Obtain Grating Spectra from HETG/ACIS−S Data

CIAO 3.4 Science Threads

Table of Contents

- ***Data Preparation***
- ***Get Started***
- ***Generate A New Level=1.5 Event File***
  1. Get position of zero−order image (tgdetect)
  2. Get region mask (tg_create_mask)
  3. Run tg_resolve_events
- ***Generate A New Level=2 Event File***
  1. Apply grade/status filters (dmcopy)
  2. Apply GTI filters (dmcopy)
- ***Run destreak***
- ***Extract a Grating Spectrum (tgextract)***
- ***Summary***
- ***Parameter files:***
  - tgdetect
  - tg_create_mask
  - tg_resolve_events
  - destreak
  - tgextract
- ***History***
- ***Images***
  - Image with region file overlaid
  - Original event file (left) and the destreaked version (right)

Obtain Grating Spectra from HETG/ACIS−S Data

CIAO 3.4 Science Threads

Overview

Last Update: 18 Dec 2006 − updated for CIAO 3.4: new calibration files in CALDB 3.3.0; change to wording of `tgdetect`/`dmcopy` warning

Synopsis:

Generate a new PHA2 spectrum file for any HETG/ACIS−S grating observation. The option to destreak the data is included; the `destreak` tool became part of standard data processing at software version DS 7.3.0.

In CIAO 3.1, the default `rand_pix_size` in `tg_resolve_events` was changed to zero, meaning that randomization is not applied to the coordinate computations. For users whose science requires the improved spectral resolution for resolving line blends or detection of weak features, we recommend reprocessing older data without randomization. (Fluxes of relatively isolated lines are not affected.)

Purpose:

To ensure that consistent calibration is used throughout the analysis.

Read this thread if:

you are working with an ACIS/HETG dataset affected by the Calibration Updates. Even if your observation was processed with the newest calibration, you may still wish to complete the last two steps of this thread to destreak the level=2 event file obtained from the pipeline and extract a cleaner spectrum.

Calibration Updates:

Get Started shows how to check the CALDB version used in processing your data.

- ***CALDB v3.3.0** (18 Dec 2006):* New OSIP files were added to the CALDB. The new OSIPs include the corrected gain function for ACIS−S1, which has been in the CALDB for some time. They also accurately specify the FRACEXPO column in the output gARF. The How CIAO 3.4 and CALDB 3.3.0 Affect Your Analysis section of the CIAO release notes explains how the files will affect your analysis.
- ***CALDB v2.26** (2 Feb 2004):* A new OSIP file was added to avoid significant clipping of the negative grating orders, which was present in the previous versions. The new file affects ***−110 C data only***; this FAQ shows how to find the focal plane temperature of an observation. Running this thread will apply the new calibration file to the data.
- ***CALDB v2.8** (13 Sep 2001):* New OSIP files make it possible to analyze −120 degrees S3 grating data with the gainmaps and FEFs that were in the prevous CALDB release. If you are working with an observation taken under these conditions, run the Apply an ACIS Gain Map thread before running this thread.
- ***CALDB v1.8.1** (7 Dec 2000):* New (corrected) order−sorting tables were added to the processing CALDB.

Related Links:

- Analysis Guide for Chandra High Resolution Spectroscopy: an in−depth discussion of grating analysis.
- Continuous Clocking Mode why topic: additional information for users working with CC−mode data.
- Why topic: Desteaking ACIS Data

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

Data Preparation

This analysis thread starts with the level 1 event file. Before beginning, users may wish to reprocess the data to create a new event file with the most recent calibration applied. Instructions on how to reprocess your data are available in the ACIS Data Preparation Analysis Guide.

If your data was processed with a software version *lower than* DS 7.4.0, it is recommended that you remove the acis_detect_afterglow correction and reprocess the data with a new bad pixel file before running this thread.

Get Started shows how to check the software version used in processing your data.

Get Started

Sample ObsID used: 459 (HETG/ACIS−S, 3C 273)

File types needed: evt1; flt1; asol1

In this thread, we assume that all relevant files are in the same working directory.

Check the CALDBVER keyword in the header:

```
unix% dmkeypar acisf00459_000N002_evt1.fits CALDBVER echo+
 1.6
```

Since this data was processed with a CALDBVER lower than 1.8.1, we need to complete this thread in its entirety.

To see which processing version was used on the data:

```
unix% dmkeypar acisf00459_000N002_evt1.fits ASCDSVER echo+
R4CU5UPD11.1
```

This observation was also processed before `destreak` became part of SDP, i.e. the `ASCDSVER` is lower than DS 7.3.0. Note that the version naming convention changed after version R4CU5UPD14 to the "DS" system, starting with DS 6.0.0.

If your data was processed with at least CALDB v1.8.1 but you wish to destreak the level=2 event file obtained from the pipeline, skip to the Run destreak step. Otherwise you do not need to run this thread at all and can go directly to the Create Grating RMFs for ACIS−S Observations thread.

Generate A New Level=1.5 Event File

1. Get position of zero−order image (tgdetect)

To find the zero−order location, the tool `tgdetect` is run:

```
unix% punlearn tgdetect
unix% pset tgdetect infile=acisf00459_000N002_evt1.fits
unix% pset tgdetect outfile=acis_459_src1a.fits
unix% tgdetect
Input L1 event file (acisf00459_000N002_evt1.fits):
Input source position(s) file from previous OBI or NONE (NONE):
Output source position(s) file name (root_src1a.fits) (acis_459_src1a.fits):

# DMCOPY (CIAO 3.4): Bad data type in filter string formatting
```

The warning may be ignored; it is due to a minor bug in the Data Model and does not affect the output of `tgdetect`.

The contents of the parameter file may be checked using plist tgdetect.

The source list may be viewed over the event file using ds9:

```
unix% ds9 acisf00459_000N002_evt1.fits &
```

Overlay the source list: Region−> Load Regions−> acis_459_src1a.fits[SRCLIST].

The following special cases require manual intervention in order to create a new PHA file. ***If these problems affect your data, it will be obvious when the source list is displayed on the event file.***

- If the zero order of the source is outside of the default search area (e.g. ***far from the aimpoint***), `tgdetect` will not find it. In this case, run the Correcting a Misplaced Zero−order Source Position thread to identify the correct source position.
- If the zero−order source is ***piled***, there is the potential for the centroid to be incorrect due to the "hole" created in the data. Additionally, the zero−order region may have been ***blocked*** via on−board software to avoid telemetry problems when observing bright sources. For either of these cases, refer to the Source Position for Grating Data with a Piled or Blocked Zero Order thread for instructions on how to proceed.

2. Get region mask (tg_create_mask)

The location of the HEG and MEG "arms" needs to be found next, via the tool `tg_create_mask`. This tool creates a region file, `acis_459_evt1_L1a.fits`, that will be used later to mask the image:

```
unix% punlearn tg_create_mask
unix% pset tg_create_mask infile=acisf00459_000N002_evt1.fits
unix% pset tg_create_mask outfile=acis_459_evt1_L1a.fits
unix% pset tg_create_mask input_pos_tab=acis_459_src1a.fits
unix% tg_create_mask
Input event file or stack (acisf00459_000N002_evt1.fits):
Output region file or stack (acis_459_evt1_L1a.fits):
Input table with zero order positions or stack (acis_459_src1a.fits):
Observed grating type (header_value|HETG|HEG|MEG|LETG) (HETG|HEG|MEG|LETG|header_value|HEADER_VA
```

The contents of the parameter file may be checked using plist tg_create_mask.

3. Run tg_resolve_events

The tool `tg_resolve_events` is now used to assign grating events to spectral orders, using the detector energy resolution for order separation:

```
unix% punlearn tg_resolve_events
unix% pset tg_resolve_events infile=acisf00459_000N002_evt1.fits
unix% pset tg_resolve_events outfile=acis_459_evt1a.fits
unix% pset tg_resolve_events regionfile=acis_459_evt1_L1a.fits
unix% pset tg_resolve_events acaofffile="@pcad_asol1.lis"
unix% tg_resolve_events
Input event file or stack (acisf00459_000N002_evt1.fits):
Input region file or stack (acis_459_evt1_L1a.fits):
Output event file or stack (acis_459_evt1a.fits):
Input aspect offset file (@pcad_asol1.lis):
```

It is important to note several things here:

- In some cases there will be more than one asol1.fits file for an observation. ***All*** the files must be input to the `acaofffile` parameter ***in chronological order*** (the time is in the filename, so "`ls`" lists them in order), either as a comma−separated list or as a stack (see ahelp stack for more information). For example, here we used:

  ```
  unix% more pcad_asol1.lis
  pcadf063874624N002_asol1.fits
  pcadf063875522N002_asol1.fits
  pcadf063902942N002_asol1.fits
  ```
- The tool may issue a warning of this form:

  ```
  # tg_resolve_events (CIAO 3.4): The following error occurred 10000 times:
          dsTREUNKNOWNINCOLERR -- WARNING: Not loading data from unrecognized level 1.5 input colu
  ```

  This warning may be ignored, and is explained in this FAQ.

The contents of the parameter file may be checked using plist tg_resolve_events.

The created region file, which has been appended to the event file as a block, may be viewed over the event file using ds9:

```
unix% ds9 acis_459_evt1a.fits &
```

Overlay the region file that was created by `tg_create_mask` (Region−> Load Regions−> acis_459_evt1a.fits[REGION]) and you should see something like Figure 1.

Generate A New Level=2 Event File

1. Apply grade/status filters (dmcopy)

Filter for bad grades (using ASCA grades) and for bit status:

```
unix% punlearn dmcopy
unix% dmcopy "acis_459_evt1a.fits[EVENTS][grade=0,2,3,4,6,status=0]" \
      acis_459_flt1_evt1a.fits opt=all
```

2. Apply GTI filters (dmcopy)

The Good Time Intervals (GTIs) supplied by the pipeline now need to be applied. We simultaneously eliminate an unnecessary column from the output:

```
unix% punlearn dmcopy
unix% dmcopy \
      "acis_459_flt1_evt1a.fits[EVENTS][@acisf00459_000N002_flt1.fits][cols −phas]" \
      acis_459_evt2.fits opt=all
```

Be sure to include the ***@ symbol*** in the filter expression; the command will not be executed properly if it is omitted.

Run destreak

There is a flaw in the serial readout of the ACIS chips, causing a significant amount of charge to be randomly deposited along pixel rows as they are read out. Although not much coincidence is expected for low−rate/low−exposure sources, ACIS−S4 (ccd_id=8) is significantly affected by this problem. The `destreak` tool detects coincidence of events in adjacent pixels along a row, flags probable streak events, and removes them. For details on how the tool works, see the Destreaking ACIS Data why topic.

The streak events are not removed by standard grade, status, or bad pixel filtering, and are therefore still present in the level=2 event file. Figure 2 📷 compares the original level=2 event file (left) and the destreaked version (right).

The `ccd_id` parameter should always be set to 8 (the default) to ensure that the destreak algorithm will be applied to the ACIS−S4 chip only. The algorithm should not be applied to chips that do not show streaks like those on ACIS−S4; see ahelp destreak for more information.

```
unix% punlearn destreak
unix% pset destreak infile=acis_459_evt2.fits
unix% pset destreak outfile=acis_459_dstrk_evt2.fits
unix% destreak
Input dataset/block specification (acis_459_evt2.fits):
Output dataset/block specification (acis_459_dstrk_evt2.fits):
```

The contents of the parameter file may be checked using plist destreak.

Extract a Grating Spectrum (tgextract)

The CIAO tool `tgextract` produces a PHA2 spectrum file from the level=2 event file:

```
unix% punlearn tgextract
unix% pset tgextract infile=acis_459_dstrk_evt2.fits
unix% pset tgextract outfile=acis_459_pha2.fits
unix% tgextract
Input event file (output event file from L1.5 processing) (acis_459_dstrk_evt2.fits):
If typeII, enter full output file name or '.'; if typeI, enter output rootname (acis_459_pha2.fi
Input ancillary response file name (none):
Input redistribution file name (none):
Source ID's to process: 'all', comma list, @file (all):
Grating parts to process: HETG, HEG, MEG, LETG, header_value (HETG|HEG|MEG|LETG|header_value) (h
Grating diffraction orders to process: 'default', comma list, range list, @file (default):
Ouput file type: typeI (single spectrum) or typeII (multiple spectra) (pha_typeI|pha_typeII) (ph
```

The contents of the parameter file may be checked using plist tgextract.

Summary

This thread is complete; the PHA2 grating spectrum file is named acis_459_pha2.fits. You should now proceed to the Create Grating RMFs for ACIS−S Observations thread.

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


##
## TGDETECT -- Create filter; run celldetect; narrow down detected
##             'zero order' source list; set source id's; match
##             sources to previous OBI source list.
##
## Note: if either "infile" or "OBI_srclist_file" are @lists, only
## the first item on the list is read in;  this tool only works on
## one set of input files;  if more than one file is listed,
## everything but the first are ignored.
##
        infile = acisf00459_000N002_evt1.fits   Input L1 event file
OBI_srclist_file = NONE             Input source position(s) file from previous OBI or NONE
       outfile = acis_459_src1a.fits  Output source position(s) file name
#
#   output file naming
#
     (temproot = )                Path and root file name to be given to temporary files
     (keeptemp = no)              Keep temporary files?
     (keepexit = no)              Keep exit status file?
#
#
#
     (zo_pos_x = default)         Center GZO filter sky X position (default=pixel(ra_nom))
     (zo_pos_y = default)         Center GZO filter sky Y position (default=pixel(dec_nom))
 (zo_sz_filt_x = default)         Size of GZO filter in X pixels (ACIS=400; HRC=1800)
 (zo_sz_filt_y = default)         Size of GZO filter in Y pixels (ACIS=400; HRC=1800)
   (snr_thresh = 40)              SNR threshold to select the detected sources
#
#   celldetect parameters
#
       (expstk = none)            list of exposure map files
       (thresh = )celldetect.thresh -> 3) celldetect source threshold
     (ellsigma = 3.0)             Size of output source ellipses (in sigmas)
     (expratio = 0)               cutoff ratio for source cell exposure variation
    (findpeaks = yes)             find local peaks for celldetect
(celldetect_log = )celldetect.log -> no) make a celldetect log file?
     (psftable = )celldetect.psftable -> /soft/ciao/data/psfsize20010416.fits) table of PSF size data,
    (fixedcell = 15)              celldetect fixed cell size to use
(fixedcell_cc_mode = 15)              celldetect fixed cell size to use for CC mode ACIS data
      (bkgfile = none)            background file, for celldetect
     (bkgvalue = )celldetect.bkgvalue -> 0) background count/pixel, for celldetect
  (bkgerrvalue = )celldetect.bkgerrvalue -> 0) background error, for celldetect
        (eband = )celldetect.eband -> 1.4967) energy band, for celldetect
      (eenergy = )celldetect.eenergy -> 0.8) encircled energy of PSF, for celldetect
      (snrfile = none)            celldetect snr output file (for convolution only)
     (convolve = )celldetect.convolve -> no) use convolutions for celldetect
```

```
      (xoffset = INDEF)           celldetect offset of x axis from optical axis
      (yoffset = INDEF)           celldetect offset of y axis from optical axis
     (cellfile = none)            output cell size image file
     (centroid = yes)             compute source centroids in celldetection?
#
#   tgidselectsrc parameters
#
(snr_ratio_limit = )tgidselectsrc.snr_ratio_limit -> 1) Value of SNR ratio to use as lower limit
     (setsrcid = )tgidselectsrc.setsrcid -> yes) Set src ids in output file?
#
#   tgmatchsrc parameters
#
(max_separation = )tgmatchsrc.max_separation -> 3) Maximum allowed separation (arcsec) for source
#
#
      (clobber = no)              OK to overwrite existing output file(s)?
      (verbose = 0)               Verbosity level (0 = no display)
         (mode = ql)
```

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


##
## TG_CREATE_MASK -- Calculates the mask regions of the grating arms
##    for AXAF flight L1 grating data files.  The output is a region
##    file(s) in sky coordinates.
##
        infile = acisf00459_000N002_evt1.fits   Input event file or stack
       outfile = acis_459_evt1_L1a.fits Output region file or stack
 input_pos_tab = acis_459_src1a.fits  Input table with zero order positions or stack
   grating_obs = header_value     Observed grating type (header_value|HETG|HEG|MEG|LETG)
     sA_zero_x = 1                Source A - x position of zero order
     sA_zero_y = 1                Source A - y position of zero order
     sB_zero_x = 1                Source B - x position of zero order
     sB_zero_y = 1                Source B - y position of zero order
     sC_zero_x = 1                Source C - x position of zero order
     sC_zero_y = 1                Source C - y position of zero order
     sD_zero_x = 1                Source D - x position of zero order
     sD_zero_y = 1                Source D - y position of zero order
     sE_zero_x = 1                Source E - x position of zero order
     sE_zero_y = 1                Source E - y position of zero order
     sF_zero_x = 1                Source F - x position of zero order
     sF_zero_y = 1                Source F - y position of zero order
     sG_zero_x = 1                Source G - x position of zero order
     sG_zero_y = 1                Source G - y position of zero order
     sH_zero_x = 1                Source H - x position of zero order
     sH_zero_y = 1                Source H - y position of zero order
     sI_zero_x = 1                Source I - x position of zero order
     sI_zero_y = 1                Source I - y position of zero order
     sJ_zero_x = 1                Source J - x position of zero order
     sJ_zero_y = 1                Source J - y position of zero order
(input_psf_tab = CALDB)           Calibration file with mirror psf vs off-axis angle
     (detector = header_value)    Detector type: ACIS | HRC-I | HRC-S | header_value
(radius_factor_zero = 50)              A scale factor which multiplies the app. calculation of th
(width_factor_hetg = 35)             A scale factor which multiplies the one-sigma width of the
(width_factor_letg = 40)             A scale factor which multiplies the one-sigma width of the
(r_astig_max_hetg = 0.5600000000000001) Max grating r coord (deg, along the dispersion) for HETG
(r_astig_max_letg = 1.1)             Max grating r coord (deg, along the dispersion) for LETG ast
(r_mask_max_hetg = 0.992)           Max grating r coord (deg) for HETG mask (to support offset po
(r_mask_max_letg = 2.1)             Max grating r coordinate (deg) for LETG mask (to support offs
# --------------------------------------------------------------------------
# The parameters below are to be set ONLY if the user wants to use their
# own grating mask sizes instead of having the masks automatically generated.
# Only ONE input file, with up to 10 soures, can be processed using the user
# params.  @ lists of multiple files can only be done with automated mask
# processing, or by running each file individually with hand set mask sizes.
# To start, you MUST set the following parameters:
#
# > pset tg_create_mask use_user_pars=yes last_source_toread=[letter A -> J]
#
# The parameter last_source_toread should be set to the last source letter
# for which you will enter parameters.  If you want to input 2 sources
# (regardless of their source id's), the last_source_toread=B. Sections
# A -> J are for (upto) 10 user specified sources.  In each sections,
# each source must have an ID, a zero order center position specified,
# as well as the grating mask width(s).  An example with 2 HETG sources,
# with src_id's 6 and 3:
#
# > pset tg_create_mask use_user_pars=yes last_source_toread=B
# > pset tg_create_mask sA_id=6 sA_zero_x=4762.34 sA_zero_y=2344.29
# > pset tg_create_mask sA_zero_rad=35 sA_width_heg=25 sA_width_meg=28
# > pset tg_create_mask sB_id=3 sB_zero_x=4063.54 sB_zero_y=6346.62
# > pset tg_create_mask sB_zero_rad=45 sB_width_heg=50 sB_width_meg=75
#             (units are all in sky pixels)
#
# NOTE: for Continuous Clocking data (CC mode), the HETG mask does not
# require the s#_width_heg, since the meg mask will encompase the entire
# data set.  HEG event processing in CC mode is done using the next
# tool tg_resolve_events.
# -------------------------------------------------------------------------
(use_user_pars = no)              Use the user defined mask parameters below: yes or no?
(last_source_toread = A)               Last source name to be read; character A->J.
# -------------------------------------------------------------------------
#                       Source A parameters
# -------------------------------------------------------------------------
        (sA_id = 1)               Source A - source id number
  (sA_zero_rad = )                Source A - radius of zero order mask
 (sA_width_heg = )                Source A - width of heg mask in sky pixels
 (sA_width_meg = )                Source A - width of meg mask in sky pixels
 (sA_width_leg = )                Source A - width of leg mask in sky pixels
# -------------------------------------------------------------------------
#                       Source B parameters
# -------------------------------------------------------------------------
..(through Source J)..
      (geompar = geom)            Parameter file for Pixlib Geometry files
      (verbose = 0)               Verbose level: 0 - no output, 5 - max verbosity
      (clobber = no)              Clobber existing outfile?
         (mode = ql)
```

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


#------------------------------------------------------------------------
#
#   tg_resolve_events.par: Parameter file for the tg_resolve_events program
#
#------------------------------------------------------------------------
        infile = acisf00459_000N002_evt1.fits   Input event file or stack
       outfile = acis_459_evt1a.fits  Output event file or stack
    regionfile = acis_459_evt1_L1a.fits Input region file or stack
    acaofffile = @pcad_asol1.lis  Input aspect offset file
(alignmentfile = )acaofffile -> @pcad_asol1.lis) Input sim offset file
      (logfile = stdout)          Output log (NONE|<filename>|stdout)
# The osipfile contains position dependent energy limits based on
# the CCD resolution, used for order-sorting.
# A value of "NONE" means that the file will not be used, and
# that the parameters, osort_hi and osort_lo will be used.
     (osipfile = CALDB)           Lookup table for order resolving (for acis data only)
#sort_hi, osort_lo specify fractional deviations from the integer
#order which will be included in order-sorting via CCD ENERGY values (PHA).
#eg. osort_lo-0.3, osort_hi=0.2 means that photons with real-valued
#orders between 0.7 < order <= 1.2 will be included in first order,
#1.7 < order <= 2.2 will be second order, etc.
     (osort_lo = 0.3)             Order-sorting lower bound fraction; order > m - osort_lo
     (osort_hi = 0.3)             Order-sorting high bound fraction; order <= m + osort_hi
  (grating_obs = header_value)    Observed grating type (header_value|HETG|HEG|MEG|LETG)
     (detector = header_value)    Detector type: ACIS | HRC-I | HRC-S | header_value
(energy_lo_adj = 1.0)             Lower Energy limit factor
(energy_hi_adj = 1.0)             Upper Energy limit factor
  (time_offset = 0)               Offset to add to event time to synch w/ alignment data
    (rand_seed = 1)               Random seed (for pixlib), 0 = use time dependent seed
(rand_pix_size = 0.0)             pixel randomization width (-size..+size), 0.0 = no randomizatio
     (eventdef = )stdlev1_ACIS ->{d:time,i:expno,f:rd,s:chip,s:tdet,f:det,f:sky,s:ccd_id,l:pha,s:
s:grade,s:fltgrade,s:node_id,s:tg_m,f:tg_lam,f:tg_mlam,s:tg_srcid,s:tg_part,s:tg_smap,x:status})
Output format definition
      (stdlev1 = )eventdef ->{d:time,i:expno,f:rd,s:chip,s:tdet,f:det,f:sky,s:ccd_id,l:pha,s:pi,f
s:grade,s:fltgrade,s:node_id,s:tg_m,f:tg_lam,f:tg_mlam,s:tg_srcid,s:tg_part,s:tg_smap,x:status})
 (stdlev1_ACIS = {d:time,i:expno,f:rd,s:chip,s:tdet,f:det,f:sky,s:ccd_id,l:pha,s:pi,f:energy,s:gr
s:fltgrade,s:node_id,s:tg_m,f:tg_lam,f:tg_mlam,s:tg_srcid,s:tg_part,s:tg_smap,x:status})
ACIS event format definition string
  (stdlev1_HRC = {d:time,f:rd,s:chip,l:tdet,f:det,f:sky,s:chip_id,s:pha,s:pi,s:tg_m,f:tg_lam,f:tg
s:tg_srcid,s:tg_part,s:tg_smap,x:status})
HRC event format definition string
# --------------------------------------------------------------------------
      (geompar = geom)            Parameter file for Pixlib Geometry files
      (verbose = 0)               Verbosity level of detail (0=none, 5=most)
      (clobber = no)              Clobber outfile if it already exists?
         (mode = ql)
```

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


        infile = acis_459_evt2.fits   Input dataset/block specification
       outfile = acis_459_dstrk_evt2.fits Output dataset/block specification
          (max = )                streak threshold  syntax:  m  OR  m:m:m:m
(max_rowloss_fraction = 5.0e-5)          Maximum fraction of avg streaks/node/frame
    (num_sigma = 1.0)             Sigma value for determining streak threshold
       (filter = yes)             Discard tagged events
         (mask = [status=0,grade=0,2:4,6]) Filter to select candidate streak events
       (ccd_id = 8)               CCD ID to filter
      (ccd_col = ccd_id)          CCD ID column name
     (node_col = node_id)         Node ID column name ('none' for single node)
      (exptime = -1)              frame time (s) (reads EXPTIME if no pos. value given)
    (countfile = )                filename for event row-count distribution
     (fracfile = )                filename for cumulative streak contam function
     (timefile = )                filename for exposure time lost per row
      (verbose = 0)               Debug Level(0-5)
      (clobber = no)              Clobber existing file
         (mode = ql)
```

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


##
## TGEXTRACT -- create 1D spectrum(a) table file(s) from the
##              L1.5 output event list
```

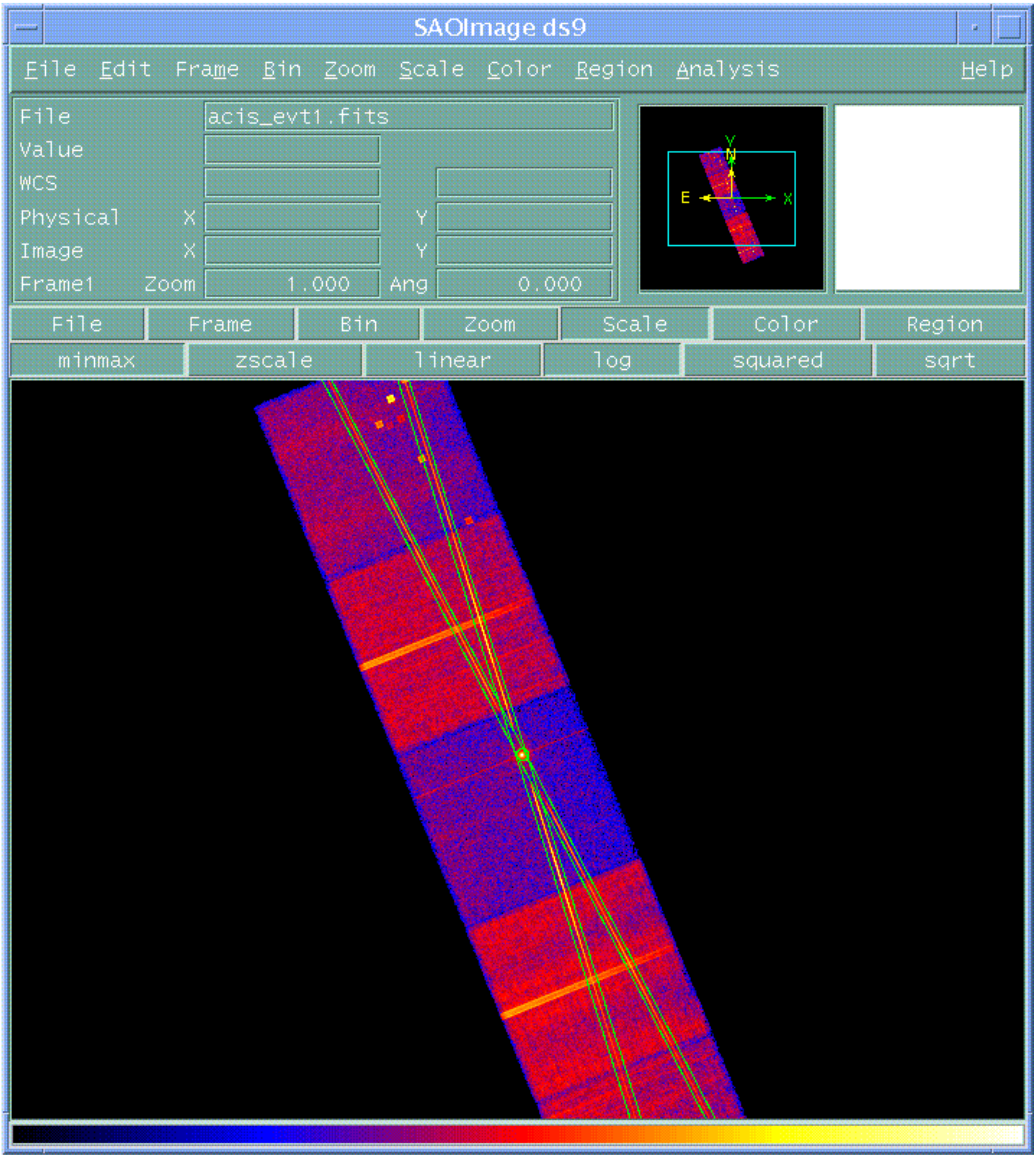
```
##
        infile = acis_459_dstrk_evt2.fits Input event file (output event file from L1.5 processing)
       outfile = acis_459_pha2.fits   If typeII, enter full output file name or '.'; if typeI, enter ou
#
# tg_srcid_list parameter explanation...
#  − "all" will process all the sources id's found in the event list
#  − a comma list is a comma separated string list of all the
#    sources to process, ie:
#       "1,2,5,7"
#  − @file is a pointer to an ascii file which contains a comma
#    separated list of the id's to process
#
 tg_srcid_list = all              Source ID's to process: 'all', comma list, @file
  tg_part_list = header_value     Grating parts to process: HETG, HEG, MEG, LETG, header_value
#
# tg_order_list parameter explanation...
#  − "default" is set to process the following:
#       if ACIS:  1, 2, 3, −1, −2, −3
#       if HRC:   −1, 1
#  − a comma list is a comma separated string list of the orders
#    the user wants to process, ie:
#       "−5, −1, 1, 3"
#  − a range list sets the min and max of the orders to process;
#    all the orders in between, will be processed, ie:
#       "−1..5"  will do orders from −1 to +5th order
#    a range list can be mixed with comma separated list
#  − @file is a pointer to an ascii file which contains a comma
#    separated list and/or range list of the orders to process
#
 tg_order_list = default          Grating diffraction orders to process: 'default', comma list, range li
      ancrfile = none             Input ancillary response file name
      respfile = none             Input redistribution file name
  outfile_type = pha_typeII       Ouput file type: typeI (single spectrum) or typeII (multiple spectra)
(inregion_file = none)            Input region file.
     (backfile = none)            Input background file name
        (rowid = )                If rowid column is to be filled in, enter name here
    (bin_units = angstrom)        Bin units (for bin parameters below): angstrom, eV, keV
  (min_bin_leg = compute)         Minimum dispersion coordinate for LEG, or 'compute'
  (max_bin_leg = compute)         Maximum dispersion coordinate for LEG, or 'compute'
 (bin_size_leg = compute)         Bin size for binning LEG spectra, or 'compute'
 (num_bins_leg = compute)         Number of bins for the output LEG spectra, 'compute'
  (min_bin_meg = compute)         Minimum dispersion coordinate for MEG, or 'compute'
  (max_bin_meg = compute)         Maximum dispersion coordinate for MEG, or 'compute'
 (bin_size_meg = compute)         Bin size for binning MEG spectra, or 'compute'
 (num_bins_meg = compute)         Number of bins for the output MEG spectra, or 'compute'
  (min_bin_heg = compute)         Minimum dispersion coordinate for HEG, or 'compute'
  (max_bin_heg = compute)         Maximum dispersion coordinate for HEG, or 'compute'
 (bin_size_heg = compute)         Bin size for binning HEG spectra, or 'compute'
 (num_bins_heg = compute)         Number of bins for the output HEG spectra, 'compute'
     (min_tg_d = default)         Minimum tg_d range to include in histogram, or use 'default'
     (max_tg_d = default)         Maximum tg_d range to include in histogram, or use 'default'
(extract_background = yes)             Extract the local background spectrum?
(min_upbkg_tg_d = default)         Minimum value of tg_d for the background up spectrum.
(max_upbkg_tg_d = default)         Maximum value of tg_d for the background up spectrum.
(min_downbkg_tg_d = default)         Minimum value of tg_d for the background down spectrum.
(max_downbkg_tg_d = default)         Maximum value of tg_d for the background down spectrum.
      (geompar = geom)            Parameter file for Pixlib Geometry files
      (clobber = no)              OK to overwrite existing output file(s)?
      (verbose = 0)               Verbosity level (0 = no display)
         (mode = ql)
```

History

16 Dec 2004	updated for CIAO 3.2: minor changes to parameter files
14 Jan 2005	created the Data Preparation section
05 Dec 2005	updated for CIAO 3.3: output filenames include ObsID; parameter file changes (`kernel` parameter removed from all "tg" tools; several changes to the `destreak` parameter file, including a new default value for the `mask` parameter)
05 Jan 2006	information on reprocessing included in Data Preparation section
04 Apr 2006	expanded information on special cases in the Get position of zero−order image (tgdetect) section
18 Dec 2006	updated for CIAO 3.4: new calibration files in CALDB 3.3.0; change to wording of `tgdetect`/`dmcopy` warning

URL: http://cxc.harvard.edu/ciao/threads/spectra_hetgacis/ Last modified: 18 Dec 2006

Image 1: Image with region file overlaid



SAOImage ds9
File Edit Frame Bin Zoom Scale Color Region Analysis Help
File
Value
WCS
Physical X Y
Image X Y
Frame1 Zoom Ang
Y
N
E
X
File Frame Bin Zoom Scale Color Region
minmax zscale linear log squared sqrt

File: acis_evt1.fits

Zoom: 1.000

Ang: 0.000

Image 2: Original event file (left) and the destreaked version (right)

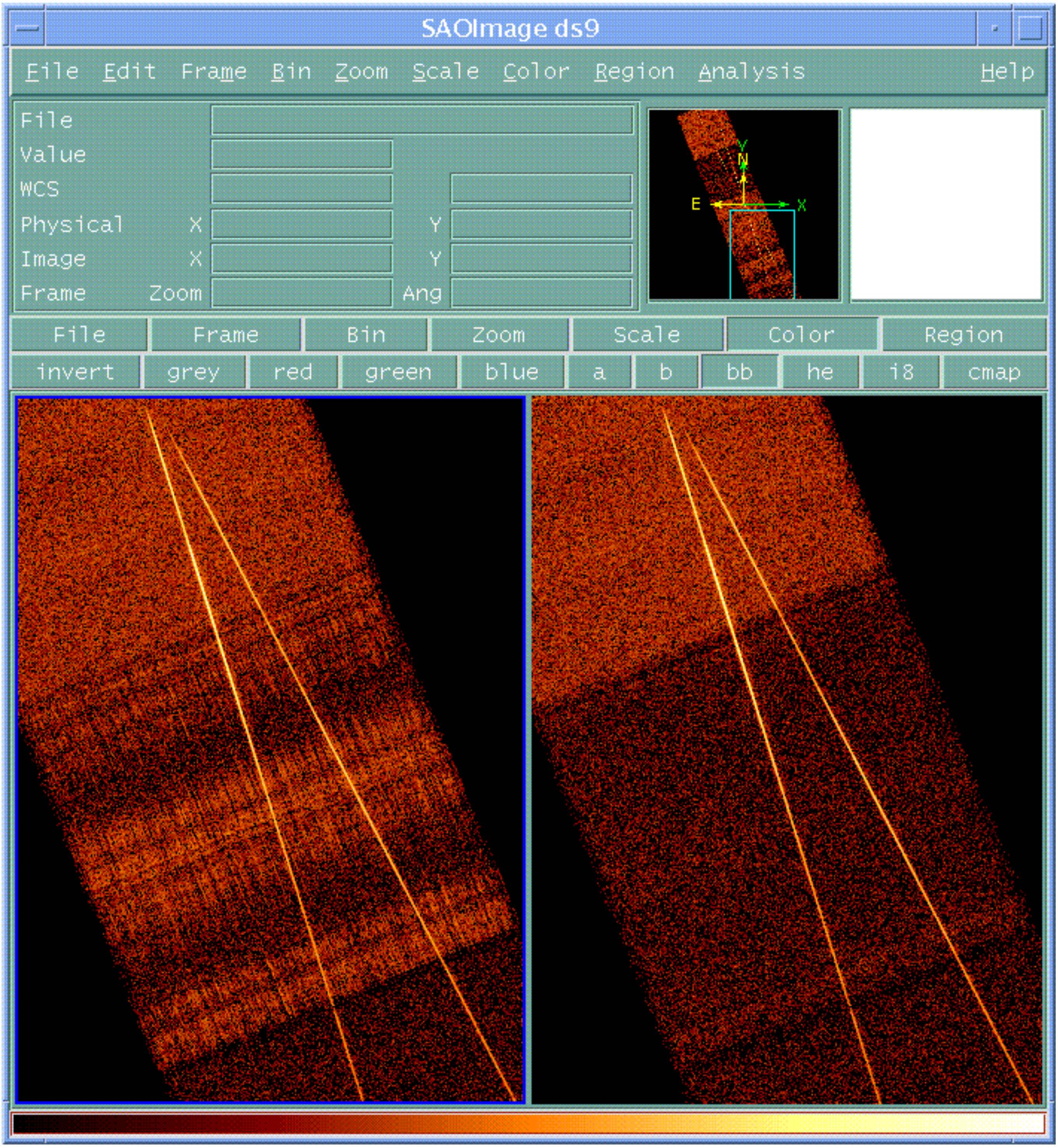

SAOImage ds9
File Edit Frame Bin Zoom Scale Color Region Analysis Help
File
Value
WCS
Physical X Y
Image X Y
Frame Zoom Ang
Y
N
E
X
File Frame Bin Zoom Scale Color Region
invert grey red green blue a b bb he i8 cmap

Image 2: Original event file (left) and the destreaked version (right)