

*AHELP for CIAO 3.4*

destreak

Context: [tools](#)*Jump to:* [Description](#) [Examples](#) [Parameters](#) [CHANGES IN CIAO 3.3](#) [Bugs](#) [See Also](#)

Synopsis

Remove streak events from ACIS data

Syntax

```
destreak infile outfile [max] [max_rowloss_fraction] [num_signal]
[filter] [mask] [ccd_id] [ccd_col] [node_col] [exptime] [countfile]
[fracfile] [timefile] [verbose] [clobber] [mode]
```

Description

Background

Images of the ACIS–S4 chip show a variable pattern of linear streaks which fall along rows of pixels with constant CHIPY. It appears that these streaks are caused by a flaw in the serial readout which randomly deposits significant amounts of charge along the pixel row as charge is read out. This excess charge appears as a string of spurious X–ray events which are not removed by standard grade filtering. Since the streak locations vary between readouts, the streak events cannot be removed using standard methods appropriate for bad columns and pixels.

Instead, streak events can be identified – and hence removed – from the fact that they occur within a single frame and deposit events along a single CHIPY row of a single node. For low count–rate observations, it is relatively rare for multiple events to mimic this behavior. A more detailed description of the theory can be obtained from The CIAO web site contains [a document which describes the theory](#) in more detail. See also the thread "[Destreak the ACIS–S4 Chip](#)".

Algorithm

To run destreak, the user must, at a minimum, enter the names of the input and output event files. The input file will typically be an ACIS level 2 FITS event file. Although destreak can filter a level 1 event file, it is important to set the mask option appropriately when doing so. The output event file will be a copy of the input file modified to flag or (if filter=yes) remove the streak events. destreak is designed to remove, as accurately as possible, streak events from affected ACIS chips (only S4 in current practice), with minimal impact on actual source events. However, in questionable cases (e.g. observations which have bright sources on ACIS–S4), the careful observer would be well advised to examine the spatial distribution of identified streak events to verify that the streak filter has not misidentified a significant number of valid source events as streaks.

Ahelp: destreak – CIAO 3.4

The user is given a number of options. CCDs can be treated as single node devices or four–node devices. destreak will filter streaks from each node of the data on a frame by frame basis. destreak filters out streak events based on a threshold value either given by the user or determined by the program; these values may be provided on a per–node basis.

Observations will typically have a low event count in each row of a node during a single frame interval. If a chipy row rarely records more than two events per row in a single frame, then three or more events might be considered a streak. In this case the threshold value could be 2 and all rows with a count higher than 2 would be flagged or discarded as streak events.

The user may specify the threshold for either 1 node, 4 nodes or none at all. The one and four node thresholds will be applied to all filtered chips. If no max value is given by the user, destreak will determine the optimal threshold for each node of each chip using a histogram of the observed events.

Streak events may be filtered from the data or propagated to the output and flagged with status bit 15. This behavior is determined by the filter parameter.

The mask parameter may be used to have destreak remove certain events from consideration as candidate streak events. In general, events already flagged as "bad" for other reasons should not be considered as candidate streak events. For example, the default value

```
mask=" [status=0,grade=0,2:4,6] "
```

ensures that destreak will consider only events with good grades and zero status as potential streak events. This ensures that the rate of candidate streak events is as low as possible, minimizing the rate of spurious event coincidences. Such a mask setting is particularly important when the input file is a level 1 event file. Alternatively, the user may avoid filtering certain spatial regions of an observation by setting the mask parameter to specify a spatial region which will be ignored by destreak. For example, the user may wish to avoid losing photons associated with a bright source, and provide a mask region that includes the source. The mask string uses the data–model syntax for filtering regions (see "ahelp dmsyntax", "ahelp dmfiltering", and/or "ahelp dmregions" for details), and can be the name of a region file (e.g. mask=region(filename)).

Normally users apply destreak with `ccd_id = 8` and default values for the threshold (`max`) parameter, but the program is capable of fine control for expert users. destreak can filter a single chip, some chips, or all of the chips used in an observation. The chip identifier can be specified in the `ccd_id` parameter. Multiple chips use the format 'a:b:c:d:e', where a,b,c,d, and e are integers between 0 and 9, and any number of chips (up to 6) may be specified. Any chips that are not in the event list will simply be ignored by the destreak filtering routine. If no chip is specified, destreak will filter all of the chips used in the observation. Additionally, If no DETNAM keyword is present in the infile header, destreak will attempt to filter all 10 ACIS chips.

The user may specify the names of the chip and node columns used in the input file, allowing destreak to handle files that are not standard CXC event files. If invalid column names are specified, destreak will default to `ccd_col="ccd_id"` and `node_col="node_id"` or `"ccdnode"`. destreak will produce an error if these columns are not found. If `node_col` has the special value "none" then the chip will be treated as a single node device.

Three statistics files can, optionally, be generated as additional output. The first of the three files, defined by the `countfile` parameter, holds the observed and predicted data for the number of counts per row and the number of occurrences. The countfile also holds the equation and parameters that were used to fit the observed event data.

The second file, defined by the `fracfile` parameter, contains the cumulative event contamination based on the observed and predicted counts.

The third file, defined by the `timefile` parameter, holds the amount of lost observation time due to streak events for each row of each node. The user should note that generating statistics files can significantly

increase program execution time.

The user also has the ability to define the length of time that will constitute a frame for the lost-time statistics file. The `exptime` parameter will default to the value of the `EXPTIME` keyword if no positive value is given (in particular, the default value of `-1` tells `destreak` to use the header keyword).

Notes

- `Destreak` should always be used with caution because it removes counts from any source sufficiently bright to generate, within a single frame-time, multiple events within a single row of one CCD node. Bright point sources are certain to be affected.
- No `EXPTIME` keyword is present in the header of Chandra level 2 event files of ACIS-HETG or ACIS-LETG observations processed prior to March 2003. So, if the user wants to generate the lost-time statistical file, a value for `exptime` must be manually supplied in these cases. For these observations, the value of the `EXPTIME` keyword can be found in the header of the associated level-1 event files.

Example 1

```
destreak infile="destreak3.fits" outfile="filtered3.fits" max=""
filter=yes mask="" ccd_id=8 ccd_col="ccd_id" node_col="node_id"
exptime=-1 countfile="" fracfile="" timefile="" verbose=1 clobber=yes
```

This example illustrates simple use of `destreak`. Identify and remove streak events from chip 8 in `destreak3.fits` event data and output the cleaned event data to `filtered3.fits` (which will be overwritten if it already exists.) Allow `destreak` to determine the optimal threshold value for streak events on each node. Use the `EXPTIME` keyword in the input file header for the frame time. Look for chip information in the `ccd_id` column of the input and node information in the `node_id` column of the input.

Example 2

```
destreak infile="destreak3.fits" outfile="filtered3.fits" max=""
filter=no mask="" ccd_id=8 ccd_col="ccd_id" node_col="node_id"
exptime=3.0 countfile="" fracfile="" timefile="" verbose=1 clobber=no
```

This example illustrates flagging instead of removing the events and specifying a specific frame time. Flag streak events from chip 8 in `destreak3.fits` event data and output the flagged event data to `filtered3.fits` (which is not to be overwritten if it already exists.) Allow `destreak` to determine the optimal threshold value for streak events on each node. Use a value of 3.0 seconds for the length of a frame exposure. Look for chip information in the `ccd_id` column of the input and node information in the `node_id` column of the input.

Example 3

```
destreak infile="destreak1.fits" outfile="filtered1.fits" max="1:1:2:1"
filter=yes mask="circle(3706,2998,424)" ccd_id=8 ccd_col="ccd_id"
node_col="node_id" exptime=-1 countfile="counts.fits"
fracfile="ratio.fits" timefile="losttime.fits" verbose=1 clobber=yes
```

Filter streaks from chip 8 in `destreak1.fits` event data and output the filtered event data to `filtered1.fits`. Identify four nodes on chip 8 and apply threshold values of 1, 1, 2 and 1 to nodes 0 through 3 respectively. Do not filter the region defined by the string `"circle(3706,2998,424)"`. Use the `EXPTIME` keyword value for the

length of a frame exposure. Create statistics files and overwrite any existing output with the same name. Look for chip information in the `ccd_id` column of the input and node information in the `node_id` column of the input.

Example 4

```
destreak infile="destreak2.fits" outfile="filtered2.fits" max=2
filter=yes mask="region(ds9.reg)" ccd_id="" ccd_col="ccd_id"
node_col="node_id" exptime=-1 countfile="counts.fits"
fracfile="ratio.fits" timefile="losttime.fits" verbose=1 clobber=yes
```

Filter streaks from all chips in `destreak2.fits` event data and output the filtered event data to `filtered2.fits`. Apply a threshold value of 2 to all nodes. Mask out the region defined by the `ds9.reg` file. Use the `EXPTIME` keyword value for the length of a frame exposure. Create statistics files and overwrite any existing output with the same name. Look for chip information in the `ccd_id` column of the input and node information in the `node_id` column of the input.

Parameters

| name | type | ftype | def | min | max | units | reqd |
|-----------------------------|---------|--------|--------------------------|--------|-----|-------|------|
| <u>infile</u> | file | input | | | | | yes |
| <u>outfile</u> | file | output | | | | | yes |
| <u>max</u> | string | | | | | | no |
| <u>max_rowloss_fraction</u> | real | | 5.0e-5 | 0.3e-4 | 1 | | no |
| <u>num_sigma</u> | real | | 1.0 | | | | no |
| <u>filter</u> | boolean | | yes | | | | no |
| <u>mask</u> | string | | [status=0,grade=0,2:4,6] | | | | no |
| <u>ccd_id</u> | string | | 8 | | | | no |
| <u>ccd_col</u> | string | | ccd_id | | | | no |
| <u>node_col</u> | string | | node_id | | | | no |
| <u>exptime</u> | real | | -1 | | | s | no |
| <u>countfile</u> | file | output | | | | | no |
| <u>fracfile</u> | file | output | | | | | no |
| <u>timefile</u> | file | output | | | | | no |
| <u>verbose</u> | integer | | 0 | 0 | 5 | | no |
| <u>clobber</u> | boolean | | no | | | | no |

Detailed Parameter Descriptions

Parameter=`infile` (file required filetype=input)

The name of the event file containing uncorrected (streaky) data.

Parameter=outfile (file required filetype=output)

The output file name for filtered (or flagged) event data.

Parameter=max (string not required)

The threshold value.

If the user supplies *m* as a maximum, *m* will be applied as the threshold for every node of every chip. If the user supplies *a:b:c:d* as a maximum; *a*, *b*, *c* and *d* will be applied to nodes 0, 1, 2 and 3 respectively for every chip specified by the parameter *ccd_id*. If the user does not supply a maximum, a threshold will be chosen by *destreak* for every node of every chip that is examined.

The threshold is the value that differentiates the streak events from the source and background events. For example, when using a threshold of 3, all events on a single node of a single row of a CCD that occur during a single frame will be filtered (or flagged) as streak events if the number of events exceeds the value of the parameter *max*. *destreak* is designed to determine the optimal threshold value when *max* is not defined by the user.

Parameter=max_rowloss_fraction (real not required default=5.0e-5 min=0.3e-4 max=1)

Maximum allowed rate at which CCD rows can be removed due to streaks

This parameter limits the total rate at which rows can be discarded due to streaks. The program will automatically adjust the value of the threshold parameter to enforce the specified *max_rowloss_fraction* rate. The *rowloss_fraction* is defined as the fraction of rows read out which were identified as streaks. For example, if $1.e4$ frames were read out using a 512 row subarray, a total of $512 * 1.e4 = 5.12e6$ rows were read out. If *max_rowloss_fraction*= $5.e-4$, the tool will allow a maximum of 2560 rows to be identified as streaks.

Parameter=num_sigma (real not required default=1.0)

Sigma value for determining streak threshold.

In normal usage, the value of this parameter should not be changed. For each CCD node, the *destreak* algorithm first constructs a histogram of the number of occurrences of CHIPY rows with a given event count per frame. This is called the row-count-per-frame distribution. The row-count-per-frame distribution of streak events appears to be an exponential. Because non-streak events tend to occur with only a very small row-count in a single frame, they appear as an excess in the histogram at low row-count values. To distinguish between streak events and non-streak events, one can derive an optimal row-count threshold to distinguish between the two distributions. All events from above-threshold rows are labelled as streak events. The optimal threshold value is defined as that row-count value at which the observed number of occurrences exceeds the rate predicted for streak events by a factor of *num_sigma* times the Poisson uncertainty in the predicted value: $(\text{observed} - \text{predicted}) = \text{num_sigma} * \text{sqrt}(\text{predicted})$.

Parameter=filter (boolean not required default=yes)

Filter the streak events, or simply flag them?

When *filter* is set to *yes*, streak events will be removed and will not appear in the output. When *filter* is set to *no*, streak events will be propagated to the output with STATUS bit 15 set to 1.

Parameter=mask (string not required default=[status=0,grade=0,2:4,6])

Filter to select candidate streak events

The mask parameter may be used to remove certain events from consideration as candidate streak events. In general, events already flagged as "bad" for other reasons should not be considered as candidate streak events. For example, the default value

```
mask="[status=0,grade=0,2:4,6]"
```

ensures that destreak will consider only events with good grades and zero status as potential streak events. This ensures that the rate of candidate streak events is as low as possible, minimizing the rate of spurious event coincidences. Such a mask setting particularly important when the input file is a level 1 event file. To block out a certain region from destreak filtering, the user may specify a region mask string or filename. The region mask string uses the data-model syntax for filtering regions (see "ahelp dmsyntax", "ahelp dmfiltering", and/or "ahelp dmregions" for details), and can be the name of a region file (e.g. mask="region(filename)"). The region specified is the region that will be ignored by destreak's filtering routine.

Parameter=ccd_id (string not required default=8)

The ACIS chip(s) to filter.

ACIS chips are numbered 0–9, beginning in the top left corner and reading left to right, and top to bottom. The user can enter a single chip or multiple chips to filter. For example '3:4:5:6:7' will apply filtering to 5 chips (3–7). If ccd_id is left blank, destreak will remove streaks from all available chips.

In normal use, the ccd_id parameter should always be set to 8 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.

Parameter=ccd_col (string not required default=ccd_id)

Name of the chip identifier column in the input table.

Parameter=node_col (string not required default=node_id)

Name of node identifier column in the input table.

Parameter=exptime (real not required default=-1 units=s)

The amount of time allocated to each frame for the lost-time statistics file.

An exptime value of -1 will default to the value of the EXPTIME header keyword.

Parameter=countfile (file not required filetype=output)

Filename for observed and predicted count statistics.

Specifies a file that will hold the observed and predicted data for the number of counts per row and the number of occurrences. The observed data could be viewed as a graphical histogram and the predicted data could be viewed as a fit to the data. The countfile also holds the equation and parameters that were used to fit the observed event data. This is an optional statistics file. The user should note that keeping statistics may significantly increase the program execution time.

Parameter=fracfile (file not required filetype=output)

Filename for event contamination statistics.

Specifies a file containing the cumulative event contamination based on the observed and predicted counts. This is an optional statistics file. The user should note that keeping statistics may significantly increase the program execution time.

Parameter=timefile (file not required filetype=output)

Filename for lost time statistics.

Specifies a file that holds the amount of lost observation time due to removed streak events for each row of each node. This is an optional statistics file. The user should note that keeping statistics may significantly increase the program execution time.

Parameter=verbose (integer not required default=0 min=0 max=5)

Verbosity level of terminal display information to user, (DataModel output included).

Parameter=clobber (boolean not required default=no)

Clobber existing output with the same name?

Boolean that specifies whether an existing output with the same name will be clobbered or not.

CHANGES IN CIAO 3.3

Ignoring Non-zero Status Bits

The tool no longer ignores non-zero status bit events. Instead, these events should be ignored by using the mask parameter. In general, events already flagged as "bad" for other reasons should not be considered as candidate streak events. For example, the new default value

```
mask=" [status=0,grade=0,2:4,6]"
```

ensures that destreak will consider only events with good grades and zero status as potential streak events.

When using destreak on TIMED VFaint mode data processed by `acis_process_events` with `check_vf_pha=yes`, make sure the mask option is set to allow non-zero status events to be considered as candidate streak events. Normally destreak should ignore the non-zero status events, but `check_vf_pha` flags them as potential cosmic rays and, in this case, such events should be kept to allow destreak to identify streaks most efficiently.

Parameter Changes

As noted in the previous section, the default value of the mask parameter has been changed to "[status=0,grade=0,2:4,6]". A new parameter, `num_sigma`, has been added. Also, the default value of the `max_rowloss_fraction` parameter has been changed from $2.0e-4$ to $5.0e-5$.

Bugs

See the [bugs page](#) on the CIAO website for an up-to-date listing of known bugs.

See Also

tools

[acis build badpix](#), [acis classify hotpix](#), [acis detect afterglow](#), [acis find hotpix](#),
[acis process events](#), [acis run hotpix](#), [acisreadcorr](#), [dmcoppy](#), [lightcurve](#)

The Chandra X-Ray Center (CXC) is operated for NASA by the Smithsonian
Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.
Smithsonian Institution, Copyright © 1998–2006. All rights reserved.

URL:
<http://cxc.harvard.edu/ciao3.4/destreak.html>
Last modified: December 2006