

Overview of the CE-STIS CTI Pre-Processor BETA

DRAFT

P. Bristow

January 2004

1. Introduction

The CE-STIS CTI pre-processor is a stand-alone processing tool for removing the effects of Charge Transfer Inefficiency (CTI) from STIS CCD raw data before it is calibrated with the standard CALSTIS pipeline. The pre-processor uses a CCD readout simulation to compute CTE effects in the image array for each raw exposure and in the appropriate dark reference file, corrects these and updates the header and bias file appropriately. The resulting “corrected raw” dataset can then be run through CALSTIS in the normal way.

In addition the script automatically runs CALSTIS on the corrected raw and the uncorrected raw and outputs various statistics allowing comparison between the two (for both calibrated and un-calibrated data) including extracted sources and extracted spectra.

We describe here briefly the operations performed by the pre-processor, the input it requires and the output it produces. Discussion of the application of the CTI correction to imaging and spectrographic data can be found in Bristow 2003a and Bristow 2003b respectively.

As the pre-processor is still undergoing testing and further development, many of the features below are subject to change and the command line output contains information that can be used by most users. We do not give here detailed instruction concerning installation and execution of the script, this information can be found in the accompanying README and script_notes.txt files.

2. Aims and Scope

We have developed a CCD readout simulation that predicts the effects of charge transfer during readout upon the charge distribution in the pixel array. This allows a predictive correction to be applied to the image array before standard CALSTIS calibration is applied to STIS CCD data.

The CTI pre-processor was developed during the testing and development of our CCD readout simulation software. We wanted to be able to apply the simulation-derived corrections to large quantities of data and have the results reduced to the extent that an immediate evaluation of the validity of the corrections could be made.

This actually becomes quite a complex task. The raw data must be partially processed before it is acceptable as input to the simulation. Each sub exposure must be dealt with individually. Dark and bias reference files must be treated in a consistent way. Finally the corrections derived from the simulation must be applied to the original raw data in a way that leaves it ready to be run through the CALSTIS pipeline as if nothing else was changed.

In order to assess the validity of the corrections applied we use the corrections of Goudfrooij & Kimble 2002 and Bohlin & Goudfrooij 2003 for photometric and spectroscopic data respectively, as a comparison. These corrections are applied automatically to extracted sources and spectra in data that has not been corrected for CTI at the pixel array level. Similar extraction is applied to the CTI corrected pixel array and sources matched and compared. (Source extraction is performed with SExtractor, Bertin 2003)

The resulting pre-processor pipeline undoubtedly performs a lot of tasks and produces a lot of output that is not useful to most users. However it does provide a relatively simple way to apply CTI correction to STIS CCD data along with information that will help the user to decide if the CTI correction has helped. While the model is still undergoing testing and development we provide the pre-processor as is to HST instrument scientists for evaluation. A stripped down version with less confusing diagnostics would be more appropriate when this software is released to a wider audience.

3. Using the CTI Pre-Processor

1. Calling the Script

```
ccd_script.py IPPSSOOT [asize=int, bsize=int, v_ext=int,  
offset=int, seed=int, trapfac=float]
```

The only required command line parameter that must be passed is the dataset name (dsname) without any suffixes or extensions. The script will look for dsname_raw.fits in the work directory. Other raw or calibrated products associated with this dataset are not required by the

script.

The optional command line parameters were included for development purposes and will rarely be needed by users.

asize	Number of rows of the array to be considered.	Default is 1024 (the entire array)
bsize	Number of columns of the array to be considered.	Default is 1024 (the entire array)
v_ext:	The number of rows in the vertical overscan.	Default is 19
offset	The horizontal offset in columns. Note bsize + offset <=10224	Default is 0
seed	Seed for random number generator.	Default is 1
trapfac	Scale factor for the trap density normalisation. (Note that the trap density is in any case scaled according to the date of the observation.)	Default is 1.33.

2. Header Information

The following header keyword values are extracted from dsname_raw.fits[0]:

TDATEOBS: The date the observation took place. This allows the correct scaling of the trap density.

NEXTEND: The number of fits extensions. In addition to [0], there are 3 for each exposure, the first of which contains the image array. We will need to process 1, 4, 7, ..., NEXTEND-2, in total NEXTEND/3 exposures. In the case of only one exposure the CALSTIS final product will be *_flt.fits, when there are two or more it will be *_crj.fits.

TEXPTIME: The total exposure time (not currently used)

CCDAMP: The readout amplifier used. 'A' and 'C' are rarely used and not supported by this script. 'D' corresponds to the register at the top (row 1043) of the STIS CCD. 'B' corresponds to the register at the bottom (row 0) of the STIS CCD, B amp exposures require rotation before being passed to the CTE code.

CCDGAIN: The gain of the readout amplifier. Pixel values are multiplied by the gain before being passed to the simulation (as we are interested in the actual size of the charge cloud during readout.). Output pixels are then divided by the gain.

BINAXIS1: X axis binning. The script only supports unbinned (BINAXIS1=1) data. Datasets with BINAXIS1!=1 are rejected.

BINAXIS2: Y axis binning. The script only supports unbinned (BINAXIS2=1) data. Datasets with BINAXIS2!=1 are rejected.

OBSTYPE: Denotes either imaging or spectroscopic data. Required for the comparison to empirical data.

DARKFILE: The reference dark file for the standard pipeline.

BIASFILE: The reference bias file for the standard pipeline.

3. Dark Reference File

A CTE corrected dark file is computed as follows:

- i) A copy of the reference dark file is put into the `post_corr` directory.
- ii) The reference is multiplied by 1100 (to make the equivalent of a 1100s dark)
- iii) `cte_el` is run with the 1100s dark as the input charge array and an empty dark file. Gain is set to 1.0, not the dataset header value (dark files are always obtained with `gain=1.0`) and the readout amplifier is set to be D (true for all dark exposures to date). Initial trap state is appropriate to the exposure.
- iv) Output is a corrected dark (the "`orig_aa.fits`" file), this is divided by 1100 to get back to the 1s normalisation.
- v) The corrected 1s dark is copied to the appropriate section (in the event that it is not the entire array) of the reference dark in the `post_corr` directory.

Further discussion of the treatment of darks in the CTI pre-processor can be found in Bristow 2004b.

4. Bias Reference File

The reference bias file contains hot columns which are automatically dealt with by `cte_el` but also some structure which `cte_el` does not account for. In order that the hot columns are not removed twice we need to modify the reference bias by subtracting the bias created by `cte_el` from the reference bias:

- i) A copy of the reference bias file is put into the `post_corr` directory.
- ii) `cte_el` is run with the a zero input charge array and the corrected dark file. (Gain is irrelevant, initial trap state is empty.)
- iii) Output is a bias file (containing just hot columns), this is multiplied by the gain and subtracted from the appropriate section (in the event that it is not the entire array) of the reference bias in the `post_corr` directory.

If the header did not contain a bias reference file name then the above is omitted. Further discussion of the treatment of darks in the CTI pre-processor can be found in Bristow 2004b.

5. Preparation of Raw Data for Simulation

We want the input to the simulation to resemble the charge distribution on the chip before read

out as closely as possible. Therefore we need to remove the bias. This is best done by running CALSTIS with the switches BLEVCORR and BIASCORR both set to PERFORM and all other processing switches set to OMIT. Note at this stage the bias reference file used is still the uncorrected one.

6. Extract Statistics from Overscan Regions

This is a step included for our own analysis. It gathers information on how the mean and variance vary in the overscan regions in real data.

7. Execute the Readout Simulation Code

The input charge distribution is the raw data prepared as in 5 above. The dark file is that corrected in step 3 above. Gain is set according to the header file and the initial state of traps is appropriate to the charge distribution.

The output is summarised in section 5 below

8. Repeat the Steps 6 and 7 For All Exposures

In doing so, the processed directory is filled with fits files containing the simulation output for each extension. Meanwhile the copy of the multi-extension raw fits file in the post-corr directory is updated with the corrected section of the image array in each of its exposure extensions.

Once all exposures have been corrected, the copy of the raw data in post_corr contains a complete set of corrected exposures (or subsections of each) which are ready to be processed with CALSTIS.

AT THIS STAGE YOU HAVE A CORRECTED RAW DATASET WHICH MAY BE USED AS YOU WOULD HAVE USED ANY RAW STIS DATASET EXCEPT THAT THE APPROPRIATE DARK AND BIAS REFERENCE FILES ARE THOSE CREATED IN STEPS 3 AND 4 ABOVE.

The script continues to calibrate the data and also make a reference copy of calibrated data without simulation corrections and then compare the two.

9. Calibrate the Data with CALSTIS

The original set of processing switches is used except that the reference dark and bias files are replaced with those created in steps 3 and 4 above. The science data extension is kept as post_corr/dsname_corr_proc.1.fits (This is the first extension of the dsname_crj.fits file for datasets having more than one exposure, otherwise it is the first extension of the dsnameflt.fits file)

A second copy of the raw data, which has none of the simulation derived corrections, is also run through CALSTIS with standard header switches and reference files. The science data extension is kept as post_corr/dsname_proc.1.fits (This is the first extension of the dsname_crj.fits file for datasets having more than one exposure, otherwise it is the first extension of the dsnameflt.fits file)

For imaging data `sextractor` is run on the two `*_proc.1.fits` files the detection lists are fed to `xcorr.py`. The results are kept in `post_corr/dsname_stats.dat` (the accompanying `post_corr/dsname_stats.col` is needed to view this with `xgobi`)

For spectroscopic data `qxspec.py` is run on the two `*_proc.1.fits` files and the results are kept in `post_corr/dsname_spec.dat`

4. Additional Analysis

The CTI pre-processor automatically runs one of two further analysis scripts, one for imaging data, the other for spectroscopic data.

1. *xcorr.py*

The purpose of `xcorr.py` is to provide a measure of the effect of the CTE correction upon sources in the image array.

To obtain source lists we run `sextractor` on the corrected and uncorrected data. This can be done on the raw exposures or on the fully calibrated (and CR cleaned) data. The application to raw exposures should provide the better comparison to empirical corrections as the signal to noise is altered in the calibrated data, however the cosmic rays tend to confuse the results.

`Sextractor` parameter files are included in this distribution that ensure that the list of parameters for each source is as expected by the `xcorr.py` script.

`xcorr.py` searches the the two source lists and matches sources with similar (within 1.0 pixel) x and y co-ordinates. For each match the following parameters are recorded:

- 1) Object number
- 2) X position in raw data (pix from left)
- 3) Y position in raw data (pix from bottom)
- 4) Aperture flux in raw data (ADU)
- 5) Peak flux in raw data (ADU)
- 6) Area within 4σ isophote in raw data (sqr pix)
- 7) Aperture flux in raw/Aperture flux in corrected data
- 8) Aperture flux/empirically* corrected flux computed with global sky background
- 9) Aperture flux/empirically* corrected flux computed with local sky background

- 10) 7/8
- 11) 7/9
- 12) SExtractor "best" flux in raw data/SExtractor "best" flux in corrected data
- 13) 12/8
- 14) Peak flux in raw data/peak flux in corrected data
- 15) FWHM in raw data - FWHM in corrected data(pixel)
- 16) Elongation in raw data / elongation in corrected data
- 17) Isophotal area (4σ) in raw/Isophotal area (4σ) in corrected
- 18) X position in raw data - X position in corrected data
- 19) Y position in raw data - Y position in corrected data
- 20) Star-galaxy classification in raw data
- 21) Star-galaxy classification in corrected data
- 22) Local background in raw data
- 23) Local background in corrected data
- 24) 22/23
- 25) Corrected data extraction list no.
- 26) Raw data extraction list no.

* The empirical correction applied is that of Goudfrooij & Kimble 2002

The multi-variable results can be plotted with `xgobi` (or a suitable `sm` script). As well as the possibility to see how the simulation correction affects the measured flux, shape and position of sources, the empirical estimate (derived by applying Goudfrooij and Kimble's algorithm to the uncorrected raw data source parameters) allows us to compare simulation and empirical corrections. Naturally agreement should only be expected for the sort of stellar sources for which the empirical correction was calibrated (i.e. limit parameters 20, 4 and 6).

Output is printed to the terminal, this should be redirected to a suitably named file.

This script was used extensively in deriving the results for Bristow 2003a.

2. *qxspec.py*

This provides a similar evaluation of the simulation corrections as `xcorr.py` but for spectroscopic data.

The application to raw exposures should provide the better comparison to empirical corrections as the signal to noise is altered in the calibrated data, however the cosmic rays tend to confuse the results.

1. The uncorrected data is scanned and the rows containing the highest signal in a seven pixel tall aperture (as Bohlin & Goudfrooij 2003), presumably containing the spectrum of interest, is found. Cosmic rays and hot pixels that, for short exposures, could lead to false identification of the rows of interest are excluded by summing the minimum value pixel in each horizontal group of four.
2. The backgrounds to be used for each column are derived by smoothing the pixel array and finding the minimum pixel value between the spectrum and the readout amp. Note: choosing the minimum value is not what one might expect from CTE trapping theory (e.g. Kimble et al 2000), however Bohlin & Goudfrooij 2003 report that this gave the best results for their empirical correction.
3. For each bin of the spectrum (which contains the counts from the seven pixel high extraction), the empirical correction is computed using the distance from the readout register, signal and background as computed above and the date of the dataset using the model of Bohlin & Goudfrooij 2003.
4. The same seven rows are extracted from the simulation corrected data. This time, for each bin, the ratio of uncorrected to corrected counts is computed. Output is printed to the terminal, this should be redirected to a suitably named file.

This script was used extensively in deriving the results for Bristow 2003b.

5. Summary of Input/Output

The script will pre-process all STIS CCD imaging and spectroscopic datasets, *with the following restrictions:*

CCDAMP	D or B	Readout amplifier D or B was used. All science exposures to date use D.
BINAXIS1	1	Data must be unbinned
BINAXIS2	1	

The script obtains from the dataset header the following essential information:

TDATEOBS	The epoch of the exposure, for scaling the radiation level to that present at the time of the exposure.
----------	---

CCDAMP	Determines in which direction the charge was read from the chip
NEXTEND	The number of FITS extensions (=3x the number of exposures)
OBSTYPE	Imaging or spectroscopic data, for statistical analysis
X1DCORR, X2DCORR	Switches determining which products CALSTIS will produce
DARKFILE	The dark reference file required to calibrate this dataset
BIASFILE	The bias reference file required to calibrate this dataset

The dark and bias reference files must also be available to the script. In addition, all other reference files that would be required to preprocess this dataset through CALSTIS with the standard header settings (eg. .wav), must also be available.

Output are the following:

processed/dsnameN_diff.fits	This is the difference between the input charge distribution (after gain scaling and removal of -ve values) and the final charge distribution for the nth extension. It is subtracted from the appropriate section of the raw data to give post_corr/dsname_DIFF.fits (below).
post_corr/dsname_DIFF.fits	This is processed/dsnameN_diff.fits subtracted from the appropriate section of the raw data. This is continually overwritten as each extension is processed.
post_corr/dsname_corr_raw.fits[N]	The Nth extension raw data with the corrected section pasted in at the appropriate location.
processed/dsnameN_orig.fits	Same as post_corr/dsname_DIFF.fits but a copy of this is kept in the processed directory for each fits exposure data extension, hence the subscript N.
processed/dsnameN.par	A copy of the simulation parameters for the Nth extension (in practice usually the same

	for each extension).
processed/dsnameN_input.fits	A copy of the input charge distribution (after gain scaling and removal of -ve values) for the Nth extension.
processed/dsnameN_reg.fits	A copy of the final charge distribution in the simulation for the Nth extension.
processed/dsnameN_ovscan.log	The overscan information extracted (see above) from the Nth extension.
IMAGING DATA (xcorr.py):	
processed/dsnameN_stats.dat	Output from xcorr.py (see below) for raw sub-exposures
processed/dsnameN_stats.col	Column headings for viewing processed/dsnameN_stats.dat with xgobi.
post_corr/dsnameN_stats.dat	Output from xcorr.py (see below) for the calibrated dataset
post_corr/dsnameN_stats.col	Column headings for viewing processed/dsnameN_stats.dat with xgobi.
SPECTROSCOPIC DATA (qxspec.py):	
processed/dsnameN_spec.dat	Output from qxspec.py (see below) for raw sub-exposures
Post_corr/dsname_spec.dat	Output from qxspec.py (see below) for the calibrated dataset

6. Summary

We have described the CE-STIS CTI pre-processor in its current of development. The pre-processor script allows CTI correction and subsequent CALSTIS calibration to be applied to STIS CCD data with minimum interaction from the user whilst also providing copious additional data to help the user assess the effect of the CTI correction upon their data.

References

- Bertin, E., <http://terapix.iap.fr/soft/sextractor/index.html>. 2003
- Bristow, P. CE-STIS-ISR 2003-001, in preparation 2003a

Bristow, P. CE-STIS-ISR 2003-002, in preparation 2003b
Bristow, P. CE-STIS-ISR 2004-002, in preparation 2004
Bohlin, R. and Goudfrooij, P., STIS ISR 2003-03, 8th April 2003
Goudfrooij, P. & Kimble, R. A. in "Proceedings of the 2002 HST Calibration Workshop", Eds Arribas, S.,
Koekemoer, A and Whitmore, B. Space Telescope Science Institute 2002
Kimble, R. A., Goudfrooij, P. and Gililand, R.L., Proc. SPIE Vol. 4013, p. 532-544, UV, Optical, and IR
Space Telescopes and Instruments, James B. Breckinridge; Peter Jakobsen; Eds. 2000