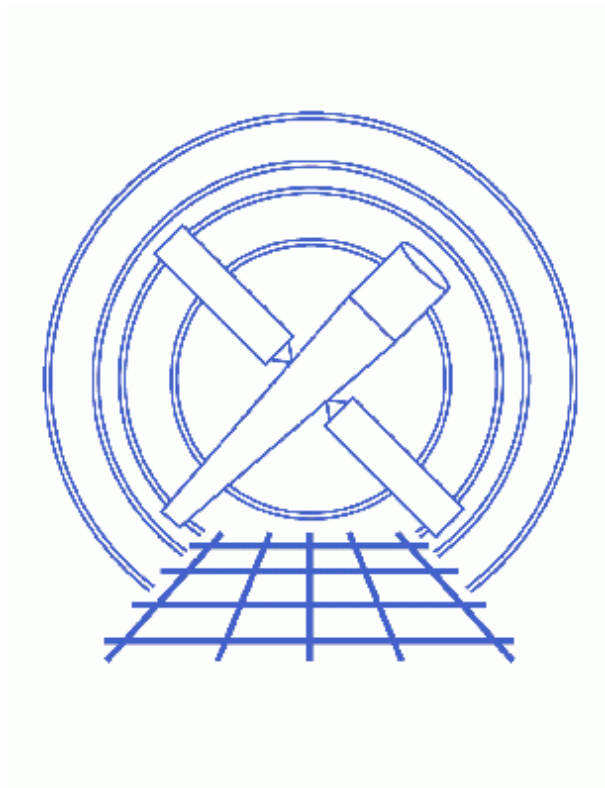


Compute an HRC-S Exposure Map and Build Fluxed Image



CIAO 3.4 Science Threads

Table of Contents

- ***Get Started***
 - ◆ Download get_sky_limits
- ***Create An Image***
- ***Compute Exposure Map***
 1. Compute the Aspect Histogram
 2. Calculate the Instrument Map
 3. Calculate the Exposure Map
 4. Combine the Exposure Maps
- ***Normalize the Image by the Exposure Map***
- ***Calculate the Source Flux***
- ***Parameter files:***
 - ◆ asphist
 - ◆ dmregrid
 - ◆ dmimgcalc
- ***History***
- ***Images***
 - ◆ Exposure map for HRC-S
 - ◆ Fluxed image for HRC-S

Compute an HRC–S Exposure Map and Build Fluxed Image

CIAO 3.4 Science Threads

Overview

Last Update: 23 May 2007 – new for CIAO 3.4

Synopsis:

`mkexpmap` generates an exposure map which may be used to convert a counts image of a source to an image in flux units. The computed exposure map is essentially an image of the effective area at each sky position, accounting for the effects of dither motion which are especially important near the edges of the detector.

Purpose:

To build an exposure map for an HRC–S observation, create a fluxed image, and find an approximation for the source flux.

Read this thread if:

you are working with an HRC–S observation and would like to create an exposure map for it.

Related Links:

- Analysis Guide: [HRC Imaging](#)
- Analysis Guide: [Extended Sources](#)
- [An Introduction to Exposure Maps](#) (PS, 12 pp): a general discussion of exposure maps.

Proceed to the [HTML](#) or hardcopy (PDF: [A4](#) | [letter](#)) version of the thread.

Get Started

Sample ObsID used: 990 (HRC–S, VEGA)

File types needed: evt2; dtf1; aso11; msk1

To create an exposure map, we will use an aspect histogram file, which contains information on the [aspect motion](#) during the observation; and an instrument map, which is essentially the product of the detector quantum efficiency

and the mirror effective area projected onto the detector surface.

Please ensure that you have set up ardlb to use the bad pixel file for your observation before following this thread; see the [Use Observation-specific Bad Pixel Files](#) thread for more information.

Download get_sky_limits

This thread uses the `get_sky_limits` script; for information about the script, consult the help file ("[ahelp get_sky_limits](#)"):

```
unix% grep Id: `which get_sky_limits`
% $Id: get_sky_limits,v 1.6 2004/11/02 16:22:30 dburke Exp $
```

Please check that you are using the most recent version before continuing. If you do not have the script installed or need to update to a newer version, please refer to the [Scripts page](#).

Create An Image

This thread creates an exposure map for all three plates of HRC-S. Some users will only need to use `chip_id=2` (the center plate), since it is most commonly used for HRC-S imaging.

First, we need to create the image which will ultimately be normalized by the exposure map. Here we blocked the image by a factor of 32:

```
unix% dmcop y \
    "hrcf00990N004_evt2.fits[bin x=0.5:65536.5:32,y=0.5:65536.5:32][opt_type=i4]" \
    990_img.fits
```

This creates an image that is 2048x2048; this information is used again in the [Calculate the Exposure Map](#) step. **You may choose to use different binning**, but make sure that you change the `xygrid` appropriately in [that step](#). Due to the size of the image, the output size is set to 4 byte integer ("`opt_type=i4`") instead of the default 2 byte integer.

Compute Exposure Map

1. Compute the Aspect Histogram

With the corrected aspect offsets file we can create a binned histogram, detailing the aspect history of the observation. Only one aspect histogram needs to be computed because the three plates share a single [GTI](#):

```
unix% dmlist hrcf00990N004_evt2.fits blocks
```

```
-----
Dataset: hrcf00990N004_evt2.fits
-----
```

| Block Name | Type | Dimensions |
|------------|------|------------|
| ----- | | |

HRC-S Exposure Map – CIAO 3.4

```
Block    1: PRIMARY                Null
Block    2: EVENTS                 Table          9 cols x 453434  rows
Block    3: GTI                   Table          2 cols x 1       rows
```

```
unix% punlearn asphist
unix% pset asphist infile=pcadf097387525N002_asoll.fits
unix% pset asphist outfile=asphist_hrcs.fits
unix% pset asphist evtfile=hrcf00990N004_evt2.fits
unix% pset asphist dtffile=hrcf00990_000N003_dtf1.fits
unix% asphist
Aspect Solution List Files (pcadf097387525N002_asoll.fits):
Aspect Histogram Output File (asphist_hrcs.fits):
Event List Files (hrcf00990N004_evt2.fits):
Live Time Correction List Files for HRC (hrcf00990_000N003_dtf1.fits):

# asphist (CIAO 3.4): WARNING: skipping 84 livetime correction records (from time: 97387355.142212 t
```

In some cases there will be more than one `asoll.fits` file for an observation. **All** the files must be input to the `infile` parameter, either as a list or as a stack (see [ahelp stack](#) for more information).

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

2. Calculate the Instrument Map

Since the mirror effective area is used to create the instrument map, and that area is energy dependent, it is necessary to decide at what energy to perform the calculation (or whether to use a spectrum as weights). Since energy is not explicitly resolved in HRC observations, the `monoenergy` parameter is determined at the discretion of the observer (the default value is 1 keV); this thread uses a value of 1.1 keV.

Note that it is not necessary for the instrument map to be congruent with the exposure map. We set the `pixelgrid` parameter to cover the entire detector area and bin by a factor of 8.

```
unix% punlearn mkinstmap
unix% pset mkinstmap obsfile="hrcf00990N004_evt2.fits[EVENTS]"
unix% pset mkinstmap pixelgrid="1:16384:#1024,1:16384:#1024"
unix% pset mkinstmap monoenergy=1.1
unix% pset mkinstmap mode=h
```

Now run the tool once for each chip:

```
unix% foreach d ( 1 2 3 )
foreach? mkinstmap detsubsys=HRC-S${d} outfile=instmap_hrcs_${d}.fits \
          maskfile="hrcf00990_000N003_msk1.fits[MASK${d}]"
foreach? end
```

3. Calculate the Exposure Map

Now we use `mkexpmap` and the aspect information stored in the histogram to project the instrument map onto the sky. The `get_sky_limits` script can be used to calculate the exposure map binning information from the existing image:

```
unix% get_sky_limits 990_img.fits verbose="1"
Checking binning of image: 990_img.fits
Image has 2048 x 2048 pixels
Lower left (0.5,0.5) corner is x,y= 0.5, 0.5
```

2. Calculate the Instrument Map

HRC-S Exposure Map – CIAO 3.4

```
Upper right (2048.5,2048.5) corner is x,y= 65536.5, 65536.5
DM filter is:
  x=0.5:65536.5:#2048,y=0.5:65536.5:#2048
mkexpmap xygrid value is:
  0.5:65536.5:#2048,0.5:65536.5:#2048
```

You can then set the `xygrid` parameter using the information provided by the script, either manually or via:

```
unix% pset mkexpmap xygrid=")get_sky_limits.xygrid"
```

(if the latter, do not run `get_sky_limits` again until after running `mkexpmap`).

If you are computing a low-resolution exposure map and speed is more important than accuracy, set `useavgaspect=yes`. In doing so, only the average aspect pointing will be used to derive the exposure map; otherwise all points in the aspect histogram will be used. The time required to compute the exposure map is proportional to the number of bins in the aspect histogram; if the aspect histogram contains 100 bins, then the use of this option reduces the run time by a factor of 100, approximately (you may also want to set `verbose` to 2, since this causes `mkexpmap` to output percentage-completed information). Using the full aspect solution will help accurately account for chip edges, bad pixels, etc.

```
unix% punlearn mkexpmap
unix% pset mkexpmap asphistfile=asphist_hrcs.fits
unix% pset mkexpmap xygrid="0.5:65536.5:#2048,0.5:65536.5:#2048"
unix% pset mkexpmap useavgaspect=no
unix% pset mkexpmap normalize=no
unix% pset mkexpmap mode=h

unix% foreach d ( 1 2 3 )
foreach? mkexpmap instmapfile=instmap_hrcs_${d}.fits \
          outfile=expmap_hrcs_${d}.fits
foreach? end
Exposure map limits: 0.000000e+00, 4.015039e+05
Writing exposure map to expmap_hrcs_1.fits
Exposure map limits: 0.000000e+00, 5.399535e+05
Writing exposure map to expmap_hrcs_2.fits
Exposure map limits: 0.000000e+00, 3.777550e+05
Writing exposure map to expmap_hrcs_3.fits
```

Since we set the `normalize` parameter to `no`, the exposure map has units of [$\text{cm}^2 \cdot \text{s} \cdot \text{counts}/\text{photon}$]. This allows us to simply divide the image by the exposure map to derive an image in units of flux ($[\text{photons}/\text{cm}^2/\text{s}/\text{pixel}]$). If the setting had been left as `yes` (the default), the units of the exposure map would be [$\text{cm}^2 \cdot \text{counts}/\text{photon}$]; see the [help file for `mkexpmap`](#) for more details on this.

4. Combine the Exposure Maps


The individual exposure maps are combined into a single, binned exposure map with the CIAO tool `dmregrid`. First we need a list of files to combine:

```
unix% ls expmap_hrcs_*.fits > expmap.lis
unix% cat expmap.lis
expmap_hrcs_1.fits
expmap_hrcs_2.fits
expmap_hrcs_3.fits
```

Now we can use this list by passing it into `dmregrid` as a stack:

```
unix% dmregrid infile=@expmap.lis outfile=expmap_hrcs.fits \
  bin="1:2048:1,1:2048:1" npts=1 \
```

```
rotangle=0 xoffset=0 yoffset=0 rotxcenter=0 rotycenter=0
```

The `bin` parameter value will need to be changed if you used a different binning specification when creating the exposure maps. The combined exposure map can be displayed in ds9 .

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

Normalize the Image by the Exposure Map

The exposure map is in units of $[\text{cm}^2 \cdot \text{s} \cdot \text{counts}/\text{photon}]$ since it was created by projecting the instrument map (in $[\text{cm}^2 \cdot \text{counts}/\text{photon}]$) onto the tangent plane of the observation. To create an image in units of $[\text{photon}/\text{cm}^2/\text{s}/\text{pixel}]$, we simply need to divide by the exposure map. This can be performed in one step with `dmimgcalc`:

```
unix% punlearn dmimgcalc
unix% pset dmimgcalc infile=990_img.fits
unix% pset dmimgcalc infile2=expmap_hrcs.fits
unix% pset dmimgcalc outfile=990_img_norm.fits
unix% pset dmimgcalc operation=div
unix% dmimgcalc
Input file #1 (990_img.fits):
Input file #2 (expmap_hrcs.fits):
output file (990_img_norm.fits):
arithmetic operation (div):
warning: CONTENT has 1 different values.
warning: DETNAM has different value...Merged...
```

The messages are related to how the tool merges the header information in the input files. The merging rules ahelp file explains the rules and how they affect the output file header.

The units of `990_img_norm.fits`  are $[\text{photon}/\text{cm}^2/\text{s}/\text{pixel}]$.

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

Calculate the Source Flux

Since the units of the fluxed image are $[\text{photon}/\text{cm}^2/\text{s}/\text{pixel}]$, adding up the pixel values around a source results in the source flux in $[\text{photon}/\text{cm}^2/\text{s}]$. Note that this flux is an *approximation* – as discussed in An Introduction to Exposure Maps (PS, 12pp) – since a spectral shape was assumed when using mkinstmap (in this example, a monochromatic source).

Using the source region "flux.reg":

```
unix% more flux.reg
# Region file format: CIAO version 1.0
circle(36496.5,29856.5,235)
```

the flux can be calculated with either dmstat:

HRC-S Exposure Map – CIAO 3.4

```
unix% dmstat infile="990_img_norm.fits[sky=region(flux.reg)]" centroid=no
990_img_norm.fits
  min:      2.2152180463e-05      @:      ( 36592.5 29648.5 )
  max:      0.079808160663      @:      ( 36496.5 29872.5 )
  mean:     0.0047503751901
  sigma:    0.013661690494
  sum:      0.78856228155
  good:     166
  null:     74
```

or dmextract:

```
unix% dmextract infile="990_img_norm.fits[bin sky=@flux.reg]" outfile="source_flux.fits"
unix% dmlist source_flux.fits"[cols COUNTS,ERR_COUNTS]" data
-----
Data for Table Block HISTOGRAM
-----

```

| ROW | COUNTS | ERR_COUNTS |
|-----|-------------------------|------------------|
| 1 | 0.78856228155382 | 0.88801029360803 |

Since the input to `dmextract` was a fluxed image, not an event list, the `COUNTS` column actually reports the total flux (in [photon/cm²/s]) for the source region. While slightly more involved, the `dmextract` method can be used on multiple sources in a single command, and the results are conveniently stored in a table.

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

```
infile = pcadf097387525N002_asol1.fits Aspect Solution List Files
outfile = asphist_hrcs.fits Aspect Histogram Output File
evtfile = hrcf00990N004_evt2.fits Event List Files
dtffile = hrcf00990_000N003_dtfl.fits Live Time Correction List Files for HRC
(geompar = geom)      Parameter file for Pixlib Geometry files
(res_xy = 0.5)        Aspect Resolution x and y in arcsec
(res_roll = 600.)     Aspect Resolution roll in arcsec
(max_bin = 10000.)    Maximal number of bins
(clobber = no)        Clobber output
(verbose = 0)         Verbose
(mode = ql)
```

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

```
##
## DMREGRID -- regrid image
##
infile = @expmap.lis      Input image
outfile = expmap_hrcs.fits Enter output file name
bin = 1:2048:1,1:2048:1 Binning specification
rotangle = 0              CCW rotation angle in degrees about rotation center
rotxcenter = 0            x coordinate of rotation center
rotycenter = 0            y coordinate of rotation center
```


HRC-S Exposure Map – CIAO 3.4

```
xoffset = 0          x offset
yoffset = 0          y offset
  npts = 1           Number of points in pixel (0='exact' algorithm)
(coord_sys = logical) Coordinate system of bin parameter
(clobber = no)       OK to overwrite existing output file(s)?
(verbose = 0)         Verbosity level (0 = no display)
  (mode = ql)
```

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

```
# parameter file for dmimgcalc
  infile = 990_img.fits      Input file #1
  infile2 = expmap_hrcs.fits Input file #2
  outfile = 990_img_norm.fits output file
operation = div              arithmetic operation
  (weight = 1)              weight for first image
  (weight2 = 1)             weight for second image
(lookupTab = ${ASCDS_CALIB}/dmmerge_header_lookup.txt -> /soft/ciao/data/dmmerge_header_lookup.t
(clobber = no)              delete old output
(verbose = 0)               output verbosity
  (mode = ql)
```

History

23 May 2007 new for CIAO 3.4

URL: http://cxc.harvard.edu/ciao/threads/expmap_hrcs/

Last modified: 23 May 2007

Image 1: Exposure map for HRC-S

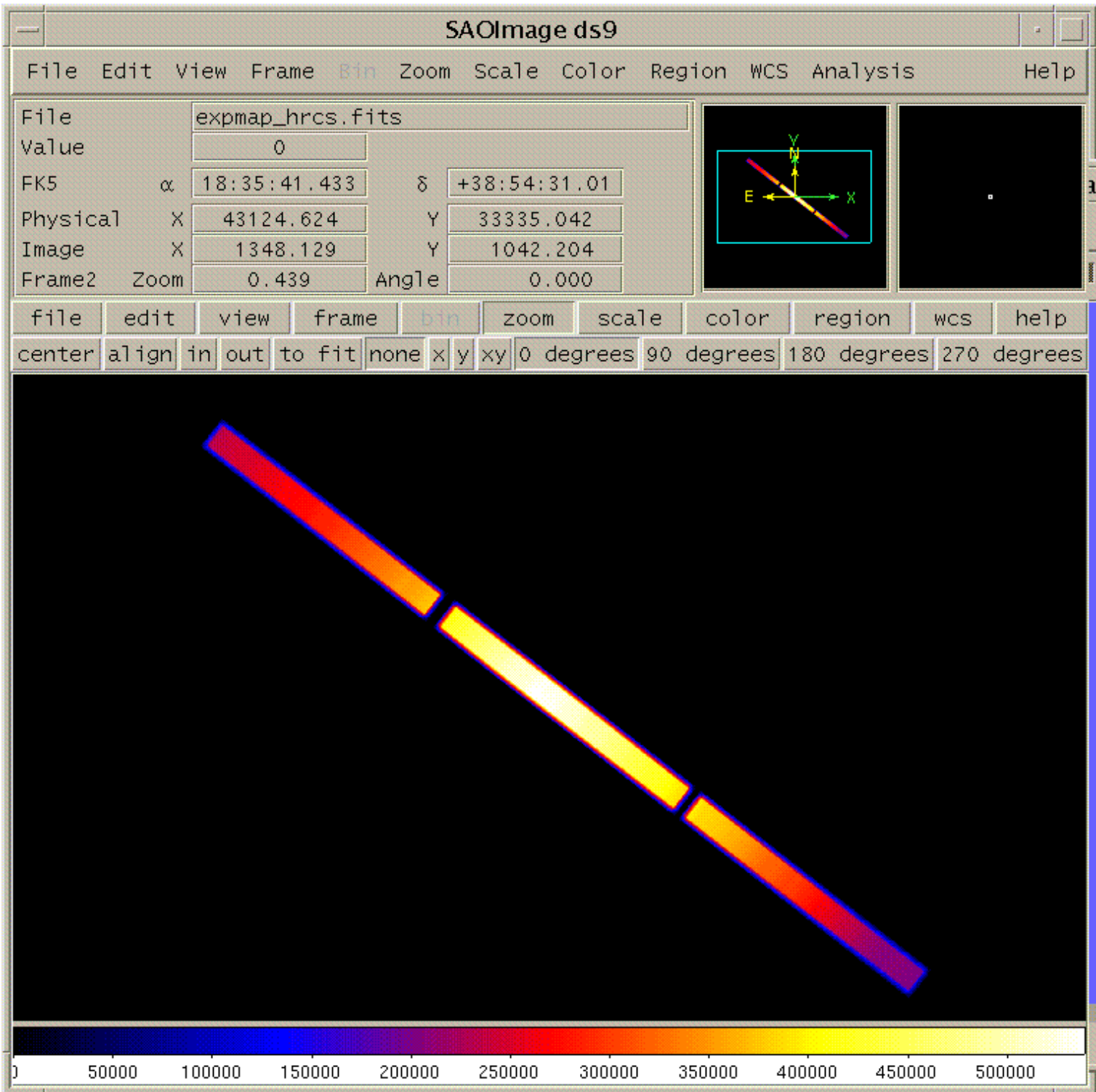
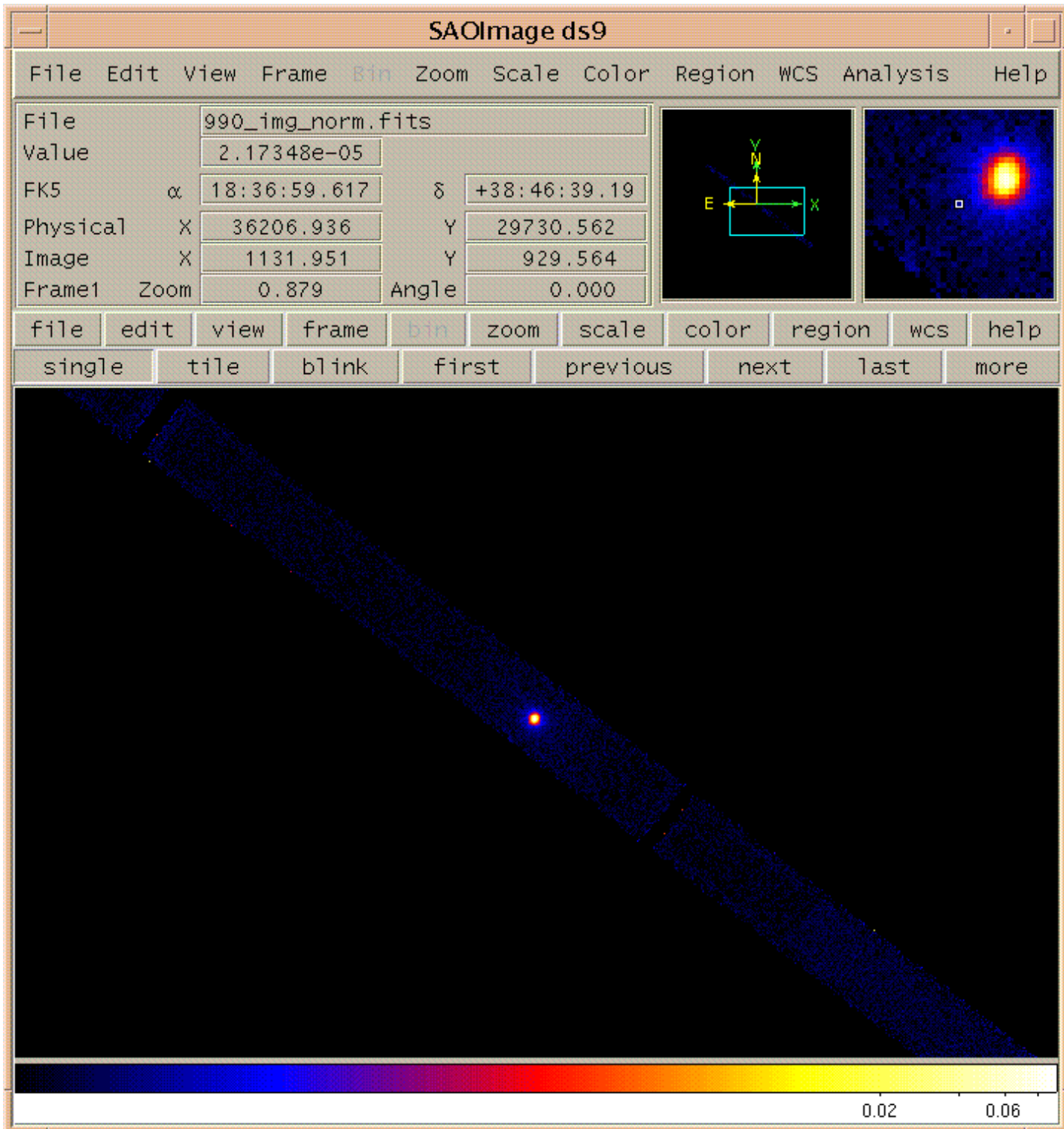


Image 2: Fluxed image for HRC-S



HRC-S Exposure Map – CIAO 3.4