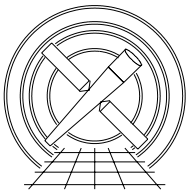
MIT Kavli Institute



Chandra X-Ray Center

Specifications: `tg_choose_method` & `tgdetect2` Algorithms and Interfaces

David Huenemoerder

September 11, 2012

Revision History

1. 2012.09.11 Version 0.2
2. 2012.09.06 Initial version 0.1

1 Introduction

The zeroth order of a *Chandra* grating observation defines the origin of the dispersed spectrum's wavelength scale. Hence, accurate determination of the zeroth order centroid is a fundamental step in spectral extraction. The ACIS detector, however, can become saturated by bright sources. The brighter the source, the greater the distortion of the zeroth order due to rejection of events in the core of the image, eventually leading to a "cratered" central region with few or no counts. To mitigate telemetry saturation for observations of extremely bright sources, ACIS itself is sometimes configured to exclude a region including the zeroth order from the transmitted data.

There are two CIAO programs for determining the zeroth order centroid, `tgdetect` (based on the `celldetect` program), and `tg_findzo`[1] (based on the ISIS `findzo` program). Each has limitations: `tgdetect` will fail to give an accurate result for distorted or blocked zeroth orders, while `tg_findzo` will fail to give an accurate result for faint sources (those without a significant frame-shift streak) or for crowded fields. Here we specify the interface and algorithm for development of a CIAO program, `tg_choose_method`, to automatically choose between the two programs.

We additionally specify a new tool, `tgdetect2`, which bundles `tgdetect`, `tg_findzo`, and `tg_choose_method` into a single program and conveniently hides all the details of each component. It is hoped that `tgdetect2` can be used in most cases, especially in pipeline processing where it will remove much of the manual intervention following V&V, since the pipeline default method is `tgdetect` which is often inappropriate, requiring off-line determination of the zeroth order centroid and submission for reprocessing.

1.1 Scope

`tg_choose_method` applies only to *Chandra* observations obtained with a grating instrument (HETG or LETG) in conjunction with the ACIS-S detector. Use with HRC-I, HRC-S, or ACIS-I is not specified.

[1] `tg_findzo` will be included in the next release of CIAO, version 4.5

tgdetect2 will only run tg_findzo and tg_choose_method if the input event file is from ACIS-S; otherwise tgdetect will be run.

2 tg_choose_method Interface & Algorithm

2.1 Overview

We require that tg_findzo has been run and produced a valid "src1a" file. This file has an HDUCLAS3 header keyword value of FINDZO, contains the zeroth order's counts (in table column NET_COUNTS), the grating arm's counts (in header keyword COUNT_TG), and the frame-shift streak's counts (in header keyword COUNT_ST). Based on empirical values from inspection of over 700 observations processed with both tgdetect and tg_findzo, criteria have been derived which are sensitive to the zeroth order quality and which provide the correct choice of method in all but a few cases. Some cases are pathological and will *always* require human intervention to review the result (e.g., very crowded fields, very bright, extended sources). Tests have shown that the correct method is chosen in all but about 2% of the cases.

The criteria are based on the count-rate-per-frame (cpf) and streak signal-to-noise ratio ($ssnr$). The rate-per-frame is used rather than counts-per-second because CCD event pile-up is sensitive to the integration time. The primary discriminant is the threshold in the grating spectrum's cpf vs. zeroth order cpf where pile-up becomes significant and the zeroth order's cpf saturates. The $ssnr$ must also be above some threshold in order for tg_findzo to be appropriate. Other tests attempt to catch common cases, such as zeroth order exclusion (a high grating rate, but a very low ratio of zeroth order to grating rate). Some graphical examples are given in Appendix A.

2.2 Interface

The interface will be the standard CIAO parameter i/o with the following parameters:

infile (string) Required input "src1a" file name, as produced by tg_findzo.

(method) (string) Output of the selected program's name, tgdetect or tg_findzo. The default value is "unknown", and an allowed value is "none" (for possible future use). (Ignored on input.)

2.3 Algorithm

Here we step through the algorith from start to finish in pseudo-code, adapted from the prototype ISIS tg_choose_method.sl script.

1. Validate the input file, which must be a src1a file written by tg_findzo, i.e., HDUCLAS3 must have the value, "FINDZO". If not, set the output parameter, "method" to "unknown" and exit with an error.

2. Define constants determined empirically from data trends. In the following, "tg" refers to rate in grating spectrum region (from the tg_findzo output file header), "zo" refers to rate in zeroth order region (also from tg_findzo output), "cpf" means counts-per-frame, and "snr" is signal-to-noise ratio.

```
snr_streak_min  =  20.0   % below this, don't use tg_findzo
snr_grating_min = 100.0   % signal-to-noise limit for grating spectrum
cpf_fzo_tg_limit_01 = 0.90  % below this => tgdetect
cpf_fzo_zo_limit_02 = 0.55  % below this => tgdetect
cpf_fzo_tg_limit_02 = 6.00  % above this => tg_findzo  % * UNUSED *
cpf_fzo_zo_limit_01 = 0.90  % above this => tg_findzo  % * UNUSED *
```

The ratio of the zeroth order's rate to the grating spectrum's rate is sensitive to craters and exclusion: below this (and above cpf_fzo_grating_limit_02) implies the need for tg_findzo:

```
cpf_ratio_zo_tg_limit = 0.06
```

Empirically determined parameter region to catch some special cases:

```
snr_streak_to_grating_min = 0.37
snr_streak_to_grating_max = 0.99
snr_grating_01 = 84.0
snr_grating_02 = 245
```

3. Read the header and data of the tg_findzo src1a file and set exposure, counts, and rate values:

```
K_rate = TIMEDEL / EXPOSURE   % rate constant, from header keywords
R_zo   = NET_COUNTS * K_rate  % zo rate, from src1a binary table column
C_tg   = COUNT_TG             % grating counts, from header keyword
R_tg   = C_tg * K_rate        % grating rate
C_st   = COUNT_ST             % streak counts, from header keyword
R_st   = C_st * K_rate        % streak rate
```

4. If the streak is too faint, use tgdetect:

```
if ( sqrt( C_st ) < snr_streak_min ) method = tgdetect
```

5. Otherwise, establish additional conditionals:

```
cond_01 = R_tg > cpf_fzo_tg_limit_01    % tg rate high enough
cond_02 = R_zo > cpf_fzo_zo_limit_02    % zo rate high enough
cond_03 = (R_zo / R_tg) <= cpf_ratio_zo_tg_limit  % zo/tg rate limit
cond_04 = sqrt( C_tg ) > snr_grating_min  % grating SNR high enough
cond_04b = sqrt( C_tg ) > snr_grating_02  % grating SNR high enough, alt
cond_05 = not  ( sqrt( C_st / C_tg ) >  snr_streak_to_grating_min
            and sqrt( C_st / C_tg ) <= snr_streak_to_grating_max )

if ( cond_01  and ( cond_02 or cond_03 )
 and ( ( cond_05 and cond_04 ) or cond_04b ) )
  method = tg_findzo
else
  method = tgdetect
```

6. Set the output parameter, method, to the selected program, tgdetect or tg_findzo.

3 tgdetect2 Interface & Algorithm

3.1 Overview

We can bundle the programs tg_findzo, tg_choose_method, and tgdetect into one program in order to provide a more convenient interface which should suffice for most cases. The individual components would remain available when finer control is required. The top-level program can be called tgdetect2 and would have the same interface as tg_findzo, since that has the smallest subset of parameters between tgdetect and tg_findzo.

3.2 Interface

The interface will be the standard CIAO parameter i/o with the following parameters, the first four of which are the same as tg_findzo and which are held in common with tgdetect. We add two parameters which specify whether the subsidiary programs should have their respective parameter files reset on execution (e.g. "punlearned").

infile (string) Input event file (required)

outfile (string) Output source table file ("src1a") (required)

(zo_pos_x) (real) Initial guess for sky-x position ($default \Rightarrow xpixel(RA_TARG, DEC_TARG)$)

(zo_pos_y) (real) Initial guess for sky-y position ($default \Rightarrow ypixel(RA_TARG, DEC_TARG)$)

(unlearn_tgdetect) (boolean) Default value is "yes" do a "punlearn" on tgdetect to set all its parameters to the defaults.

(unlearn_tg_findzo) (boolean) Default value is "yes", to do a "punlearn" on tg_findzo to set all its parameters to the defaults.

3.3 Algorithm

1. If specified, punlearn the subsidiary parameter files.
2. The coordinate parameters (zo_pos_[xy]) must be either both set to a real number, or both set to default. If not, generate an error. Values should be passed to the subsidiary programs, tgdetect or tg_findzo.
3. If the input event file refers to detectors HRC-S, HRC-I, or ACIS-I, then run tgdetect.
4. If the input event file is from ACIS-S but has a READMODE which is not TIMED, then run tgdetect.
5. If GRATING is NONE, then run tgdetect.
6. Otherwise, run tg_findzo.
7. If tg_findzo has been run, then run tg_choose_method on the src1a file which was written by tg_findzo.
8. If tg_choose_method has been run, check the output method parameter, and if it is equal to tgdetect, then run tgdetect.

A Graphical Examples

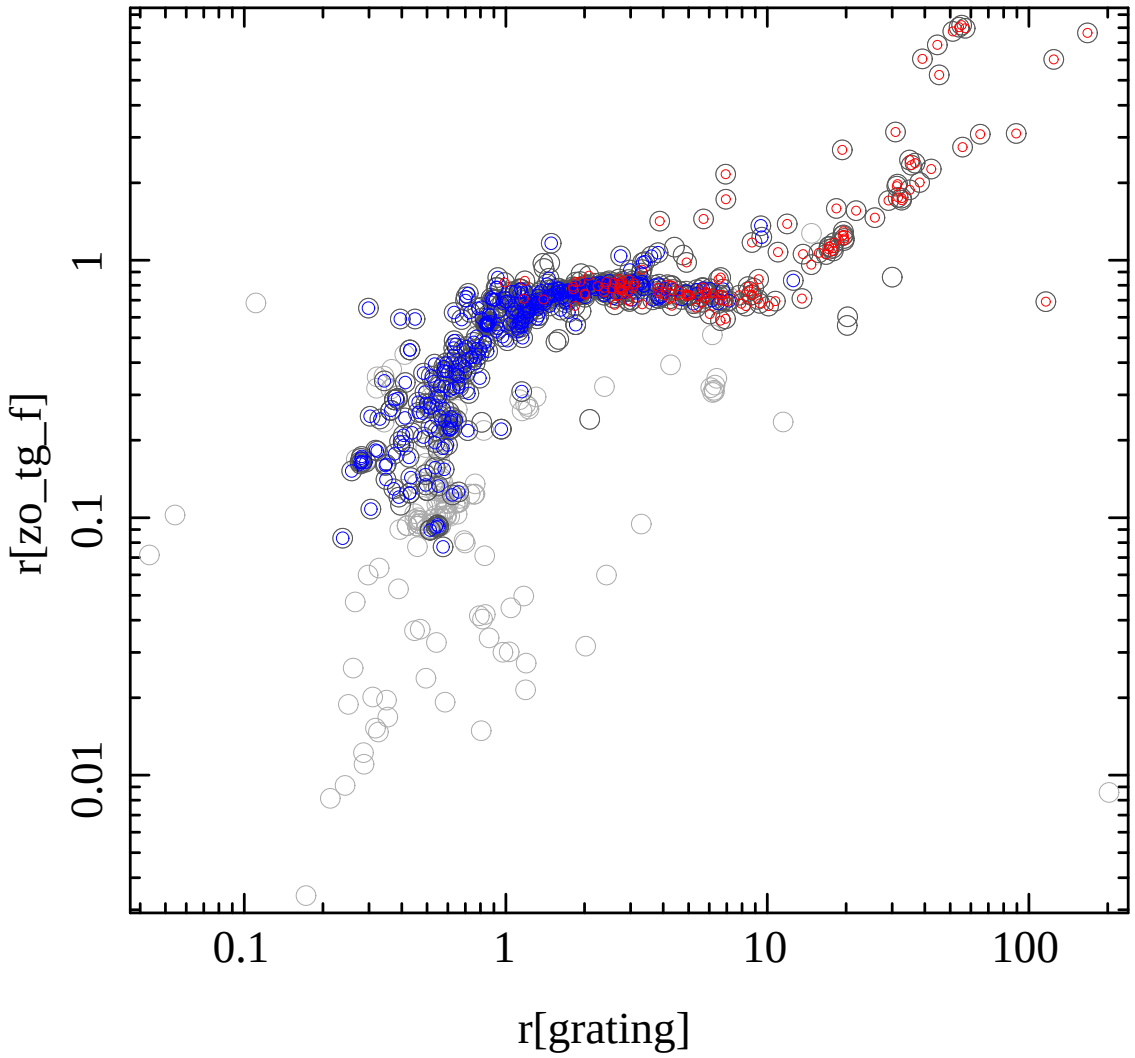


Figure 1: The grating rate (counts-per-frame; x-axis) vs. the zeroth order rate (y-axis), taken from `tg_findzo` output. The curve flattens as pile-up becomes important (near $(x, y) = (1, 1)$). Red circles used `findzo` in *TGCat*, while the blue used `tgdetect` (some extractions in *TGCat* use the method, "none", which means an a priori specified source position). There is a large range on the flat part of the curve where either method works well. The large scatter of outliers are due to the complicating factors of zeroth order exclusion or cratering, extended sources, or multiple sources, yielding inaccurate values.

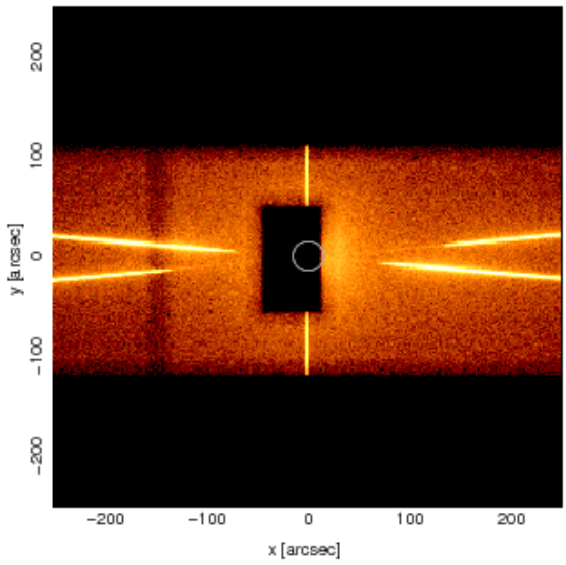


Figure 2: The zeroth order region for an observation with zeroth-order exclusion. `tgdetect` would select some bright point, `tg_findzo` yields an accurate position. The streak and grating counts are strong; the zeroth order to grating rate ratio is low. (ObsID 3808)

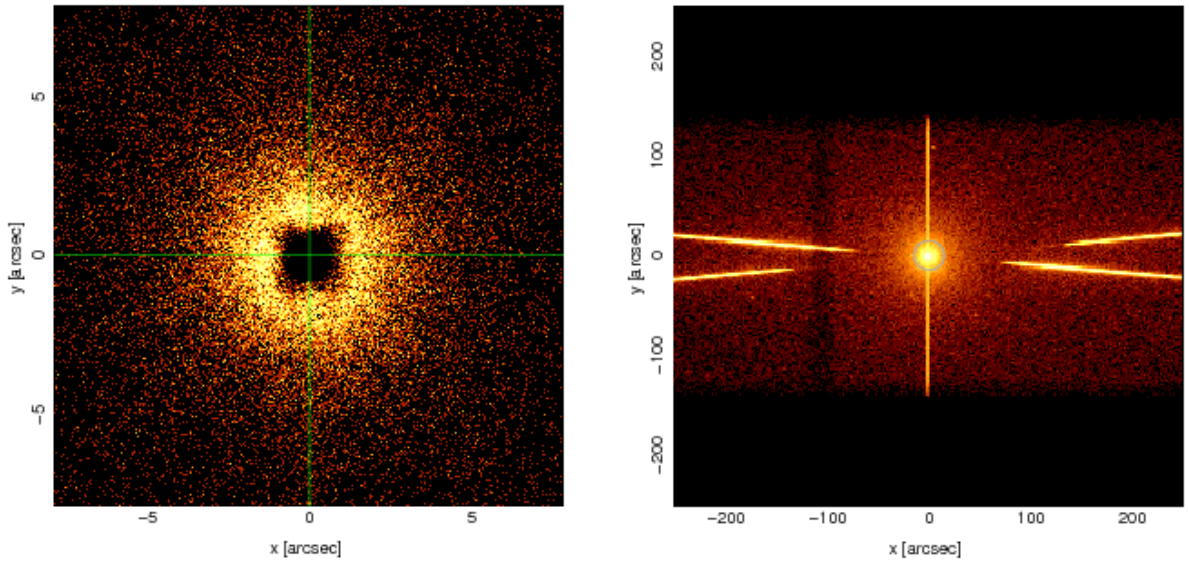


Figure 3: The zeroth order region for an observation with a very saturated zeroth order. `tgdetect` would select some bright point on the crater rim, while `tg_findzo` yeilds an accurate position. The streak, grating, and zeroth order rates are all high. (ObsID 2739)

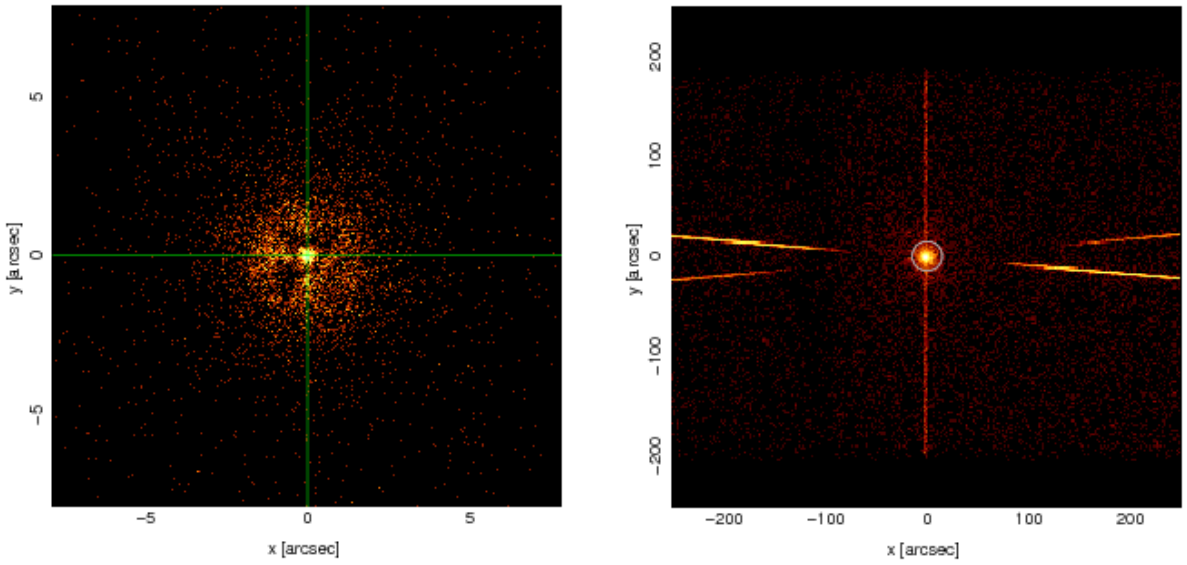


Figure 4: The zeroth order region for an observation with a moderately saturated zeroth order. While the bright point might give the correct centroid via `tgdetect`, `tg_findzo` will definitely yeild an accurate position. The bright center is due to ACIS event-repositioning; without that option, the central region's rate might be depressed and `tgdetect` would give an inaccurate position. (ObsID 3167) 2739)

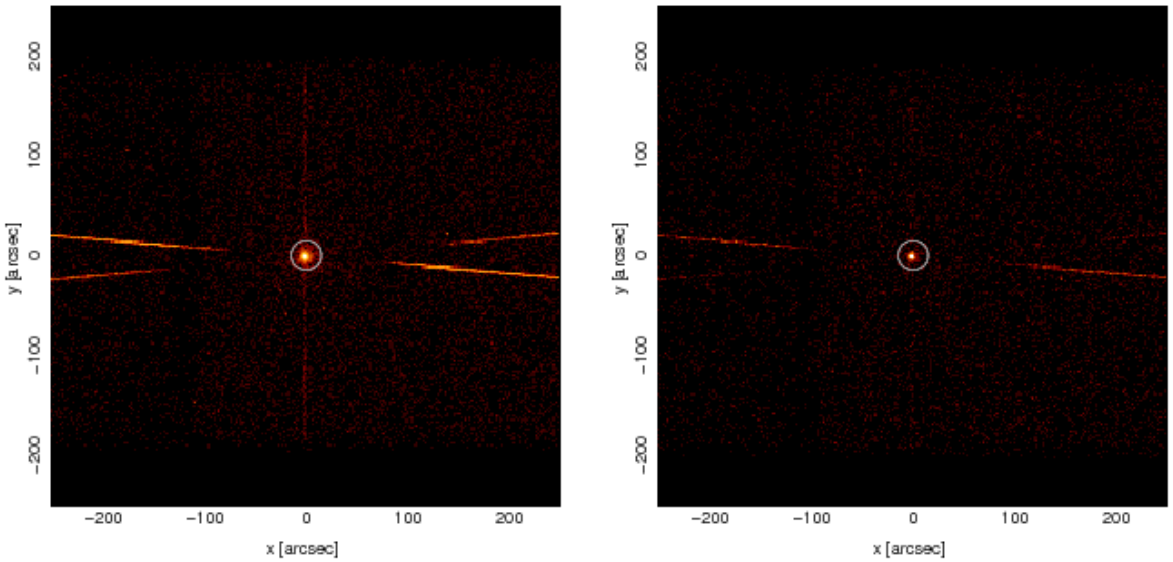


Figure 5: The zeroth order regions for observations with a moderately strong (left) and weaker (right) zeroth order. The brighter is near the threshold for `tg_findzo` use. (ObsIDs 10801, 4284)

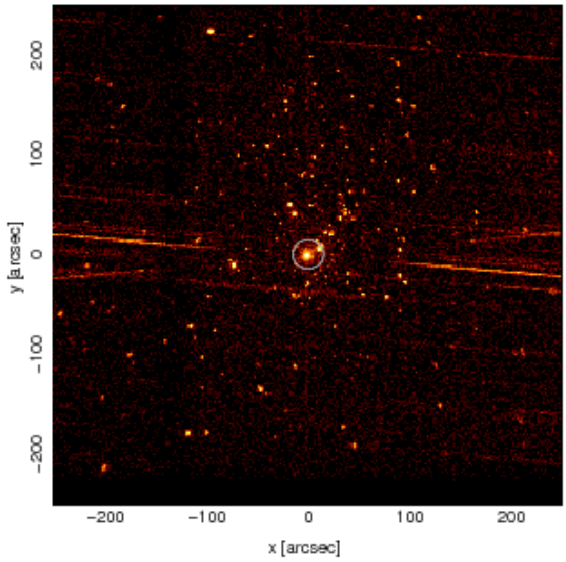


Figure 6: The zeroth order region for an observation with multiple relatively bright sources. None have a strong streak, but the multiple sources could contribute to presumed streak counts. `tg_findzo` fails, but `tgdetect` will center on *some* source. (ObsID 7409)