fell within the regions stored in the library). However, `mkpsf` currently *cannot* access this weight information, and so cannot account for the total number of photons used to create the PSF. Users should beware of this limitation if the normalization of the PSFs (i.e. the fraction of PSF photons that fall within the output region) is important (e.g. when attempting to calculate encircled energy fractions, calculating the PSF profile at large distances from a source).

Unless you specify a position and energy that corresponds to one of the grid points in the PSF libraries, `mkpsf` will use linear interpolation to create the model PSF. In this case, the model can only be considered an approximation to the true PSF, and must be used with care.

Energy

The two nearest energies to the energy used in this thread (3 keV) are 1.4967 keV and 4.51 keV. [Figure 4](#)  shows the PSFs extracted at these energies; the circle has the same radius in all three images.

The energies used to evaluate the PSFs are stored in the ENERGY_BINS block of the library:

```
unix% dmlist "$CALDB/data/chandra/acis/cpf/2dpsf/aciss1998-11-052dpsf1N0002.fits[ENERGY_BINS]" data
-----
Data for Table Block ENERGY_BINS
-----
ROW      ENERGY_BIN  ENERGY
-----
1         1           0.2770
2         2           1.49670
3         3           4.510
4         4           6.40
5         5           8.60
```

Position

The PSF library contains PSFs evaluated at a number of spatial locations (grid points) in the azimuth and elevation (off-axis angle) coordinate system (as discussed in the [PSF library manual](#) (PS, 16pp) and the [CIAO dictionary](#)). To find the nearest grid points to your source, convert the polar coordinates (PHI, THETA) to their cartesian values (azimuth, elevation). For the source used above, we have:

```
PHI      = 6.75 deg
THETA    = 1.282 arcmin
```

which is

```
azimuth  = 0.15 arcmin
elevation = 1.27 arcmin
```

The four nearest positions in the *fI* library are therefore (+1,0), (+2,0), (+1,+1), and (+2,+1) in (azimuth, elevation).

While you can extract the information directly from the AXAF_2DPSF block of the PSF library, the returned dataset will not be rotated to match your observation, the pixel size may not match, and the WCS information will not match that of your data. Here we show how to obtain the 4.510 keV ACIS-S PSF at the (+8,-4) position of the *fI* library.

First we need to know which part of the library to extract. Listing the psf library file shows that the first two axes of the library are the spatial dimensions, #3 is the defocus setting, #4 is the energy, and #5, #6 are the grid position.

```
unix% dmlist "$CALDB/data/chandra/acis/cpf/2dpsf/aciss1998-11-052dpsf1N0002.fits[AXAF_2DPSF]" cols
-----
Columns for Image Block AXAF_2DPSF
-----
ColNo  Name                               Unit           Type           Range
-----
1      AXAF_2DPSF[256,256,1,5,21,11]      Real4(256x256x1x5x21x11)  -Inf:+Inf
-----
Physical Axis Transforms for Image Block AXAF_2DPSF
```


Create a PSF – CIAO 3.4

```
-----  
Group# Axis#  
 1 1,2 PSF(PSFX) = (+0)[pixel] +(0.250)* ((#1)-(128.0))  
      (PSFY) (+0) (0.250) ((#2) (128.0))  
 2 3 Z = #3  
 3 4 AXIS4 = #4  
 4 5,6 DET(DETX) = (+4096.50)[pixel] +(121.9512)* ((#5)-(11.0))  
      (DETY) (+4096.50) (121.9512) ((#6) (6.0))  
-----
```

World Coordinate Axis Transforms for Image Block AXAF_2DPSF


```
-----  
Group# Axis#  
 1 1,2 RELPOS(Y) = (+0)[mm] +(0.0240)* (PSF(PSFX)-(0))  
      (Z) (+0) (0.0240) ( (PSFY) (0))  
 2 3 DEFOCUS = Z  
 3 4 ENERGY = AXIS4  
 4 5,6 MSC(THETA) = (+0)[deg] +TAN-P[(0.000136667)* (DET(DETX)-(4096.50))]  
      (PHI) (+0) (0.000136667) ( (DETY) (4096.50))  
 4 5,6 LSI(LSIY) = (+0)[mm] +(0.0240)* (DET(DETX)-(4096.50))  
      (LSIZ) (+0) (0.0240) ( (DETY) (4096.50))  
 4 5,6 AZEL(AZ) = (+0)[deg] +(0.000136667)* (DET(DETX)-(4096.50))  
      (EL) (+0) (0.000136667) ( (DETY) (4096.50))  
-----
```

From the listed transformations, we find that:

```
AZ = 0.000136667 * 121.9512 * ((#5)-(11.0)) (degrees)  
    = #5 - 11 (arcminutes)  
EL = 0.000136667 * 121.9512 * ((#6)-(6.0)) (degrees)  
    = #6 - 6 (arcminutes)
```

Therefore, (+8,-4) location corresponds to axis location (#5=19,#6=2) and an energy of 4.51 keV corresponds to #4=3 (from above). As there is only one defocus setting in the library, the following DM filter produces a (256,256,1,1,1) copy of the PSF image:

```
unix% dmcop \  
      "$CALDB/data/chandra/acis/cpf/2dpsf/aciss1998-11-052dpsf1N0002.fits[AXAF_2DPSF][#1=1:256,#  
      psflib_4.51keV_p8_m4.fits
```

The resulting image is shown in [Figure 5](#) .

Common Errors

A common mistake is to use the library of a different detector, for instance using aciss1998-11-052dpsf1N0002.fits when the ACIS-I array was at the telescope focus.

```
Parameters for /home/username/cxcds_param/dmextract.par
```

```
#-----  
#  
# DMEXTRACT -- extract columns or counts from an event list  
#  
#-----  
      infile = acis_1838_evt2.fits[sky=circle(4069.5,4250.5,20)][bin energy=300:8000:100] Input  
      outfile = energy_histogram.fits Enter output file name  
      (bkg = ) Background region file or fixed background (counts/pixel/s) suk  
      (error = gaussian) Method for error determination(poisson|gaussian|<variance file>  
      (bkgerror = gaussian) Method for background error determination(poisson|gaussian|<var
```

Create a PSF – CIAO 3.4

```
(bkgnorm = 1.0)           Background normalization
  (exp = )                Exposure map image file
  (bkgexp = )             Background exposure map image file
  (sys_err = 0)           Fixed systematic error value for SYS_ERR keyword
  (opt = generic)         Output file type: phal
  (defaults = ${ASCDS_CALIB}/cxo.mdb -> /soft/ciao/data/cxo.mdb) Instrument defaults file
  (wmap = )                WMAP filter/binning (e.g. det=8 or default)
  (clobber = no)          OK to overwrite existing output file(s)?
  (verbose = 0)           Verbosity level
  (mode = ql)
```

Parameters for /home/username/cxcds_param/dmstat.par

```
  infile = img_src_0.3-8keV.fits[sky=annulus(4069.5,4250.5,80,90)] Input file specification
  out_columns = EVENTS_IMAGE      Output Column Label
  out_min = 0                     Output Minimum Value
  out_min_loc = 4069.5,4160.5     Output Minimum Location Value
  out_max = 2                     Output Maximum Value
  out_max_loc = 4022.5,4179.5     Output Maximum Location Value
  out_mean = 0.05896577381        Output Mean Value
  out_median =                    Output Median Value
  out_sigma =                     Output Sigma Value
  out_sum = 317                   Output Sum of Values
  out_good = 5376                 Output Number Good Values
  out_null = 27385                Output Number Null Values
  out_cnvrqd =                    Converged?
  out_cntrd_log =                 Output Centroid Log Value
  out_cntrd_phys =                Output Centroid Phys Value
  out_sigma_cntrd =               Output Sigma Centroid Value
  (centroid = no)                 Calculate centroid if image?
  (median = no)                   Calculate median value?
  (sigma = no)                    Calculate the population standard deviation?
  (clip = no)                     Calculate stats using sigma clipping?
  (nsigma = 3)                    Number of sigma to clip
  (maxiter = 20)                  Maximum number of iterations
  (mode = ql)
```

Parameters for /home/username/cxcds_param/mkpsf.par

```
#
# MKPSF -- retrieve PSF from library for given (energy,x,y) position
#
#####
#
#   COORDINATE SYSTEM parameters
#
#   coord = SKY                    input coordinate system
#
#   PSF binning parameters
#
#   binspax = INDEF                PSF binning in x direction
#   binspay = INDEF                PSF binning in y direction
#
#   PSF size parameters
#
#   sizeoutx = INDEF               PSF size in x direction
#   sizeouty = INDEF               PSF size in y direction
#
#   input file parameters
```

Create a PSF – CIAO 3.4

```
#
#   infile = img_src_0.3-8keV.fits input file
#
#   PSF position parameters
#
#   energy = 3           energy in keV
#     x = 4069.5         x
#     y = 4250.5         y
#
#   PSF library file parameters
#
#   psfplibfile = /soft/ciao/CALDB/data/chandra/acis/cpf/2dpsf/aciss1998-11-052dpsf1N0002.fits PSF
#
#   output file parameters
#
#   outfile = psf_3keV.fits output file
#   outpsffile = psfplib data output basename ('.' to use output file)
#
#   pixlib geometry parameter file
#
#   (geompar = geom)      Parameter file for Pixlib Geometry files
#
#   PSF roll parameter
#
#   (rotpts = 9)          number of pixel points in x or y direction for rotation
#
# debug print control
#
#   (verbose = 0)         verbose mode
#
# system variables
#
#   (clobber = no)        overwrite existing output file?
#   (mode = ql)
```

Parameters for /home/username/cxcds_param/dmimgcalc.par

```
# parameter file for dmimgcalc
#   infile = psf_3keV.fits Input file #1
#   infile2 = none Input file #2
#   outfile = psf_3keV_norm.fits output file
#   operation = add arithmetic operation
#   (weight = 12005) weight for first image
#   (weight2 = 1) weight for second image
#   (lookupTab = ${ASCDS_CALIB}/dmmerge_header_lookup.txt -> /soft/ciao/data/dmmerge_header_looku
#   (clobber = no) delete old output
#   (verbose = 0) output verbosity
#   (mode = ql)
```

History

14 Dec 2004 updated for CIAO 3.2: include dmcoords asolfile parameter

12 Dec 2005 updated for CIAO 3.3: default value of dmextract error and bkgerror parameters is "gaussian"; changes to dmstat parameter file

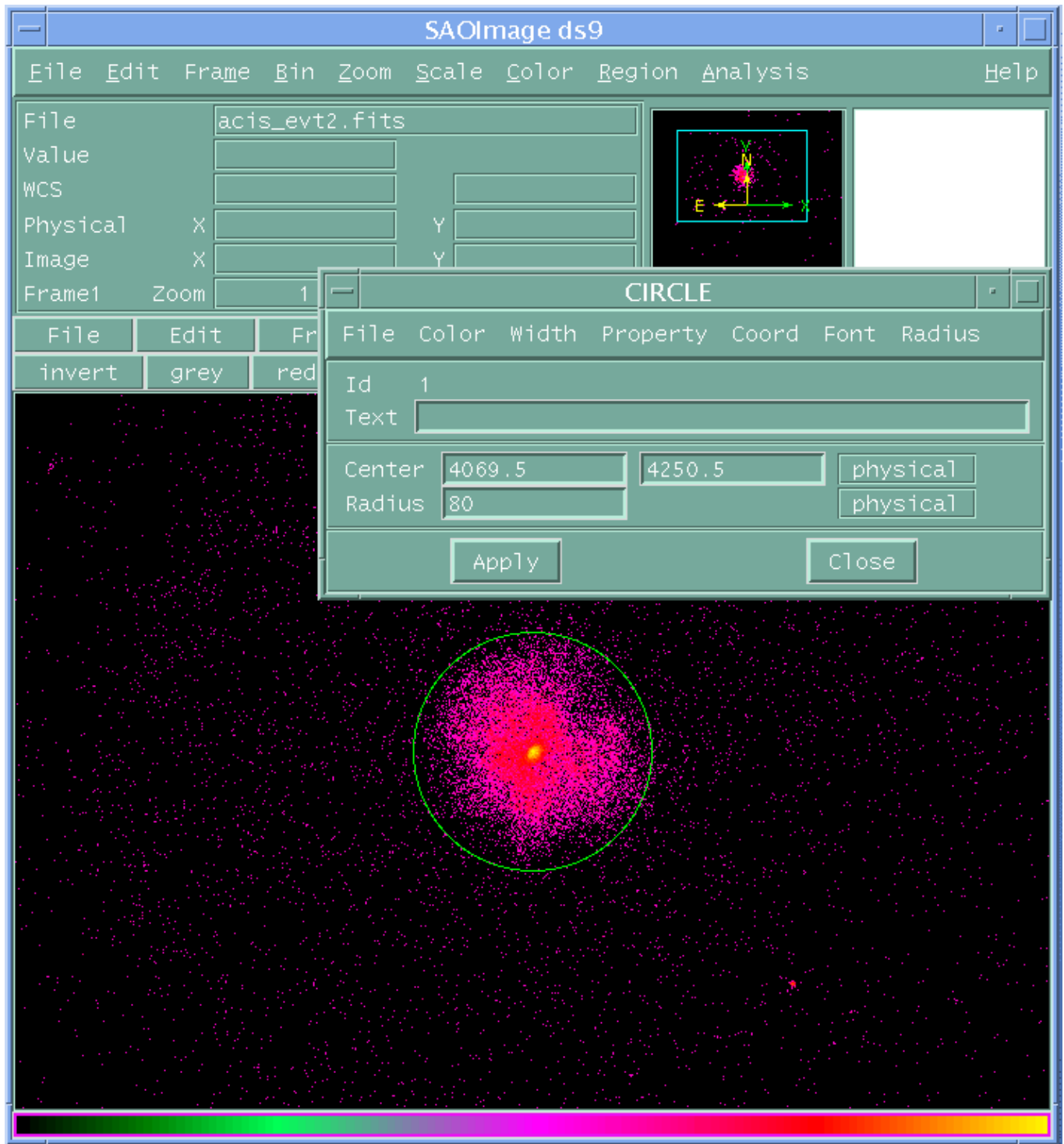
Create a PSF – CIAO 3.4

01 Aug 2006 corrected use of azimuth (ϕ) and elevation (also called the off-axis angle, θ)

01 Dec 2006 updated for CIAO 3.4: ChIPS version

URL: <http://cxc.harvard.edu/ciao/threads/psf/>

Last modified: 1 Dec 2006

Image 1: Source region defined on the image

The ``Region → Get Info...'' dialog box is overlaid on the main ds9 window. Use this GUI to adjust the size of the region for your observation.

Image 2: Source Spectrum

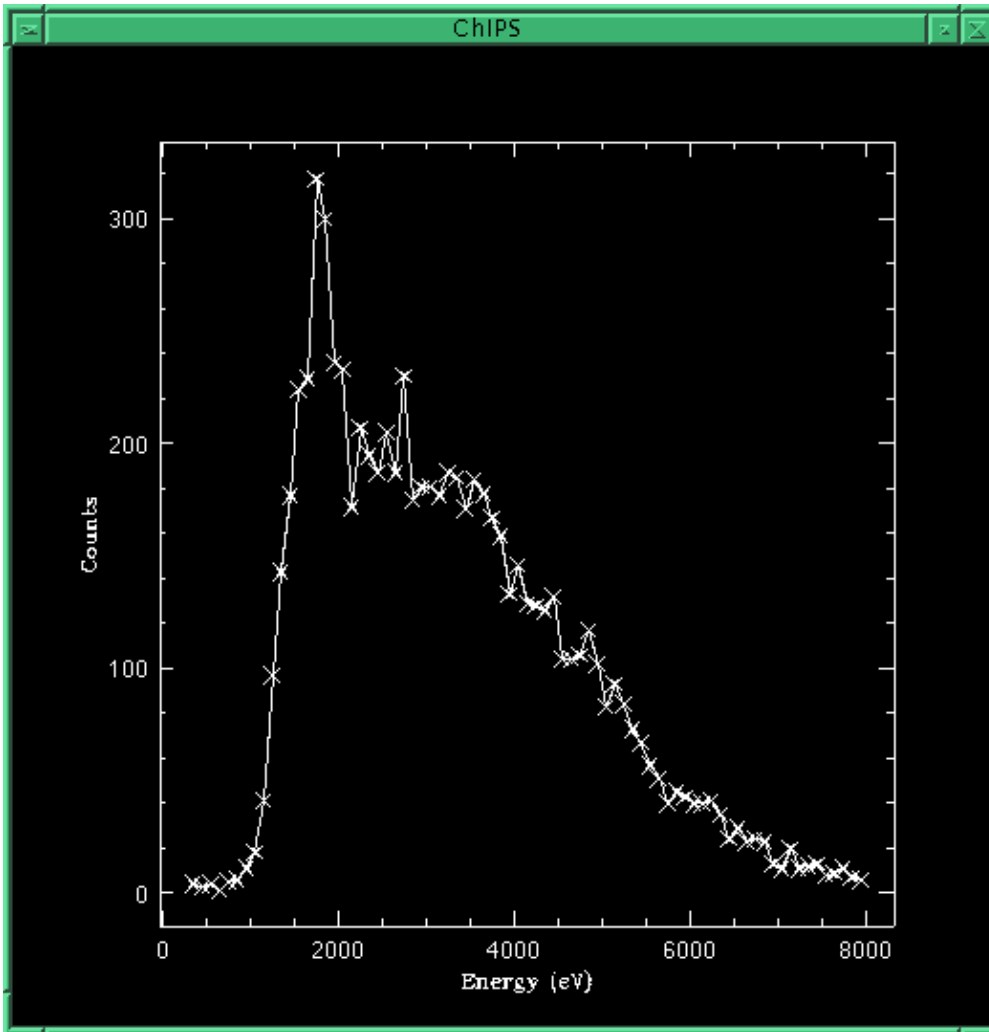


Image 3: Image of the PSF at an energy of 3 keV

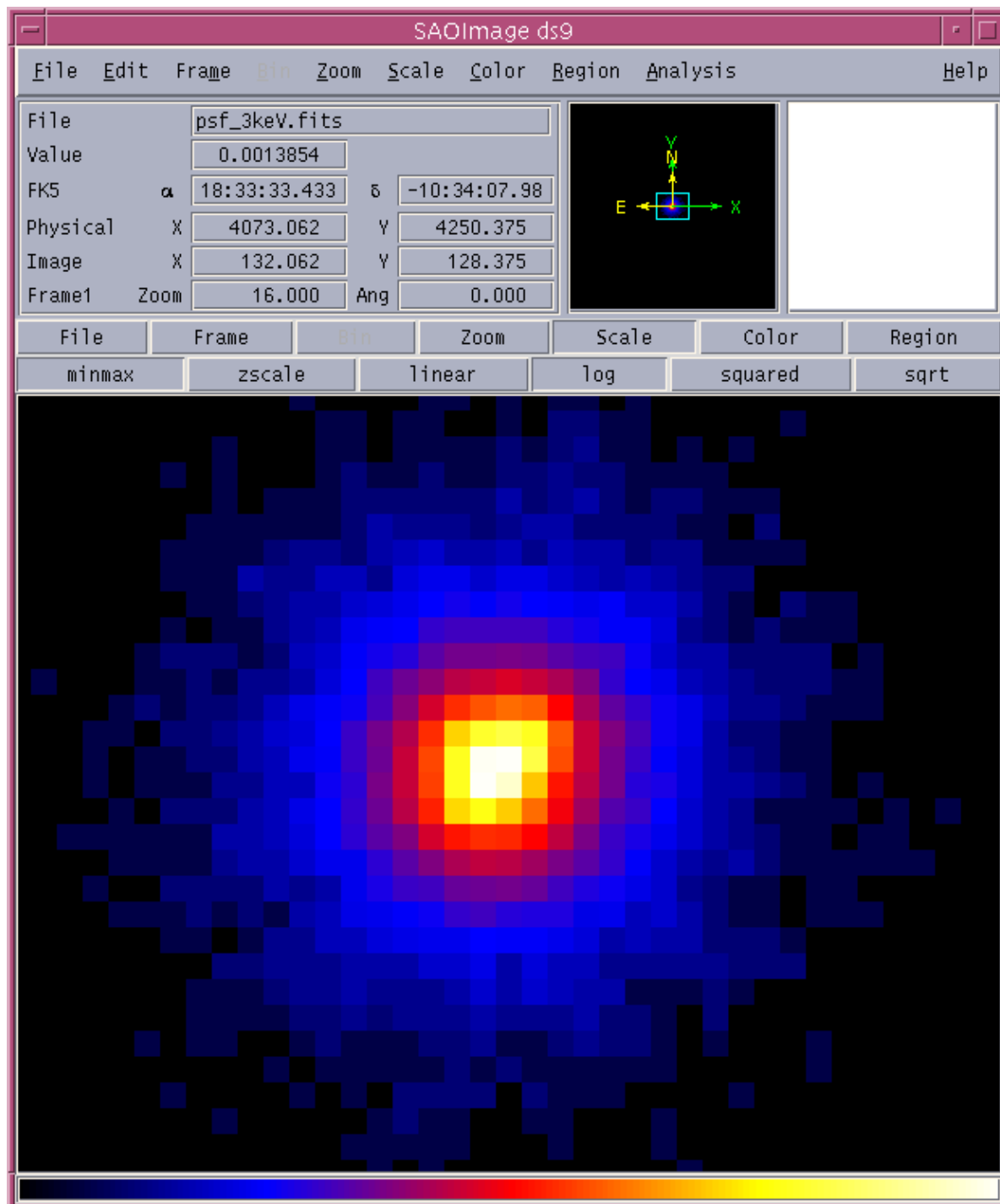


Image 4: PSFs extracted at 3 keV, 1.4967 keV, and 4.51 keV

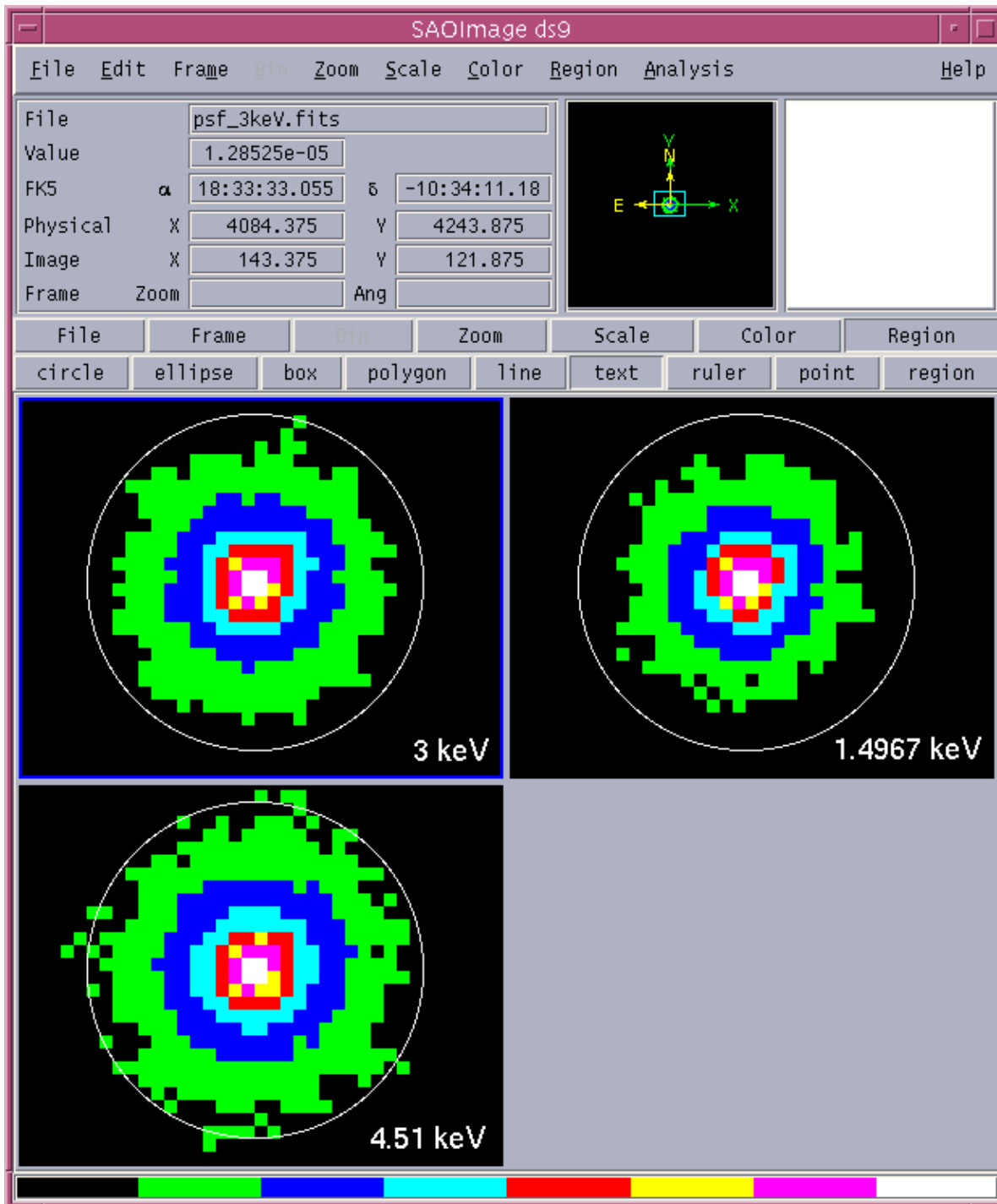


Image 5: Image of off-axis PSF

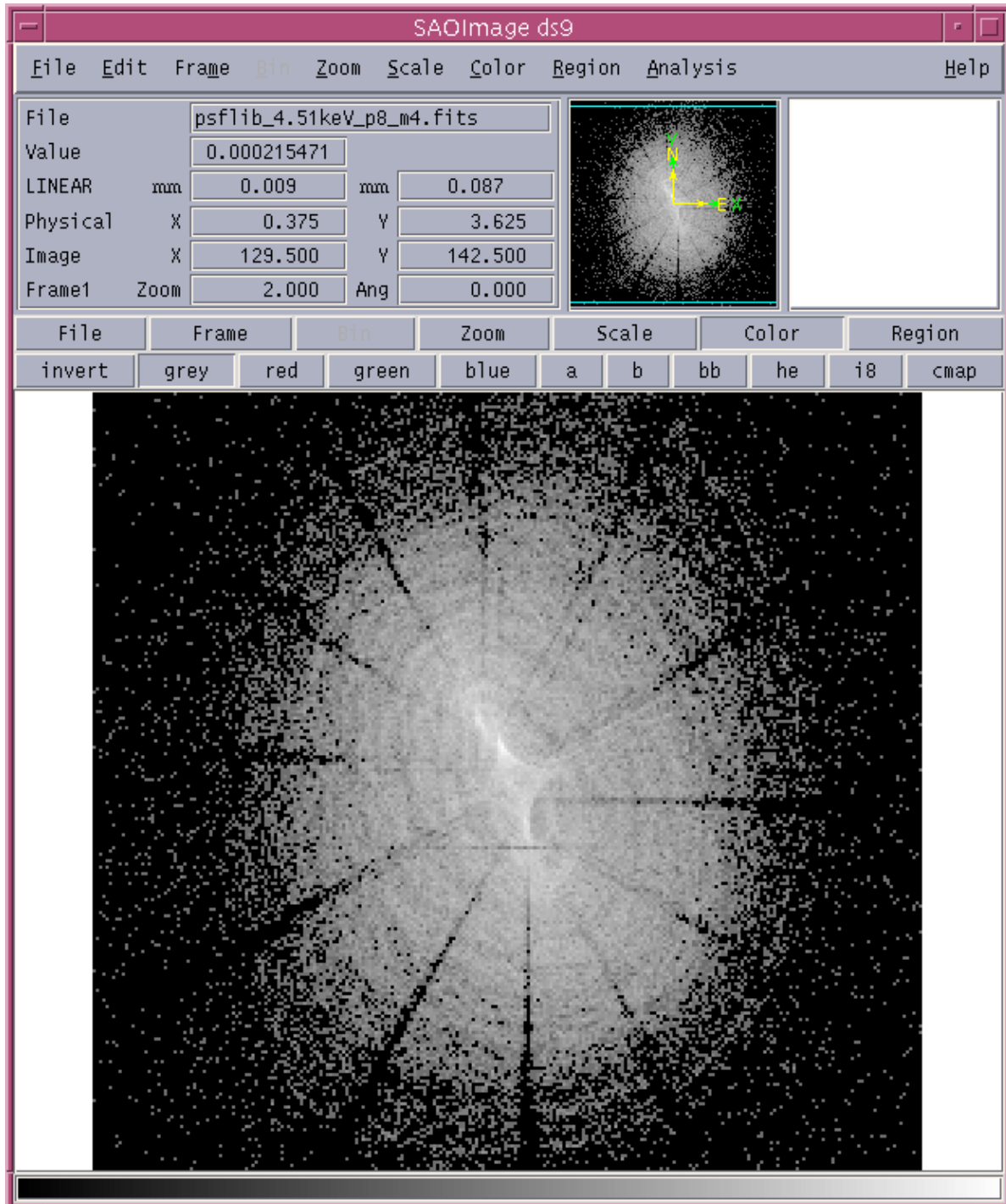


Image 5: Image of off-axis PSF

Create a PSF – CIAO 3.4