CHANDRA
X-ray Center 60 Garden St., Cambridge Massachusetts 02138 USA

MEMORANDUM

Date: March 4, 2008
From: Richard J. Edgar
To: ACIS Team
Subject: Generation of Trap Maps to model CTI of ACIS
Cc: CXC Calibration Group
File: `memo.tex`
Version: `1.2`

1 Abstract

This memo details CXC Calibration Group procedures for generating trap map files for the charge transfer ineficiency corrector which is a part of `acis_process_events`.

2 Introduction

Shortly after launch, the front-illuminated (FI) chips in Chandra's Advanced CCD Imaging Spectrometer (ACIS) suffered radiation damage, which dramatically increased the Charge Transfer Ineffeciency (CTI) from immeasurably small ($<\sim 10^{-7}$) to of order 10^{-4}. One consequence of this damage is energy resolution which is strongly dependent on row number, `chipy`.

The back-illuminated (BI) devices also have CTI, at a lower level. Unlike the FI devices, where CTI is confined to the imaging array, it exists in the imaging array, the frame store, and the serial output register. This complicates the process of modeling and correcting the effects of CTI for the BI devices.

The work of Townsley et al. (2000), supplemented by that of the MIT ACIS team (e.g. Grant et al 2004), has suggested a partial correction for this effect. By examining the pulse-height values of all 9 pixels in an event island in light of calibration data, it is possible to restore a portion of the lost resolution.

A description of the correction algorithm to implement this method was provided by Grant to the Chandra X-ray Center (CXC). Calibration files (known as *Trap Map Files*) for the ACIS Imaging array (I array) and the S2 chip on the spectroscopic array were also provided by Grant. The algorithm was implemented within the `ciao` program `acis_process_events`, which is used to produce Level 1 and Level 2 ACIS event lists.

The present memo details efforts by the CXC ACIS Calibration Group to learn to generate trap map files and to generate them for other epochs and chips than those provided by the ACIS PI team.

3 Creation of Trap Map Files for the FI Devices

This section will discuss in some detail the process of generating trap map files for FI CCDs at the CXC.

line	energy (eV)	strength
Mn/Fe–L	670	very weak
Al–Ka	1487	strong
Au–M	2112	very weak
Ti–Ka–esc	2771	very weak
Ti–Ka	4511	strong
Ti–Kb	4932	moderate
Mn–Ka	5894	strong
Mn–Kb	6490	moderate
Au–La	9711	weak
Au–Lb	11100	weak

Table 1: Spectral lines in the External Calibration Source used in the generation of trap map files.

3.1 Data Preparation

Trap map files are based on spectral data obtained by ACIS when observing the External Calibration Source (ECS). This ^{55}Fe source has titanium and aluminum targets, and so the strong lines in its spectrum are the K shell fluorescent lines of Al, Ti and Mn. In addition, a number of weak lines are visible, as listed in table 1.

Observations of the ECS are cataloged with annotations periodically by Bev Lamarr (MIT ACIS PI team). Her memo is posted in several places by the CXC. The unedited memo in text and postscript format is on the internal ACIS calibration site. A selection of "good" observations is also made and posted on the public ACIS Calibration site. The selection is based on usage of the standard observing modes (Timed Event mode; chips I0–I4 plus S2 and S3, or S0–S5; focal plane temperature close to $-120°$C).

ECS data are available from the Chandra archive in one of two ways. The "Provisional Retrieval Interface" (`http://asc.harvard.edu/cgi-gen/cda/retrieve5.pl`) allows one to select ECS observations, manually, one at a time, given the obsid.

Internally at the CXC there is a command-line interface to the archive via the program `arc4gl`, which is wrapped and made more user-friendly in Aldcroft's script `arc5gl`. Since this is a command line program, it is easy to script, making automation of this process possible. Accordingly, a script was generated which identifies good ECS data sets for a given epoch (quarterly periods starting at the beginning of the $-120°$C era on 31 January 2000), obtains level 1 event lists from the archive, and stores them in a standardized directory structure.

For many epochs (3-month periods), this will produce a sufficient quantity of good data. For others many if not most (or, for epoch 3, all) of the ECS runs for either the extended I array or the S array occurred when the temperature was slightly out of Bev's spec. Joe DePasquale is working on a solution to this problem. It involves creating a revised good time interval (GTI) extension for each observation, by filtering on a smoothed version of the focal plane temperature (which is found in the mtl1 files, and can be accomplished with the dmgti tool). This results in the recovery of a significant amount of useable data for some epochs. See Joe's (forthcoming, as of this writing TBR) memo for further details.

These data must be sorted by chip and quadrant. Sorting by chip is desirable since the trap map files are generated one chip at a time. Sorting by quadrant is necessary to make the size of the ultimate merged data files more managable.

Some of the data in the archive have been CTI corrected, and some have not. For our purposes here, uncorrected data are required, so it is necessary to run the program `acis_process_events` on each file (one for each obsid, chip, and quadrant) with the CTI corrector and TGAIN correction turned off. The event lists should also be filtered on good grades (ASCA grades 0, 2, 3, 4, and 6) This gives us an uncorrected level 1 event list, suitable for further processing.

One feature of `acis_process_events` needs to be worked around. When a file has zero events in it, and it is fed to `acis_process_events`, the program generates a single event with nonsense data in it. These must be filtered out during merging. An alternative strategy that proved unwieldy would be to keep track of which obsid used which chips, and select only chips for which we are likely to have many (i.e. more than zero) events.

The event lists should be filtered for bad events, i.e. with the status word nonzero. Care must be taken to allow events on node boundary columns to be considered, however. The syntax for such a filter is:

[`status` = 0000000000000000000000x0000]

which says the "bad column" bit can be either zero or one.

These files tend to be large, and so eliminating the (useless) tdet, det and sky coordinates columns can help limit the file sizes and save disk space.

The program `dmmerge` can then be used to merge the files into one file per node per chip per epoch. Further merging of four epochs gives a full year's worth of observing, which is enough to populate even the very weak lines, and to provide significant numbers ($\sim 1 - 2 \times 10^4$) of events in each column in the strong lines.

3.2 Analysis of Scatterplots

These level 1 event files (one per chip per quadrant, merged to cover a whole year's observations) are then analyzed using the IDL program `event browser` (EB), which was developed at Penn State University by Pat Broos (`http://www.astro.psu.edu/xray/docs/TARA/`).

There are two types of scatterplot display which need analysis: `PHA` vs. `chipy` and the upper pixel `up_pix` vs. `chipy`. The former shows the loss of charge due to CTI effects. The latter shows how much of the charge is recovered by the trailing pixel in the 3×3 pixel island, and so is avaliable for the CTI corrector to restore to the central pixel. Since we wish to examine chip coordinates, the X and Y coordinates should be set to `chipx` and `chipy` respectively.

Before processing the scatterplots, however, it is a good idea to look for anomalies. For example, the above-mentioned anomaly in `acis_process_events` produces events with very large negative pulse heights, chip coordinates, or other properties. These tend to skew the autoscaling in the display graphics routines of EB.

Another anomaly seen on some quadrants of some chips (TBR: which?) is low pulse height noise in certain columns. On the plot of `PHA` vs. `chipy` this appears as many events below the Al K–α line. An examination of the `PHA` vs. `chipx` plot shows that these events tend to be confined to a few columns, where they can be quite bright (brighter in terms of events per pixel than the spectral lines). TBR: example.

Absent anomalies, one can proceed to fit straight lines to the trend of `PHA` vs. `chipy` for 11 spectral lines. EB provides a linear regression with sigma-clipping facility. This permits the user to include events seen visually while excluding those from ajacent lines. Special care must be used if the CTI is appreciable, when fitting Ti and Mn K–β lines, as they tend to blend (for large `chipy`) with the K–α line.

The K–α lines are easily seen on the scatter plots. The normal contrast plot for I1c3 in epochs 18–21 is shown in Figure 1 An adjustment of the contrast brings up the K–β lines. Zooming in so that the Mn K–β line just fits on the display is advised. Zooming in even further can be helpful, especially for blended lines. Also, minimizing the pixel size helps to show the trends more clearly. The gold L lines at high energy are useful for controlling the trend, and can be seen at extreme contrast. The weak low-energy lines are most easily seen by filtering out all but ASCA grade 0 events (i.e. single pixel events), which minimizes the background. One could worry whether the CTI trend for these events differs from that of the full standard grade set, but no such trend was observed.

Slopes, intercepts, and sigmas are recorded in a text file. Using systematic names helps with the downstream data handling scripts, so I've used names like `epoch1-4_s4c0_tara.txt`, giving epochs, chip, and node in the file name itself.

When the above fits are complete, changing the y–axis of the scatterplot to `up_pix` (which is Generic4 in the EB menu) brings up the second plot for analysis. The y–axis range should be adjusted to something like -17 to 30 ADU, which will show the trend of the upper pixel increasing from near zero at low row numbers to a larger value (which is energy-dependent) at high row numbers. This is another consequence of CTI: some of the charge lost from the central pixel is recovered in the upper pixel.

We then selected events by energy from the spectral plot. This practice requires that the gain file used by `acis_process_events` to create the `ENERGY` column in the FITS file be reasonably accurate. The `up_pix` vs. `chipy` plot is then a tight scatterplot, which can be fit easily using the linear regression with sigma clipping routine of EB. Again, intercepts, slopes, and sigma values are recorded. The filenames in this case were of the form `epoch18-21_charge_loss_s4c0.txt`. In practice, we did not use the gold L lines for this process.

The text files containing slopes from these two sets of fits were then massaged into RDB tables, and then fit. First fit the slope of the `pha` vs. `chipy` plot as a function of it's intercept to a power law, using `QDP` or similar software. The slopes of this trend is the parameter α, which is the power law index in the relation between charge cloud size and energy. Typical values for α range from 0.5 to 0.7. The results (both power law index and normalization factor) are placed in a file such as `S4_alpha.rdb`.

We then fit charge loss vs. charge trailing to obtain the parameter f. The charge loss parameter is the slope of the `pha` vs. `chipy` trend, and the charge trailing parameter is the slope of the trailing pixel pulse height (`up_pix`) vs. `chipy` trend. This is fit to a straight line using `QDP` and the slope (f) and intercept placed in a file such as `trailing_fraction_s4.rdb`.

These last two files are then merged, with all others from other chips, into file `alpha_f.rdb`.

The chip ID, quadrant number, and the fitted values of α and f are put into an RDB table for future use.

3.3 Fitting to produce the Trap Map file

With values of α and f in hand, we proceeded to use Grant's IDL program `nkquad.pro`, which fits a quintic polynomial to the `PHA` vs `chipy` plot for either the Al or Mn K–α line. We selected the Al line in each case. Data for all columns for a given quadrant are used, and the coefficients of the least-squares best-fit fifth degree polynomial are saved.

These in turn are fed into the program `nkmap.pro`. This program attempts to fit a function to the `PHA` vs `chipy` plot for each column. Since there are not overwhelming numbers of events in a single column, even for a whole year's observations, it is presumed that the high-order terms of the polynomial from `nkquad.pro` are the same for all columns, and only the contant and linear terms are fit.

The program then generates an image, evaluating the best-fit function for every pixel of the device. This image, plus (in the FITS header) the values of α and f are the information in the trap map file.

The images in these files can be inspected using `sd9` or other FITS image-viewing tool. If all is well, the image will be somewhat streaky, with no black and no very bright columns. The `PHA` vs. `chipy` plot for a typical column is shown in Figure 2. There are a few categories of "bad" columns which can show up at this point:

- Columns with step functions in the PHA vs chipy plot. These seem to be manufacturing defects, or other damage produced very early in the program (prior to launch). In principle we could model these jumps and correct at least some of these columns, but as the number is fairly small (about a dozen), it has not been thought worthwhile at this time. We show two examples of this effect. There is a small step-function shown in Figure 4, which could in principle be modeled. However, there are a few columns with huge step-funtions such as that in Figure 5.
- Related to step functions (possibly a similar occurence in the Frame Store) are columns with anomalously low pulse heights that are otherwise normal. These can be modeled if the aluminum line is above the noise. An example is s0c2, column 661.
- Soft noise. Some columns have an abundance of low pulse-height noise. Sometimes this rises to the level of the (sagging) Al K–α line, and interferes with the fit. Adjusting the thresholds for the fit can help with this effect. For this purpose, there is a file of bad columns which is an input to nkmap.pro. Sometimes a second iteration is required to correct for this effect. An example of this is shown in Figure 3.
- Other bad fits, actual bad columns, etc. Roughly a dozen columns on all eight FI devices cannot be fit for other reasons, or have no data because they are otherwise flagged as "bad". These columns are typically copied from neighboring columns. Again, the number of such columns is very small; of order 0.1% of the columns on the ACIS FI devices.

One such file is generated for each chip. Glen Allen of SDS reformats a complete set of such files to make a cti file for the CALDB.

At this point Grant and others can test the trap map files. I include a few plots from Grant (http://space.mit.edu/home/cgrant/cticorr/) showing improved energy resolution and flattening of the mean PHA for strong lines as functions of chipy. They can also be tested in-house, by reformatting the trap maps to CALDB specifications (using Glenn Allen's S/Lang code), correcting event lists, and examining such properties as mean pulse height and line widths vs. chipy.

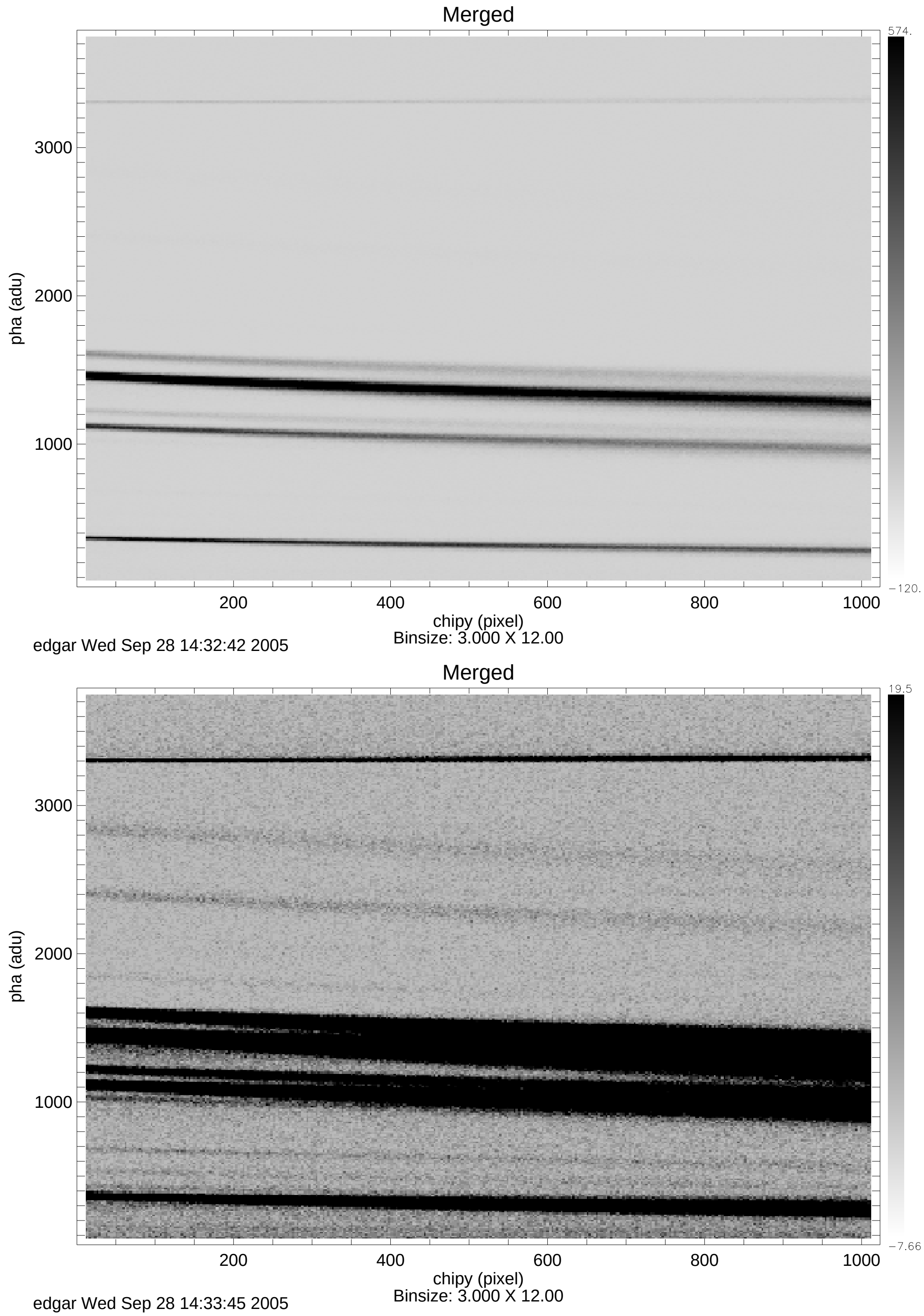

Figure 1: `PHA` vs `chipy` plot for a typical quadrant, at two contrast settings. The stong spectral lines (K–α and K–β lines) are clearly visible. Enhancing the contrast brings up the weaker lines.

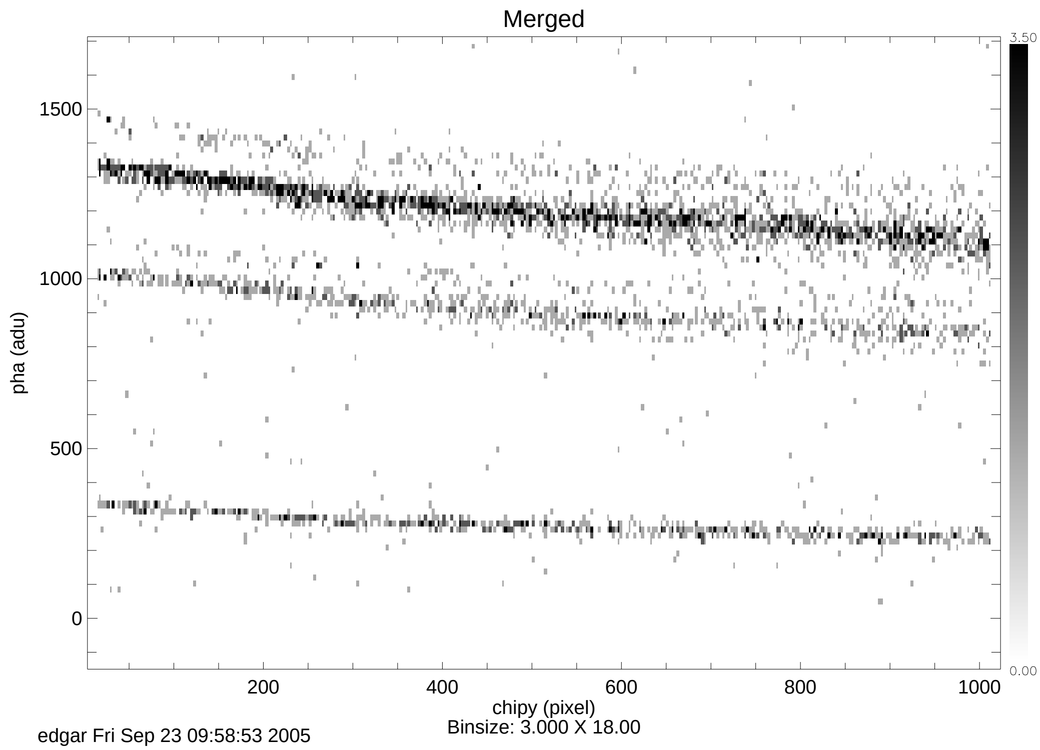

Figure 2: `PHA` vs `chipy` plot for a typical "good" column: Epoch 18–21, S4c1, `chipx`=482.

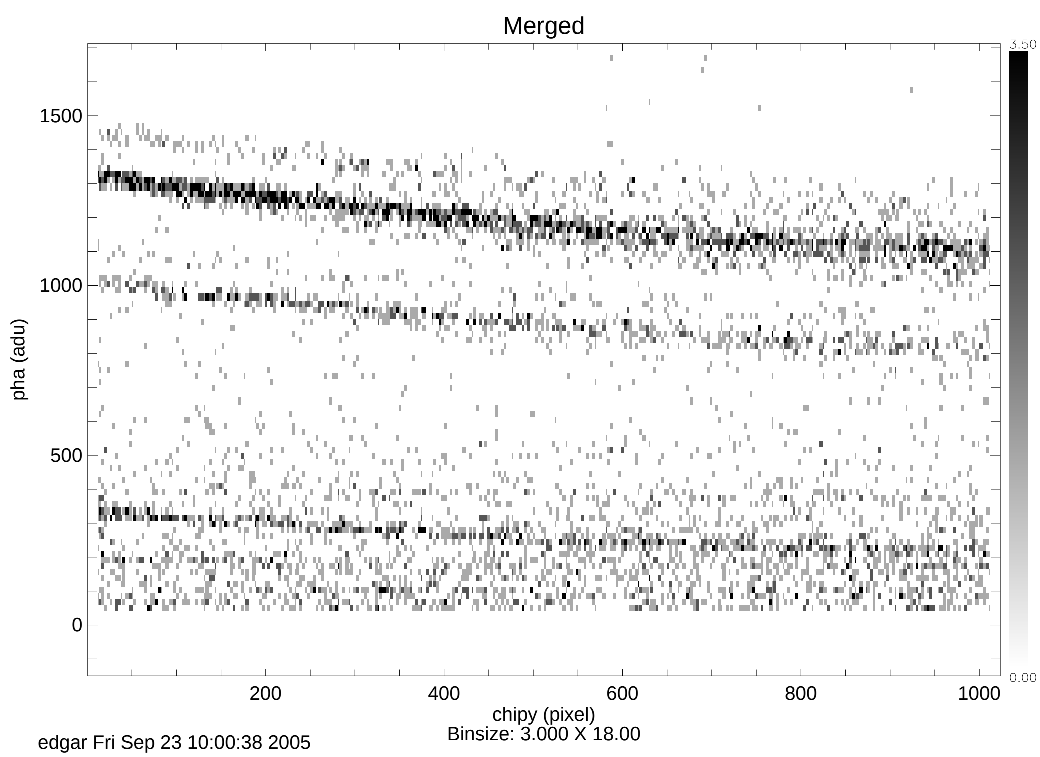

Figure 3: `PHA` vs `chipy` plot for a column with low-PH noise, Epoch 18–21, S4c1, `chipx`=487.

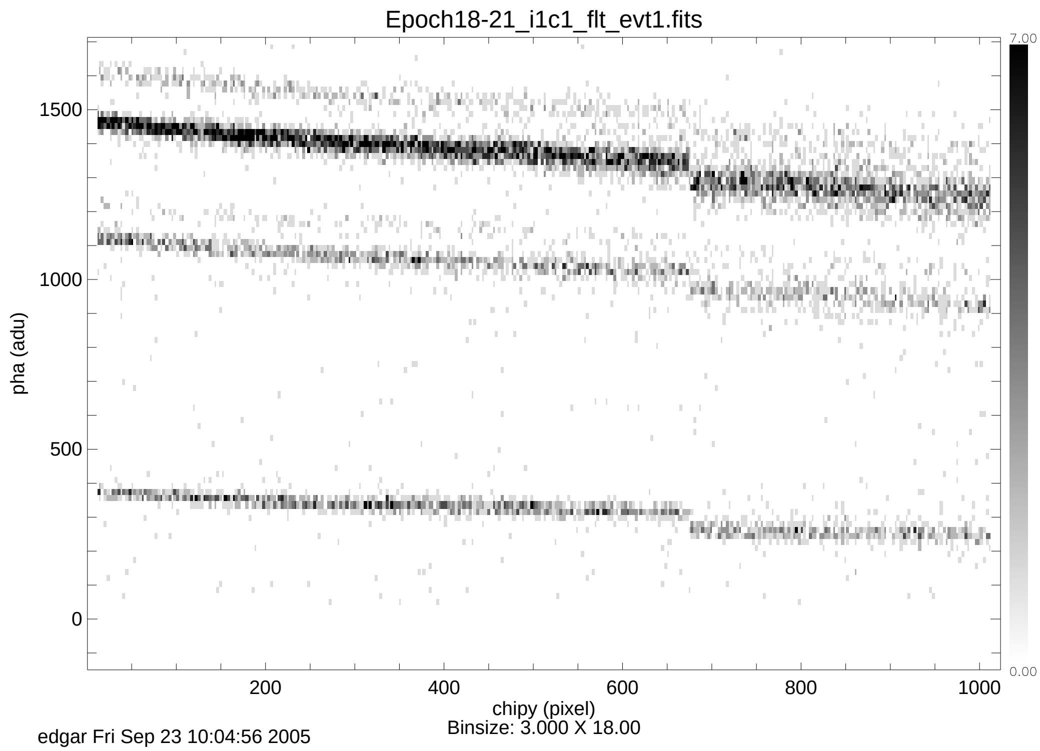

Figure 4: `PHA` vs `chipy` plot for a column with a small step-function, Epoch 18–21, I1c1, `chipx`=332.

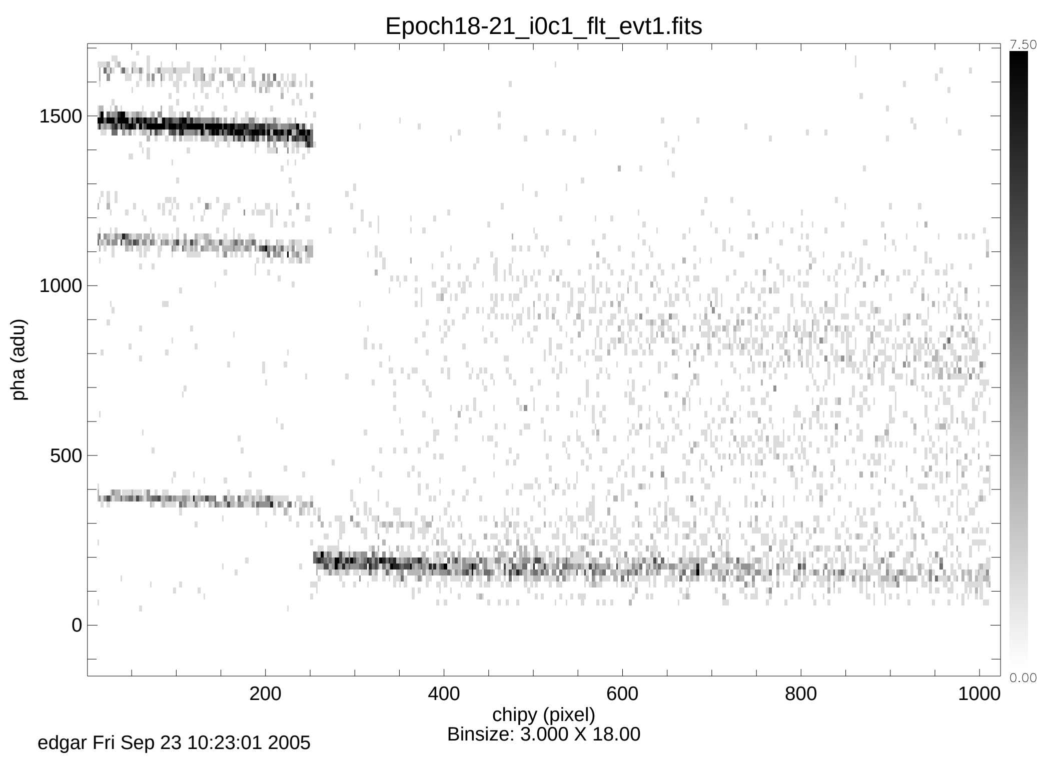

Figure 5: `PHA` vs `chipy` plot for a column with a large step-function, Epoch 18–21, I0c1, `chipx`=303. The line showing to the right of the step function is believed to be the Mn K–α line.

3.4 Response Files

Since the response of ACIS is improved by the CTI corrector, we need response matrices generated for use with CTI-corrected data. Alexey Vikhlinin (ref? TBR) provides these, as documented elsewhere. They come in two parts: an ideal CCD response, and a position-dependent CTI scatter matrix. These are convolved to generate response matrix files for many regions on the devices.

Also needed are time-dependent gain corrections. These correct the most obvious secular change in the instrument response, namely the decrease of mean `PHA` for a given photon energy as a function of time, due to radiation damage. They are issued quarterly, and are known as TGAIN files. Vikhlinin also generates these files based on the ECS data.

When all the response products are in hand, tests can begin using the CIAO software suite. Response matrices are generated for each and every zone of the FI chips, and the ECS data for select epochs are fit. Mean energies are compared to the known actual photon energies. The goal is to be correct to within 0.3% at all energies. In practice, this goal is difficult to meet at Al K–α, where this specification is about one ADU.

The ECS also has no strong lines softer than Al K–α, which at 1.49 keV is rather harder than we'd like. This situation can be remedied at some locations on some chips by fitting observations of 1E0102.2–7219 (known as "E0102"), an oxygen-rich supernova remnant in the Small Magellenic Cloud. This bright SNR is about 1.5 arcmin in diameter, and has a remarkably simple spectrum: lines of O VII, O VIII, Ne IX, and Ne X dominate the power emitted by the source. There is very little continuum, and no iron in the spectrum. Iron L-shell lines would complicate fitting of the Ne lines.

4 Testing

Catherine Grant has an IDL-coded CTI corrector that is useful for testing new trap map files. In order to judge whether the CXC process was adequate, we generated a trap map for I3 using data from epochs 1–4 (the year from February 2000 through January 2001), which was the first year of operation at a temperature of $-120°$C.

A plot of the summed data from epochs 1–4 for the entire I3c3 quadrant is shown in Figure 6. The pre-launch spectrum is also shown, for comparison. The two corrections are nearly identical, showing that the present trap map generation method is quite comparable to that of the PI team, released in the original CTI corrector CALDB. Also shown are plots of the mean `PHA` vs `chipy` (Figure 7), the FWHM vs `chipy` (Figure 8), and the fractional difference in the FWHM (Figure 9). These plots show that the present work is 1–2% better than the original PI team trap map, which is quite comparable.

We therefore proceeded to generate other trap map files.

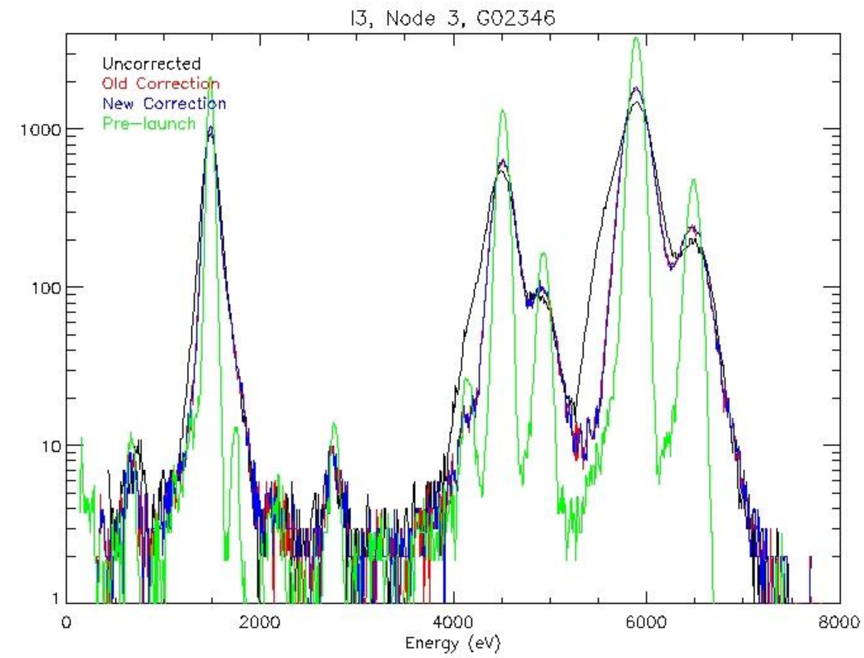

Figure 6: Spectra of I3c3, epochs1–4, at launch (green), uncorrected(black), and with two corrections which are nearly indistinguishable on this plot: the original PI Team work (red) and the present work (blue).

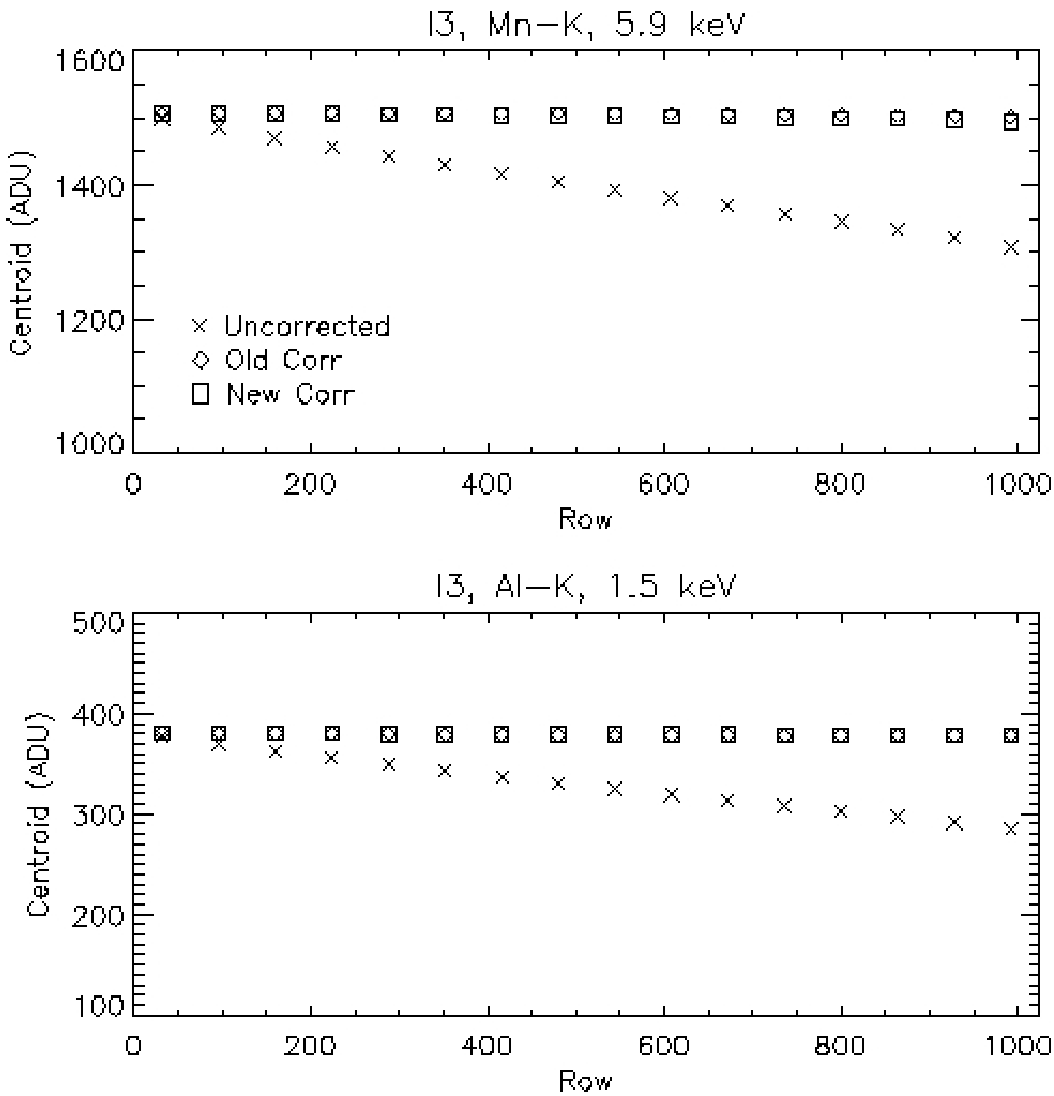

Figure 7: Plot of means for Mn and Al K–α lines vs `chipy`.

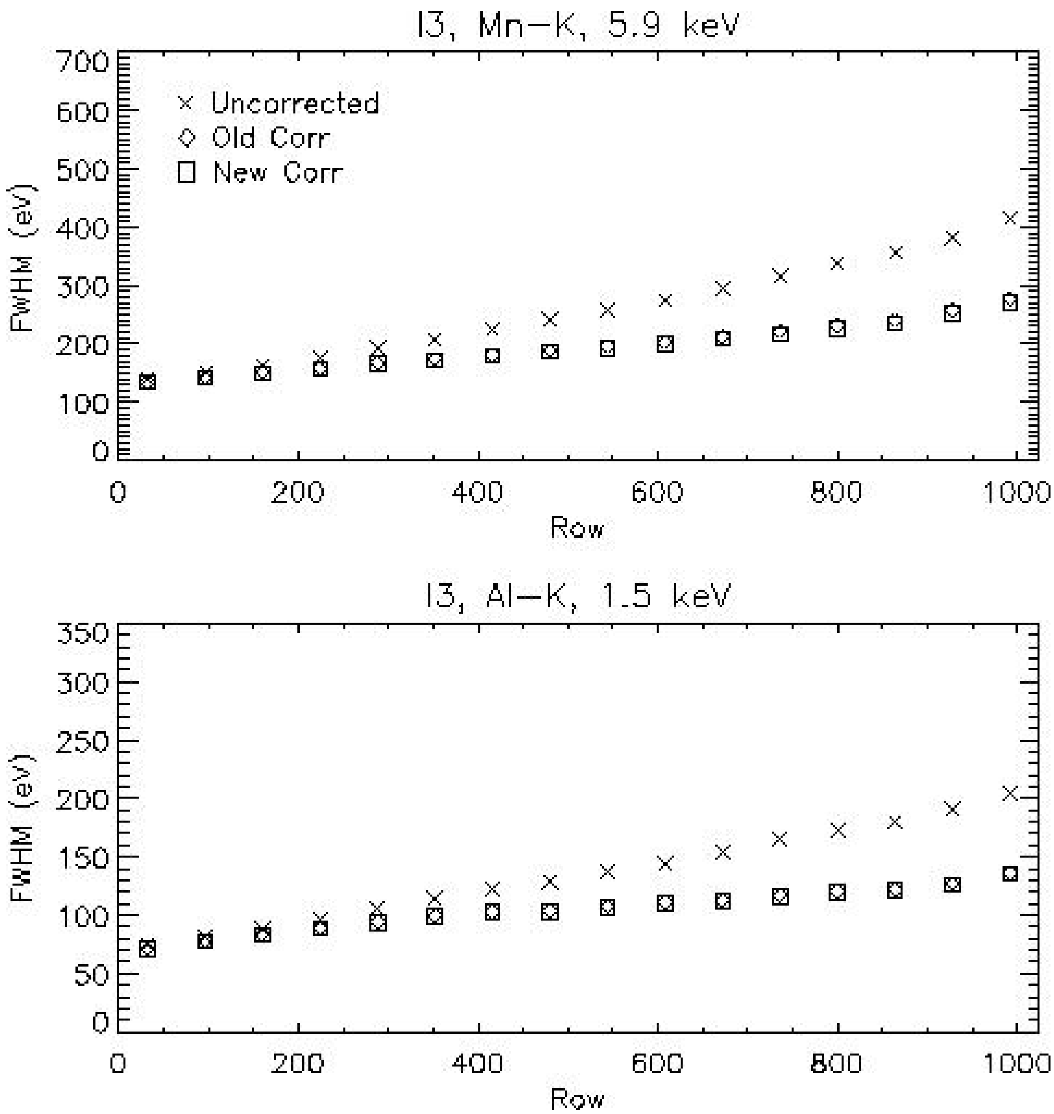

Figure 8: Plot of full-width at half-max for Mn and Al K–α. Note the two corrections are nearly identical.

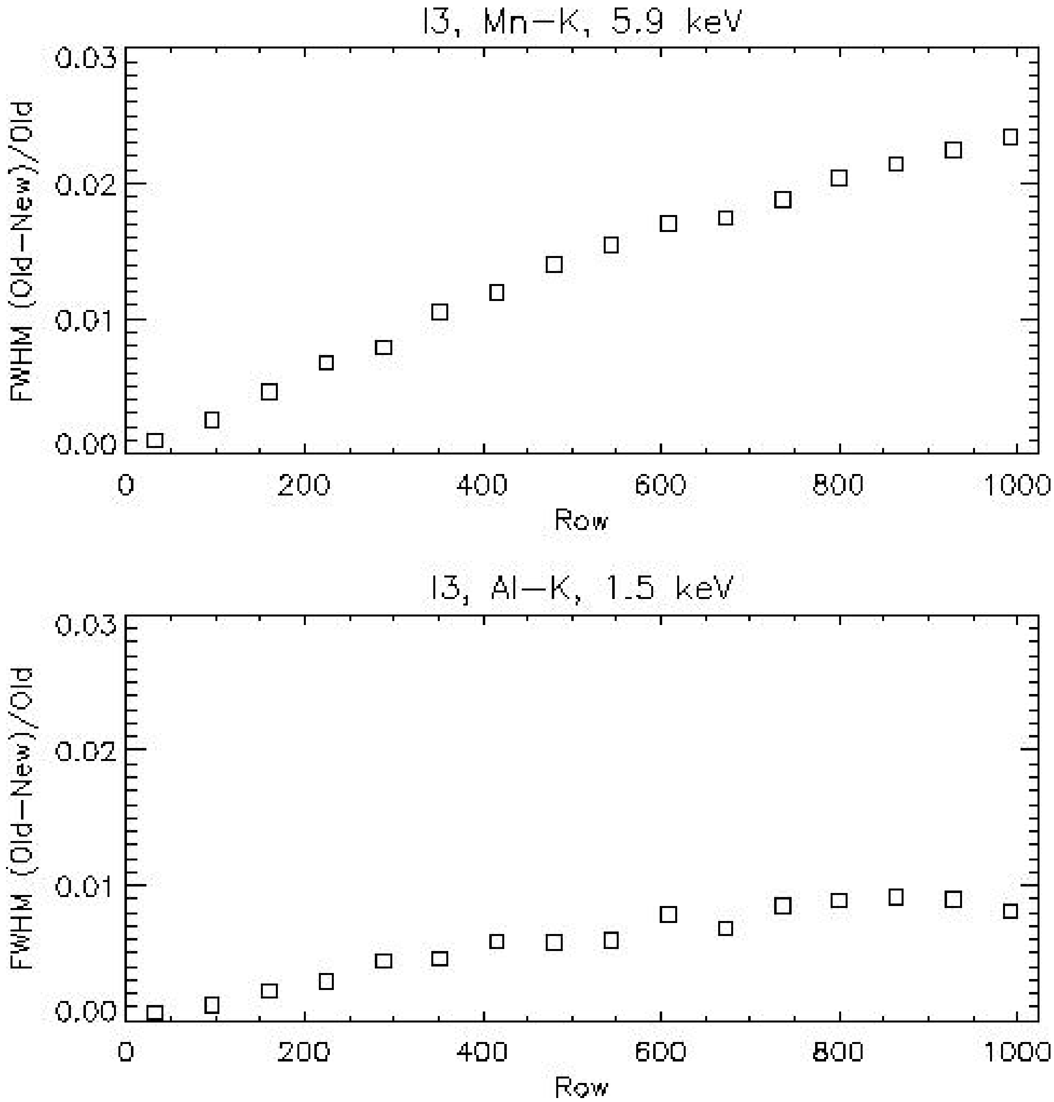

Figure 9: Plot of the fractional difference in FWHM between the CALDB correction and the present work. There is a 1–2% improvement with the present correction.

5 Creation of Trap Map Files for the BI Devices

The CTI on the Back-Illuminated (BI) devices (S3 and S1) is different in character from that on the FI devices. It arises from manufacturing defects rather than on-orbit radiation damage, and is changing only very slowly as the mission proceeds. It also exists not only in the imaging array (where the FI CTI is exclusively) but also in the frame store and the serial readout register.

It is difficult to tease apart parallel CTI in the imaging array and the framestore, so they are treated the same, with one additional parameter, which represents a pedestal value added to the detailed maps, representing the CTI in the frame store.

Serial CTI is treated in much the same way as parallel CTI, but must be separately calibrated and parametrized.

This section will detail differences in the procedure for trap map generation between the FI and the BI devices.

5.1 Serial CTI on the BI Devices

First we fit for the serial CTI. Data are taken from the first 16 rows, i.e. `chipy` < 17, filtered by grade and for bad pixels and columns. It's important to turn off any TGAIN and/or CTI correction in the data.

Even for a whole year, the collection of events that result are rather few. Thus fitting the K-shell lines is possible, but the weaker lines are nearly invisible. The Fe/Mn L complex can just be seen above the noise. This gives us six lines to work with.

It's also important to remember that the trailing pixel is either the left pixel (for nodes 1 and 3; this is `generic3` in Event Browser) or right pixel (for nodes 0 and 2; select `generic2`). Also, many of the slopes are negative, so it's important to take absolute values before fitting trends vs. energy. A last thing to note is that the intercepts are often far from zero, because while the serial CTI is essentially zero at the readout pixel, the column number there is in general non-zero.

The values of α and f obtained for the serial direction are often quite different from those for the parallel direction. This is expected, in part due to the different clocking speeds in the two directions. These numbers are placed in a file called `alpha_f_bi.rdb`, by way of files such as `S3_bi_alpha.rdb` and `trailing_fraction_s3_bi.rdb` and the script `cat_bi_results.csh`. There is one more column in these files than for the FI chips: a field called `serpar` is either `S` for serial or `P` for parallel parameters.

The serial parameters are fed to the IDL program `nkmap_bis.pro` which calls `ctiserialcalc.pro`. This takes a set of four event list files with names such as `Epoch0_s3c0_g0bot.fits`, whose name reflects filtering for only grade zero (or grades zero and six for S1), and the bottom 16 rows (`chipy` < 17). The program generates a serial trap map with a name such as `test_s3_0_trapdenss.fits`. Here, the zero is the epoch of the data used in the fit, and the s just prior to the dot represents serial CTI. This program does a linear fit to the pulse height of the aluminum line as a function of `chipx` for each quadrant. It can be confused by low pulse-height noise, so check to verify the results are sensible.

This file must be transformed into a CALDB-format trapmap, using Glenn Allen's S/Lang script `mk_cti_file.sl`. It's important to put 'none' in the place for the parallel trap map file in the input list for this file, since we wish to perform only the serial CTI correction on the data in the next step. At present, it is necessary to edit the name of the output cti file into the S/Lang script.

The resulting file can be used to run only the serial CTI correction on the event lists, using `acis_process_events`. If needed, a parameter study can be done at this point by varying α_s and f_s, regenerating the serial trap map, reformatting it, and correcting events lists. One can then examine the trend of line centroids and widths with `chipx` to determine the optimum parameters.

5.2 Parallel CTI on the BI Devices

The parallel trap map generation is much the same as for the FI chips, with the notable exception that one should use serial CTI-corrected event lists.

The trend of trailing pixel pulse height vs `chipy` does not go to zero at the bottom of the imaging array, because of the parallel CTI in the frame store. Estimating the (negative) chipy for which the trailing pixel would go to zero is one way of arriving at an estimate for the pedestal parameter. In practice, we obtained 1.6 for S3 and 7.0 for S1. We used the same values for both Epoch 0 (the −110 C era) and Epoch 1-4 (the first year of −120 C operation). The performance of the CTI corrector seems to be not very sensitive to the value of this parameter (which is hard-coded into the program `nkmap_bi_quad.pro`).

The programs which fit columns and generate trapmaps for the BI chips are called `nkquad_bi.pro` and `nkmap_bi_quad.pro`. Bad columns, of which there only a few on the BI devices, are hard-coded into the latter program (unlike the FI program, which read them from an rdb table).

The resulting trap map file, whose default name is something of the form `test_s3_0_trapdensp.fits` can be fed into the S/Lang reformatting program `mk_cti_file.sl` with its serial counterpart to generate a CALDB-format cti file. Trap maps for other chips can also be put into this file; the proper CALDB file has entries for all devices in a single file. The CALDB-format cti file can be used with `acis_process_events` to do a full correction (both serial and parallel) of event lists for testing purposes, such as parameter studies to obtain the best values of the parallel parameters α_p and f_p, and the pedestal value.

6 References

- Grant, C. E., Bautz, M. W., Kissel, S. M., and LaMarr, B. "A charge transfer inefficiency correction model for the Chandra Advanced CCD Imaging Spectrometer," SPIE, 5501, 177 (2004). Also astro-ph/0407199.

- Townsley, L. K., Broos, P. S., Garmire, G. P., and Nousek, J. A. "Mitigating charge transfer inefficiency in the Chandra X-ray Observatory Advanced CCD Imaging Spectrometer," ApJ, 534, L139 (2000).

- Bautz, M., Ford, P., Grant, C., Kissel, S., LaMarr, B., and Prigozhin, G. "ACIS CTI Correction," Internal MIT/CSR memo (22 March 2001). `http://space.mit.edu/ACIS/cticor.pdf`

- Grant, Catherine, Townsley, Leisa, and the ACIS IPI Team, "CTI Correction Approaches," MIT/CSR memo (26 April 2001) `http://space.mit.edu/ACIS/ctireport.pdf`