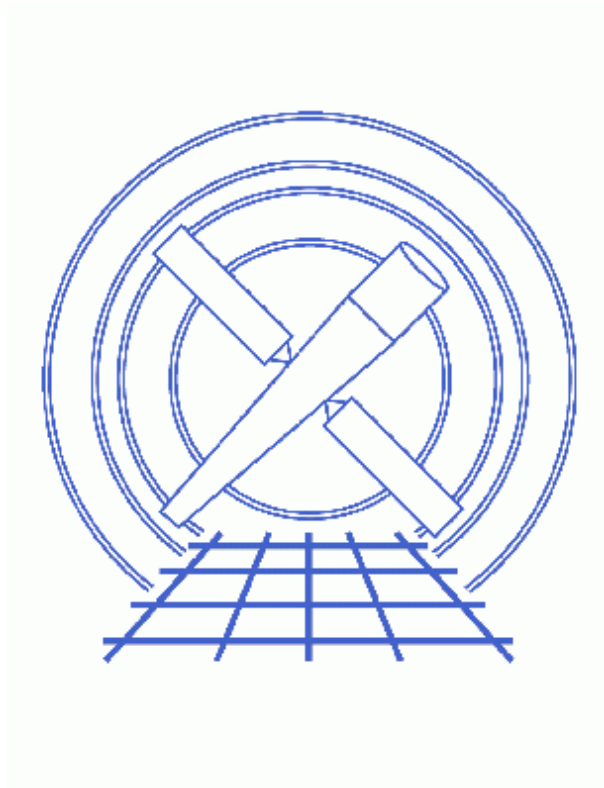


Compute an HRC-I Exposure Map and Build Fluxed Image



CIAO 3.4 Science Threads

Table of Contents

- *Get Started*
- *Create An Image*
- *Compute Exposure Map*
 1. *Compute the Aspect Histogram*
 2. *Calculate the Instrument Map*
 3. *Calculate the Exposure Map*
- *Normalize the Image by the Exposure Map*
- *Calculate the Source Flux*
- *Parameter files:*
 - ◆ *asphist*
 - ◆ *mkinstmap*
 - ◆ *mkexpmap*
 - ◆ *dmimgcalc*
 - ◆ *dmimgthresh*
- *History*
- *Images*
 - ◆ *Histogram of the observation dither*
 - ◆ *Exposure map for HRC-I*
 - ◆ *"Rings" in the fluxed image*
 - ◆ *Fluxed image for HRC-I*

Compute an HRC-I Exposure Map and Build Fluxed Image

CIAO 3.4 Science Threads

Overview

Last Update: 1 Dec 2006 – updated for CIAO 3.4: CIAO version in warning; parameter file update for `mkinstmap`

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-I observation, create a fluxed image, and find an approximation for the source flux.

Read this thread if:

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

Calibration Updates:

- **CALDB v2.17 (1 Oct 2002):** A new HRC_QE file (`hrCID1999-07-22qeN0006.fits`) was added to the CALDB. This file will automatically be picked up by the tool `mkinstmap` via `ardlib.par`.

Related Links:

- Analysis Guide: [HRC Imaging](#)
- Analysis Guide: [Extended Sources](#)
- [An Introduction to Exposure Maps](#) (PS, 12 pp): a general discussion of exposure maps.
- [Calculating Spectral Weights](#) thread: if you are interested in apply spectral weights to your data at the [Calculate the Instrument Map](#) step, run this thread first.

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

Get Started

Sample ObsID used: 144 (HRC-I, G21.5-0.9)

File types needed: evt2; dtf1; asol1

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.

Create An Image

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

```
unix% dmcop "hrcf00144N003_evt2.fits[bin x=32,y=32]" 144_img32.fits
```

This creates an image that is 1024 x 1024 (given that a full resolution HRC-I image is 32767 x 32767); 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.

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:

```
unix% punlearn asphist
unix% pset asphist infile=pcadf084154631N003_asol1.fits
unix% pset asphist outfile=asphist.fits
unix% pset asphist evtfile="hrcf00144N003_evt2.fits[chip_id=0]"
unix% pset asphist dtffile=hrcf00144_000N003_dtf1.fits
unix% asphist
Aspect Solution List Files (pcadf084154631N003_asol1.fits[@hrcf00144N003_evt2.fits[chip_id=0]]):
Aspect Histogram Output File (asphist.fits):
Event List Files (hrcf00144N003_evt2.fits):
Live Time Correction List Files for HRC (hrcf00144_000N003_dtf1.fits):
# asphist (CIAO 3.4): WARNING: skipping 40 livetime correction records (from time: 84154850.638844 to time
```

In some cases there will be more than one `asol1.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.

Plotting the 2-dimensional POS_OFFSET vector – composed of the X_OFFSET and Y_OFFSET columns – shows the dither pattern used during the observation .

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). See [An Introduction to Exposure Maps](#) (PS, 12pp) and the [Calculating Spectral Weights](#) thread for more details on spectral weighting. 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).

Note that it is not necessary for the instrument map to be congruent with the exposure map. We set the pixelgrid parameter to create a 1024 x 1024 pixel image that covers the entire detector area.

```
unix% punlearn mkinstmap
unix% pset mkinstmap obsfile="asphist.fits[asphist]"
unix% pset mkinstmap outfile=instmap_1.0kev.fits
unix% pset mkinstmap detsubsys=HRC-I
unix% pset mkinstmap pixelgrid="1:16384:#1024,1:16384:#1024"
unix% pset mkinstmap spectrumfile=NONE
unix% pset mkinstmap monoenergy=1.0
unix% mkinstmap
Output File Name (instmap_1.0kev.fits):
Energy Spectrum File (see docs) (NONE):
Energy for mono-chromatic map [keV] (0.1:10) (1):
Pixel grid specification x0:x1:#nx,y0:y1:#ny (1:16384:#1024,1:16384:#1024):
Name of fits file + extension with obs info (asphist.fits[asphist]):
Detector Name (HRC-I):
Grating for zeroth order ARF (NONE|LETG|HETG) (NONE):
NONE, or name of ACIS window mask file (NONE):
```

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

3. Calculate the Exposure Map

Now we use mkexppmap and the aspect information stored in the histogram to project the instrument map onto the sky. We set the xygrid parameter to produce a 1024 x 1024 pixel output map; this corresponds to a bin size of 32 for the sky axes, which was used when creating an image from the event list. The get_sky_limits script (part of the [CIAO Scripts distribution](#)) can be used to easily calculate this information from the existing image:

```
unix% get_sky_limits 144_img32.fits verbose="1"
Checking binning of image: 144_img32.fits
Image has 1024 x 1024 pixels
Lower left (0.5,0.5) corner is x,y= 0.5, 0.5
Upper right (1024.5,1024.5) corner is x,y= 32768.5, 32768.5
DM filter is:
x=0.5:32768.5:#1024,y=0.5:32768.5:#1024
mkexppmap xygrid value is:
0.5:32768.5:#1024,0.5:32768.5:#1024
```

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


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

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

HRC-I Exposure Map – CIAO 3.4

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 `mkexppmap` to output percentage-completed information). Using the full aspect solution will help accurately account for chip edges, bad pixels, etc.

```
unix% punlearn mkexppmap
unix% pset mkexppmap instmapfile=instmap_1.0kev.fits
unix% pset mkexppmap outfile=expmap_1.0kev.fits
unix% pset mkexppmap xygrid="0.5:32768.5:#1024,0.5:32768.5:#1024"
unix% pset mkexppmap asphistfile=asphist.fits
unix% pset mkexppmap normalize=no
unix% pset mkexppmap useavgaspect=no
unix% mkexppmap
Aspect Histogram File (asphist.fits):
Output File Name (expmap_1.0kev.fits):
Name of Instrument Map (instmap_1.0kev.fits):
grid specification syntax x0:x1:#nx,x0:x1:ny (0.5:32768.5:#1024,0.5:32768.5:#1024):
Use Average Aspect Pointing (no):
Exposure map limits: 0.000000e+00, 6.967126e+06
Writing exposure map to expmap_1.0kev.fits
```

The exposure map can be [displayed in ds9](#) .

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

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 mkexppmap](#) for more details on this.


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=144_img32.fits
unix% pset dmimgcalc infile2=expmap_1.0kev.fits
unix% pset dmimgcalc outfile=144_img32_norm.fits
unix% pset dmimgcalc operation=div
unix% dmimgcalc
Input file #1 (144_img32.fits):
Input file #2 (expmap_1.0kev.fits):
output file (144_img32_norm.fits):
arithmetic operation (add|sub|mul|div|tst) (div):
warning: CONTENT has 1 different values.
```

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.

HRC-I Exposure Map – CIAO 3.4

The units of `144_img32_norm.fits`  are $[\text{photon}/\text{cm}^2/\text{s}/\text{pixel}]$.

When creating a normalized image with HRC data, the processing results in a pattern of rings on the final image. Viewing the image with ds9, using color map i8 makes these rings  most evident. They are the result of an HRC instrument characteristic and do not affect the observational data.

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

Alternatively, one may use dmimgthresh for a cleaner final product. The strongly variable exposure near the edge of a dithered field may produce "hot" pixels when divided into an image. While technically proper, these hot pixels can be an eyesore, drawing attention to a noisy, uninteresting portion of the image. The `dmimgthresh` tool may be used to make a "threshold cut" before dividing the image by the exposure map, thus removing the hot pixels:

```
unix% punlearn dmimgthresh
unix% pset dmimgthresh infile=144_img32.fits
unix% pset dmimgthresh outfile=144_img32_clean.fits
unix% pset dmimgthresh expfile=expmap_1.0kev.fits
unix% pset dmimgthresh cut=1.5%
unix% pset dmimgthresh value=0.0
unix% dmimgthresh
Input dataset/block specification (144_img32.fits):
Output dataset/block specification (144_img32_clean.fits):
```

Here we set our threshold at 1.5% of the maximum value of the exposure map. All image pixels with values of exposure *less than* this value will be set to 0.0 in the output file. The next step would be to divide `144_img32_clean.fits` by the exposure map for a final, fluxed image. You may want to adjust these values for your own observation.

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

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(16312.5,16488.5,376)
```

the flux can be calculated with either dmstat:

```
unix% dmstat infile="144_img32_norm.fits[sky=region(flux.reg)]" centroid=no
144_img32_norm.fits
  min:      2.713362619e-07          @:      ( 16496.5 16208.5 )
  max:      0.00017011030286        @:      ( 16336.5 16496.5 )
  mean:     5.6320753944e-06
  sigma:    1.1985221304e-05
```

HRC-I Exposure Map – CIAO 3.4

sum:	0.0024161603442
good:	429
null:	147

or `dmextract`:

```

unix% dmextract infile="144_img32_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	0.00241616034421	0.04915445396107

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`

```

#-----
#
#       Parameter file for the ASPECT HISTOGRAM Tool
#
#-----
infile = pcadf084154631N003_asoll.fits Aspect Solution List Files
outfile = asphist.fits      Aspect Histogram Output File
evtfile = hrcf00144N003_evt2.fits[chip_id=0]  Event List Files
dtffile = hrcf00144_000N003_dtf1.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/mkinstmap.par`

```

outfile = instmap_1.0kev.fits Output File Name
#-----
# Energy Band Info
#-----
# Currently, this file is a simple ascii file with two columns
spectrumfile = NONE      Energy Spectrum File (see docs)
monoenergy = 1           Energy for mono-chromatic map [keV]
#
pixelgrid = 1:16384:#1024,1:16384:#1024 Pixel grid specification x0:x1:#nx,y0:y1:#ny
obsfile = asphist.fits[asphist] Name of fits file + extension with obs info

```


HRC-I Exposure Map – CIAO 3.4

```
detsubsys = HRC-I           Detector Name
  grating = NONE           Grating for zeroth order ARF
  maskfile = NONE         NONE, or name of ACIS window mask file
  (mirror = HRMA)        Mirror Name
#
  (pbkfile = NONE)       NONE, or the name of the parameter block file
  (dafile = NONE)       NONE, CALDB, or name of ACIS dead-area calibration file
#
(ardlibparfile = ardlib.par) name of ardlib parameter file
  (geompar = geom)      Parameter file for Pixlib Geometry files
#
  (verbose = 0)         Verbosity
  (clobber = no)        Overwrite existing files?
  (mode = ql)           Enter mode for parameter file.
```

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

```
  asphistfile = asphist.fits   Aspect Histogram File
  outfile = expmap_1.0kev.fits Output File Name
  instmapfile = instmap_1.0kev.fits Name of Instrument Map
#
  xygrid = 0.5:32768.5:#1024,0.5:32768.5:#1024 grid specification syntax x0:x1:#nx,x0:x1:ny
  useavgaspect = no           Use Average Aspect Pointing
#-----
# Aspect Histogram Parameters
# If UseAvgAspect is set to yes, then only the average pointing derived from
# the many pointings in the aspect histogram will be used.
#-----
#asphistfile,f,a,"../data/aciss_asphist.fits",,, "Aspect Histogram File"
  (normalize = no)            Normalize exposure map by exposure time
#
  (geompar = geom)          Parameter file for Pixlib Geometry files
  (verbose = 0)             Verbosity
  (clobber = no)            Overwrite existing files?
  (mode = ql)               Enter mode for parameter file.
```

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

```
# parameter file for dmimgcalc
  infile = 144_img32.fits      Input file #1
  infile2 = expmap_1.0kev.fits Input file #2
  outfile = 144_img32_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)
```

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

HRC-I Exposure Map – CIAO 3.4

```
infile = 144_img32.fits      Input dataset/block specification
outfile = 144_img32_clean.fits Output dataset/block specification
(expfile = expmap_1.0kev.fits) Exposure map file
  (cut = 1.5%)              Threshold value
  (value = 0)               Replacement value
(verbose = 0)               Debug Level(0-5)
(clobber = no)             Clobber existing file
  (mode = ql)
```

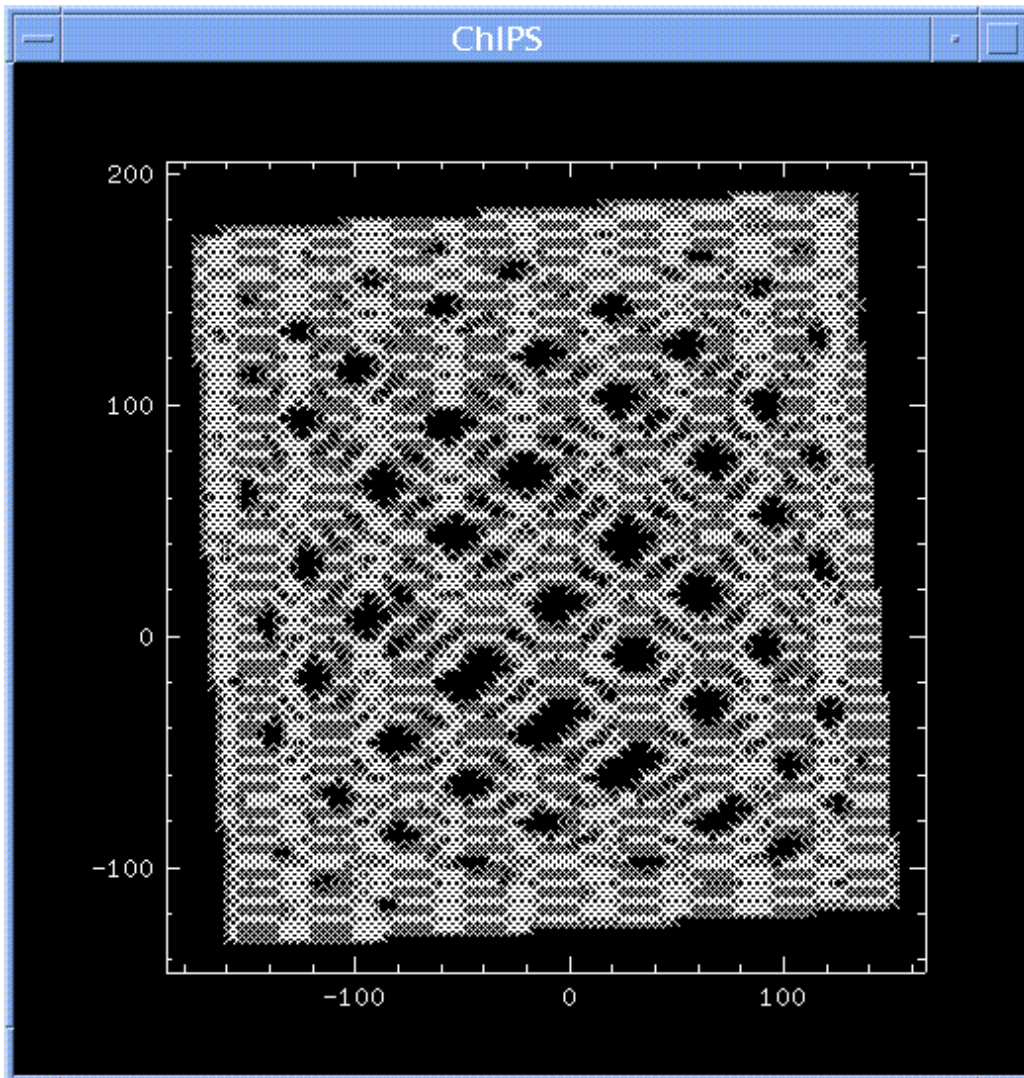
History

- 23 Dec 2004 reviewed for CIAO 3.2: no changes
 - 23 Jun 2005 CIAO 3.2.2 patch: change to `asphist` parameter file
 - 19 Dec 2005 updated for CIAO 3.3: default value of `dmextract` error and `bkgerror` parameters is "gaussian"; updated syntax for `asphist` (GTI filter is associated with the event file rather than the aspect solution); corresponding changes to screen output
 - 24 May 2006 changed "det" abbreviation to full parameter name ("detsubsys") in `mkinstmap` call
 - 01 Dec 2006 updated for CIAO 3.4: CIAO version in warning; parameter file update for `mkinstmap`
-

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

Last modified: 1 Dec 2006

Image 1: Histogram of the observation dither



The abscissa shows the X_OFFSET and the ordinate the Y_OFFSET columns of the aspect histogram.

Image 2: Exposure map for HRC-I

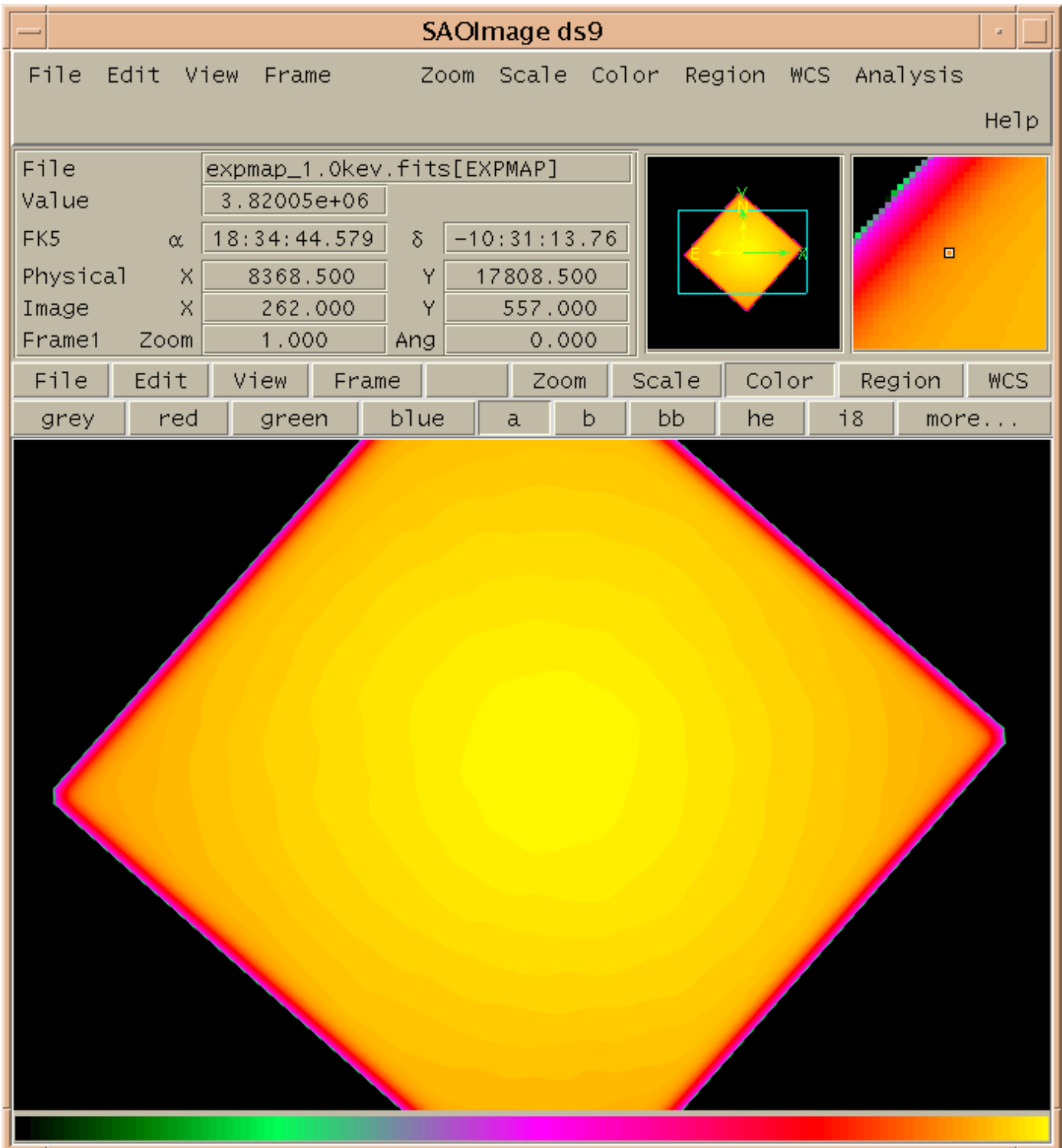


Image 3: "Rings" in the fluxed image

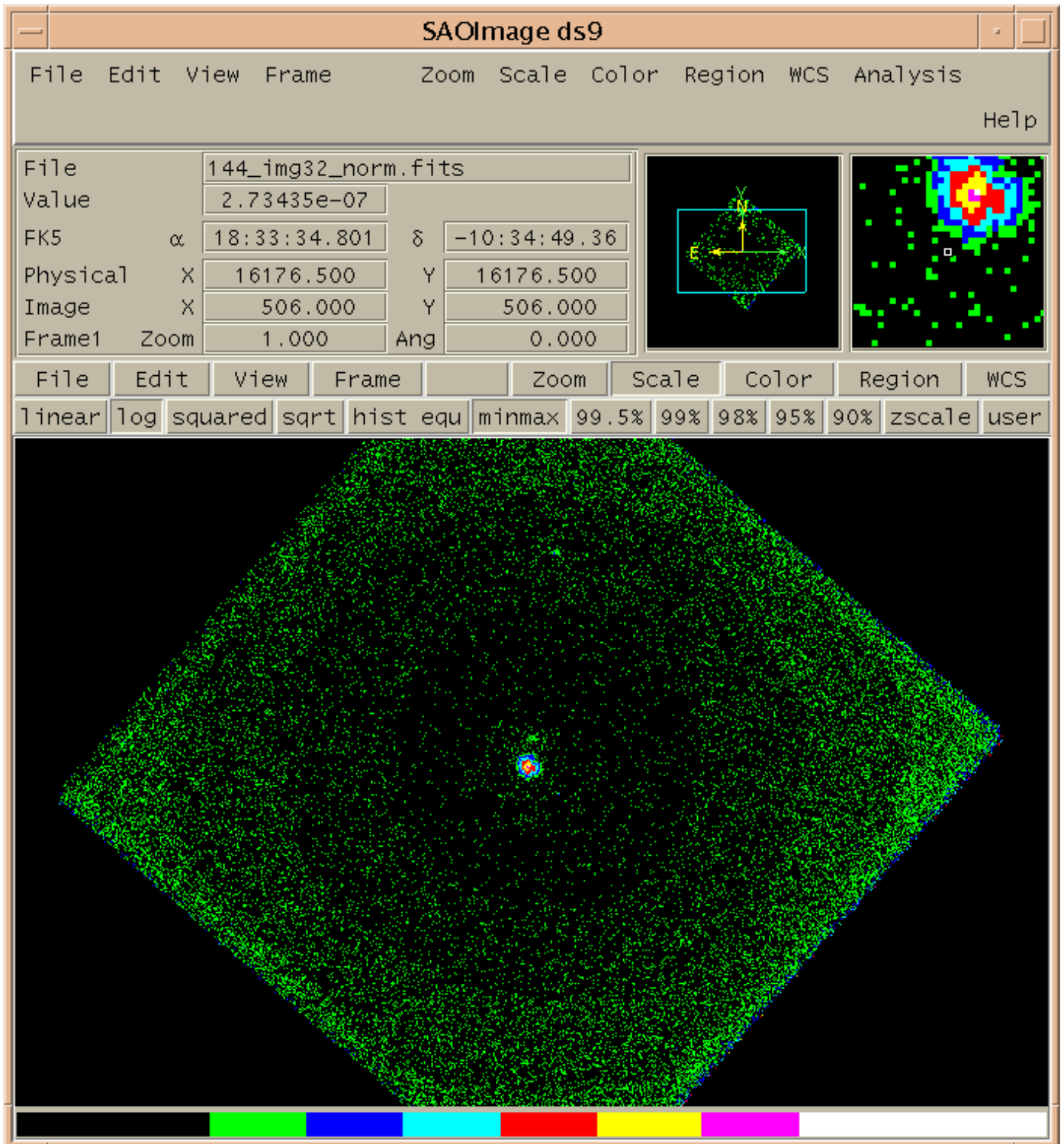


Image 3: "Rings" in the fluxed image

Image 4: Fluxed image for HRC-I

