SPACE TELESCOPE SCIENCE INSTITUTE
Operated for NASA by AURA

Instrument Science Report NICMOS 2009-007

CalTempFromBias: Implementation and Testing

N. Pirzkal, D. Grumm, E. Bergeron
Jul 21, 2009

ABSTRACT

This ISR describes the `CalTempFromBias` *program which was developed to estimate the temperature of each of the three NICMOS detectors individually. Unlike previous NICMOS temperature estimates, which are derived from a mounting cup sensor, these temperatures are derived using the bias level of the detectors themselves. The uncertainty associated with these temperature-from-bias values are 0.05K and are the basis for the new temperature-dependent calibration products for NICMOS data. This ISR describes the program* `CalTempFromBias`*, its options, and its applications.*

Introduction

Temperature-from-bias is a technique that allows one to infer the temperature of each of the three NICMOS detectors using nothing but the bias information in a NICMOS `_raw` file. This method allows for a direct temperature estimate to be made at the beginning of the observation. The only requirement is that the observation must be a MULTIACCUM dataset with more than 1 NSAMP. This technique, to be described in more detail in Bergeron 2009, NICMOS ISR in preparation, relies on the principle that each detector can be thought of as a diode. Because the behavior of this diode is temperature-dependent, one can estimate the temperature of this diode (i.e. detector) by calibrating the temperature-bias level relation for each detector. Two slightly different techniques have been developed to relate bias levels to temperature, the Blind correction and the Quietest Quad correction. These are to be described in detail in Bergeron 2009, NICMOS ISR in preparation and `CalTempFromBias` implements both of these techniques. The default method however is the one that results in the lowest temperature-from-bias error es-

timates, the Blind correction. The typical uncertainty of the Blind correction is 0.05K while the uncertainty of the Quietest Quad method is 0.10K.

As part of an effort to use this method to deduce more accurate temperature of the detector (the previous method relied on a mounting cup temperature sensor that is not directly connected to the detectors but is instead attached to the back side of the mounting plate), a new program was created.

The goal of this effort is to be able to compute and use more accurate estimates of the temperature of each of the NICMOS detectors separately. This is to allow for temperature dependent calibration files of the NICMOS data, as described in ISR 2009-003.

CalTempFromBias

In order to facilitate the computation of temp-from-bias temperatures, a new software, `CalTempFromBias` was written. This software was written in Python and can be easily run on unprocessed NICMOS `_raw` files. `CalTempFromBias` relies on the ability to measure the bias level of the first IMSET in a MULTIACCUM dataset and hence cannot be applied to calibrated `_cal` files which have already been bias subtracted.

`CalTempFromBias` can be run from the command line directly, from within the Pyraf environment, as described below. It is also run in the OPUS pipeline as described below.

Requirements

External Files

`CalTempFromBias` requires the use of the NICMOS non-linearity calibration file. The name of that file is contained inside the FITS header keyword `NLINFILE` of the `_raw` NICMOS FITS files. It is available as part of the CDBS. By default, `CalTempFromBias` will search for this file in whatever location the environment variable `nref` points to. This can be overwritten using the command line option `-n NREF_PAR`, where `NREF_PAR` points to the location of a directory containing the NICMOS calibration files.

NICMOS Data

`CalTempFromBias` will only run on `_raw` NICMOS datasets containing MULTIACCUM data with at least 1 IMSET (i.e. NSAMP>=1). Any attempt to use `CalTempFromBias` on other types of NICMOS data will result in `CalTempFromBias` exiting after displaying an error. There are no option to force `CalTempFromBias` to attempt to compute a temperature-from-bias for any other NICMOS dataset types.

Header Keywords

`CalTempFromBias` will by default expect some keywords to be populated in the main FITS extension of the `_raw` NICMOS file. The main keyword that the program expects is `TFBCALC`. This keyword should be automatically populated by the MAST archive when data is requested. If set to `PERFORM`, `CalTempFromBias` will attempt to compute a temperature from the bias level of the 0th read. If this is successful, `CalTempFromBias` will update a few keywords in the header of the original file (`TFBMETH`, `TFBDATE`, `TFBVER`, `TFBTEMP`, and `TFBERR`) and set `TFBDONE` to `PERFORMED`. These keywords are listed in Table 1. Note that the data array of the `_raw` file is not modified but that the header of the file is modified. If `CalTempFromBias` is unsuccessful, the `TFBDONE` keyword will be set to `SKIPPED`.

If the `TFBCALC` keyword is not present, or is set to `OMIT`, `CalTempFromBias` will not compute the temperature from bias. To force the temperature-from-bias to be computed, he `--force Blind` option should be used. The various options are listed in Table 2.

Table 1: Header keywords related to the temperature-from-bias computation.

Keyword	Description	Possible Values
TFBCALC	Control keyword	PERFORM/OMIT
TFBMETH	Method used by `CalTempFromBias`	Blind/Quietest/None
TFBDATE	Date TFBCALC was run	Date string
TFBVER	Version of `CalTempFromBias`	2.03 (currently)
TFBTEMP	Temperature (Kelvin) computed by `CalTempFromBias`	Float
TFBERR	Error Estimate of the temperature (Kelvin) computed by `CalTempFromBias`	Float
TFBDONE	Status of `CalTempFromBias`	OMITTED/SKIPPED/PERFORMED

Table 2: The command line options of `CalTempFromBias`.

Option (short name)	Option (long name)	Outcome
-h	--help	Displays help message

Option (short name)	Option (long name)	Outcome
-q	--quiet	Displays nothing on the screen while running
-v	--verbose	Verbose mode, display information about the temperature-from-bias computation
-e EDIT_TYPE	--edit EDIT_TYPE	Type of the file to edit. Default is RAW. This should not be changed since only NICMOS `_raw` files are supported.
-k HDR_KEY	--hdr_key=HDR_KEY	Header keyword name to use to store the computed temperature. Default is TFBTEMP.
-s ERR_KEY	--err_key=ERR_KEY	Header keyword name to use to store the computed temperature. Default is TFBERR.
-n NREF_PAR	--nref_dir==NREF_PAR	Location of the NICMOS calibration file. This can be used to direct `CalTempFromBias` to the location of the NICMOS nonlinearity file whose name is stored in the NLINFILE header keyword.
-f FORCE	--force=FORCE	Forces `CalTempFromBias` to compute a temperature, even of TFBCALC is not present or set to PERFORM. Can be set to 'Blind'/'Quietest Quad'/'Auto' or 'B'/'Q'/'A'. The default is 'Blind'.
-c NOCLEAN	--noclean=NOCLEAN	Option to not avoid cleaning the 0th read. Default is False. This option should not be used since uncleaned bias frames will produce incorrect temperatures.
-d	--do_not_write_keys	Do not update or write any keyword in the original `_raw` file.

Stand Alone Mode

`CalTempFromBias` can be used from the command line on a single NICMOS `_raw` file.

The program is installed during the STSDAS installation process in the `nictools` subdirectory of the main Python `site-package` directory (This location varies depending on whether you are running on a Solaris, Linux, or OSX system and in the later case whether your version of Python is within an OSX Framework or not.)

Upon reading the file, `CalTempFromBias` will read the header keyword `TFBCALC` and compute the temperature-from-bias value if the `TFBCALC` keyword is set to `PERFORM`, and populate the `TFB*` header keywords listed in Table 1. For example, to run `CalTempFromBias` on a single NICMOS datatset:

Input:

```
CalTempFromBias.py n8tf30juq_raw.fits -v
```

Output:

```
Temp_from_bias run on  Sat Jun 20 17:12:18 2009 , version:  2.03


Calculating temp for file  0  :  n8td01ajq_raw.fits

Camera:  3
The results of the quadmean call:
[-20810.88593673 -20783.12708154 -20398.13165191 -20031.85854116]
 WARNING : TFBCALC is not set in the input file.
 Algorithm - Blind Correction:  76.5515634619  (K) +/-  0.05  (sigma)
 The algorithm used is  BLIND CORRECTION
 The headers have been updated.
 The parameters used are :
  input_file list:   n8td01ajq_raw.fits
  edit_type:   RAW
  hdr_key:  TFBTEMP
  err_key:  TFBERR
  nref_par:   /grp/hst/cdbs/nref/
  force:  Blind
  noclean:  False
  dry_run:  None
  verbosity:  2
 For the files given by the input file list:
  input_file[ 0 ]:   n8td01ajq_raw.fits
  nonlinearity file[ 0 ]:  /grp/hst/cdbs/nref/na20854kn_lin.fits
```

The temperature shown above using the Verbose mode (-v) is automatically entered in the header keyword `TFBTEMP` and an estimate of its error is entered in `TFBERR`, as shown in Table 1. In the event that `TFBCALC` is not set in the header of the file, as it can be the case with older NICMOS datasets, one can force `CalTempFromBias` to run using the `-force Blind` option on the command line.

Pyraf Mode

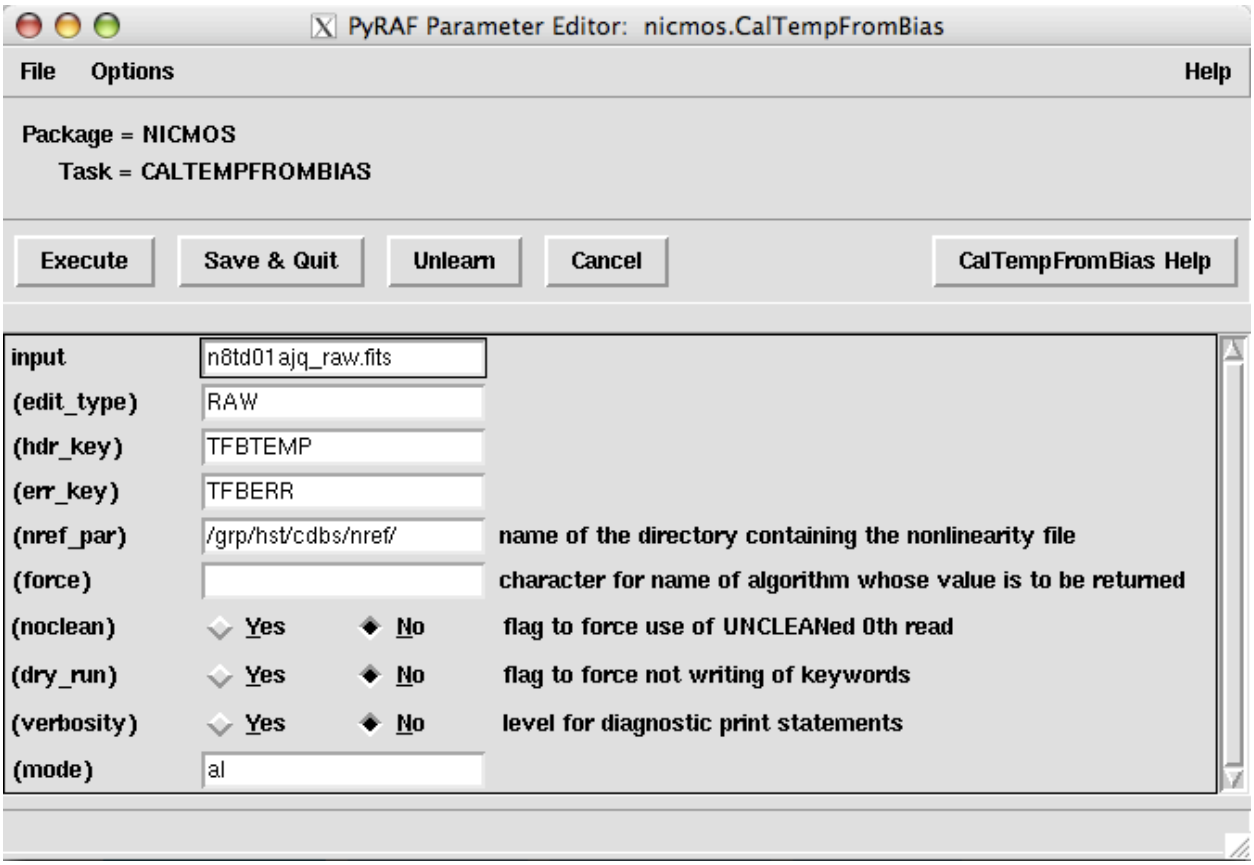
`CalTempFromBias` can also be run from within the Pyraf environment, as part of the `STSDAS/HST_CALIB/NICMOS` package within Pyraf. It requires Pyraf to run and will not run on in stand-alone IRAF environment. The Pyraf access to the `CalTempFromBias` code offers exactly the same options listed in Table 2. One can access an EPAR Pyraf panel for this task as for any other Pyraf task, using:

```
stsdas
hst_calib
nicmos
epar CalTempFromBias
```

Figure 1 shows the resulting panel that is then displayed.

Use of `@filelist` as input is also possible, in which case the `CalTempFromBias` code will be run on all the files listed in the file `filelist`. which is a text file listing the names of the nicmos files, one per line.

Figure 1: `CalTempFromBias` epar panel using Pyraf



Pipeline Mode

The `CalTempFromBias` code was designed to be integrated in the MAST OPUS system as part of the NICMOS calibration pipeline. This mode is identical to the Stand Alone Mode outlined above. The `CalTempFromBias` code was however designed so that its actions can be controlled using a set of FITS header keywords in the NICMOS data file itself. These are also listed in Table 1. When running the `CalTempFromBias` with no options on a FITS file, `CalTempFromBias` will examine the header keyword `TFBCALC` and, if it is set to `PERFORM`, `CalTempFromBias` will compute the temperature-from-bias. If `TFBCALC` is set to `OMIT`, `CalTempFromBias` will not compute any temperature. If the process completes successfully, the `TFBDONE` keyword is set to `PERFORMED` and the `TFBTEMP` and `TFBERR` header keywords are populated with the estimates of the temperature-from-bias and error, respectively.

Python Mode

The `CalTempFromBias` code can be integrated into pure Python code, with no dependence on IRAF or Pyraf. In this case, a temperature can be computed for a _raw NICMOS FITS file using for example:

```
from nictools import CalTempFromBias
tfb =CalTempFromBias.CalTempFromBias("n8td01ajq_raw.fits",force="B"
         ,noclean=False,dry_run=1,verbosity=1)
[temp, sigma, winner, in_flag, dry_run ]= tfb.calctemp()
tfb.update_header(temp,sigma,winner,edit_type="RAW"
         , hdr_key="TFBTEMP",err_key="TFBERR")
```

will read the file, compute the temperature-from-bias temperature and populate the header keywords `TFBTEMP` and `TFBERR`. The constructor `CalTempFromBias` as well as the methods calctemp() and update_header() are shown with the optional parameters explicitly listed. These constitute the API of `CalTempFromBias` that can be used directly from native Python code.

Testing

`CalTempFromBias` has been successfully run on all available NICMOS MULTIACCUM datasets. For the purpose of this ISR, we have run it on about 50,000 NIC1, NIC2, and NIC3 datasets. As of the writing of this ISR, `CalTempFromBias` is now part of the default NICMOS calibration and is automatically run on all appropriate NICMOS data retrieved via the MAST archive.

Pre-NCS comparison

The algorithms and corresponding coefficients of `CalTempFromBias` were calibrated so that the temperature that program returns match those of the warm up phase of the pre-NCS epoch (i.e. before the year 2002). This will be described in detail in Bergeron 2009, NICMOS ISR in preparation. The intrinsic uncertainties in the temperature estimates derived from the bias levels are 0.05K. They are sensitive enough that inter-orbit temperature variations, as well as variation of the temperature of the detectors related to the previous use of these detectors, can be measured. The following plots show the pre- and post-NCS temperatures computed for the three detectors. Figures 2, 3, and 4 show the mounting cup temperature of NIC1 (green, as obtained using the NDWTMP11 keyword in the _spt associated FITS files) for all three NICMOS detectors between 1998 and 1999. The temperature-from-bias values are shown in red.

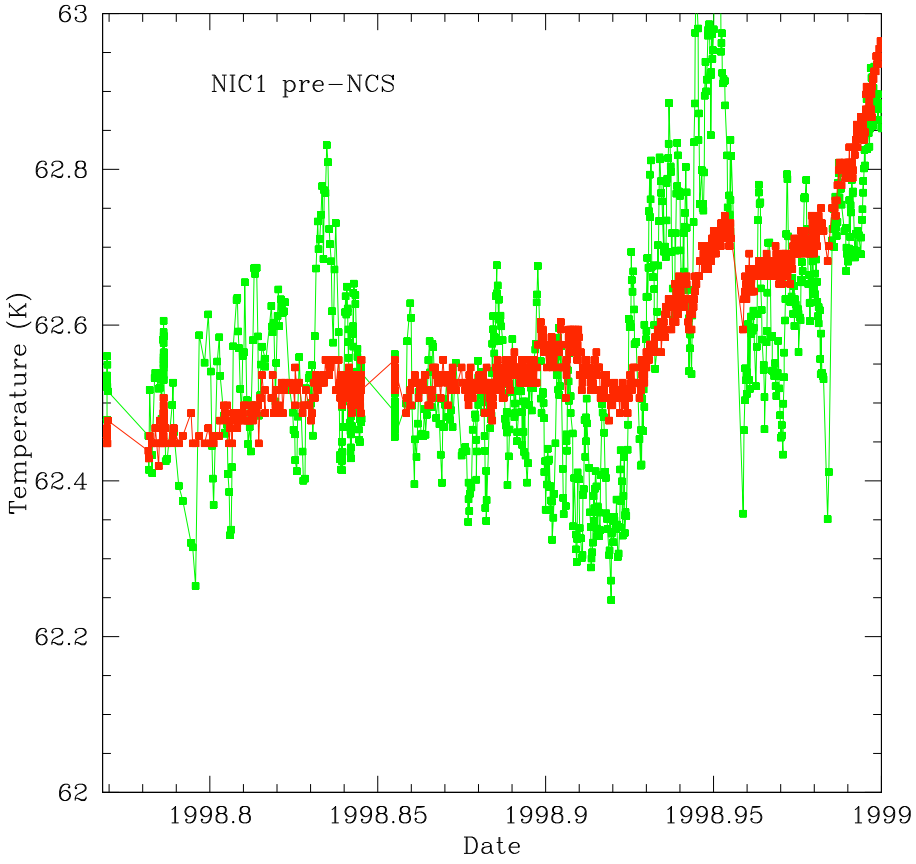


Figure 2: Mounting Cup temperature for NIC1 (red) and temperature-from-bias (green) for the pre-NCS period of 1998 to 1999.

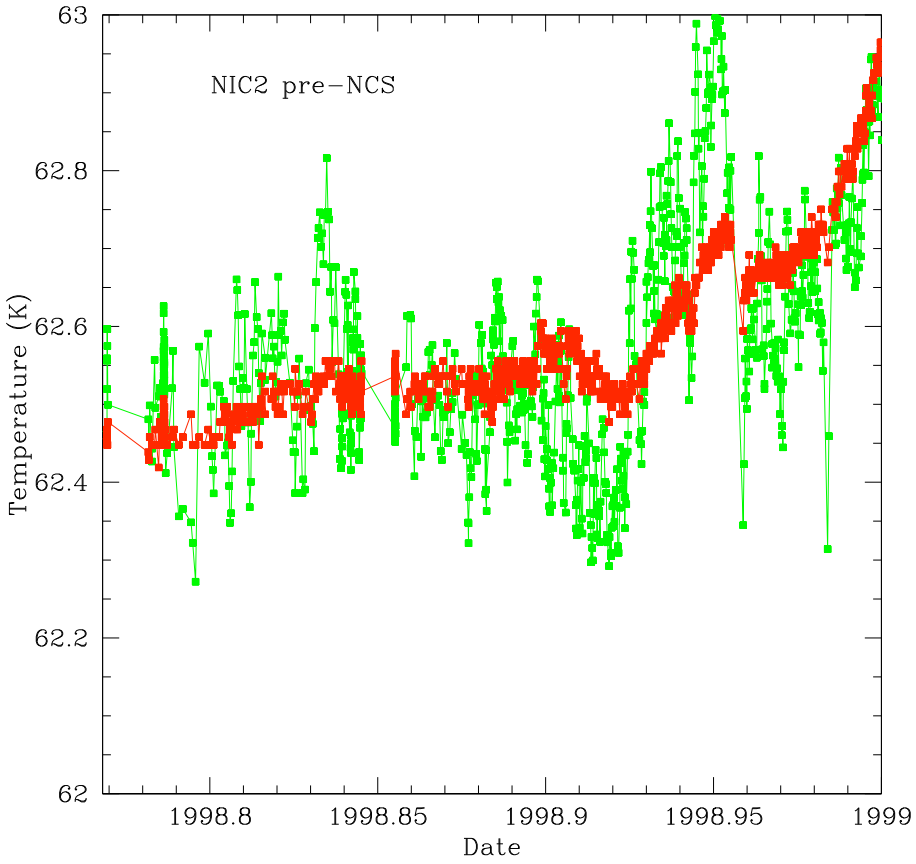


Figure 3: Mounting Cup temperature for NIC2 (red) and temperature-from-bias (green) for the pre-NCS period of 1998 to 1999.

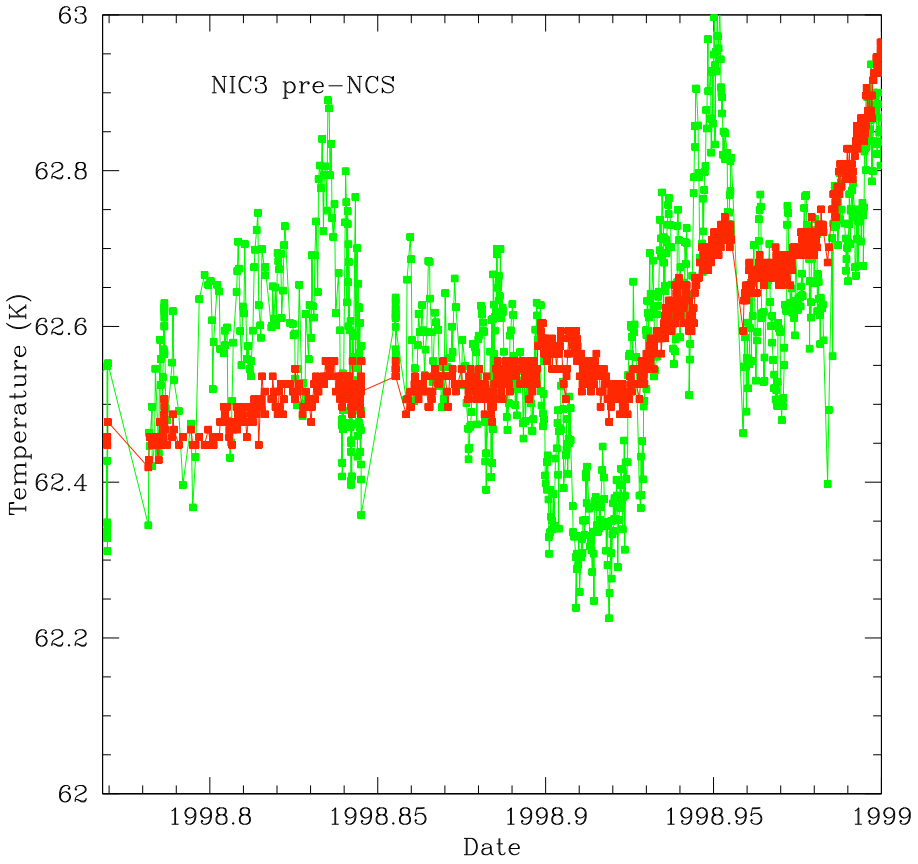


Figure 4: Mounting Cup temperature for NIC3 (red) and temperature-from-bias (green) for the pre-NCS period of 1998 to 1999.

Post-NCS comparison

Figures 5, 6, and 7 show the evolution of the mounting cup temperature as well as the temperature-from-bias since the installation of the NCS. The actual change in the temperature of each of the three detectors is clearly visible while the mounting cup temperature is being kept at a constant temperature by the NCS. This evolution, which is seen in the dark and flat-fields of NICMOS is also described in ISR 2009-002, ISR 2009-003, and Bergeron 2009, NICMOS ISR in preparation.

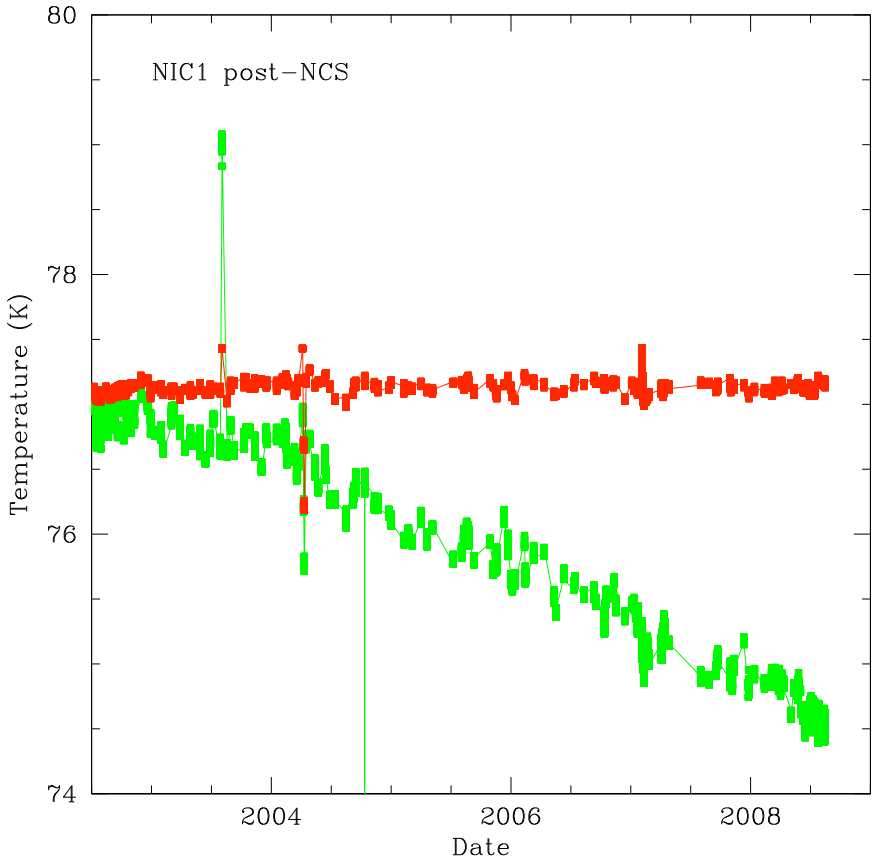


Figure 5: Mounting Cup temperature for NIC1 (red) and temperature-from-bias (green) for the post-NCS period of 2003 to 2008.

Conclusion

A new software was developed to determine the actual temperature of the NICMOS detectors, following the methodology of Bergeron 2009, NICMOS ISR in preparation. This method relies on the fact that the bias level of each detector is temperature dependent. The software, `CalTempFromBias`, has been successfully applied to all available NICMOS datasets and provides temperature estimates with uncertainties of 0.05K. `CalTempFromBias` can be easily applied to `_raw` nicmos MULTIACCUM datasets using either a command line interface, Pyraf or from within Python code.

References

Dahlen, T., Barker, E., Bergeron, E., & Smith, D., NICMOS ISR-2009-002, STScI, *Tepera-ture Dependent Dark Reference Files: Linear Dark and Amplifier Glow Components*

Dahlen, T., Sosey, M., & Bergeron, E. 2009, NICMOS ISR-2009-003, STScI, *Updates to Calnica: Using Temperature Dependent Reference Files*

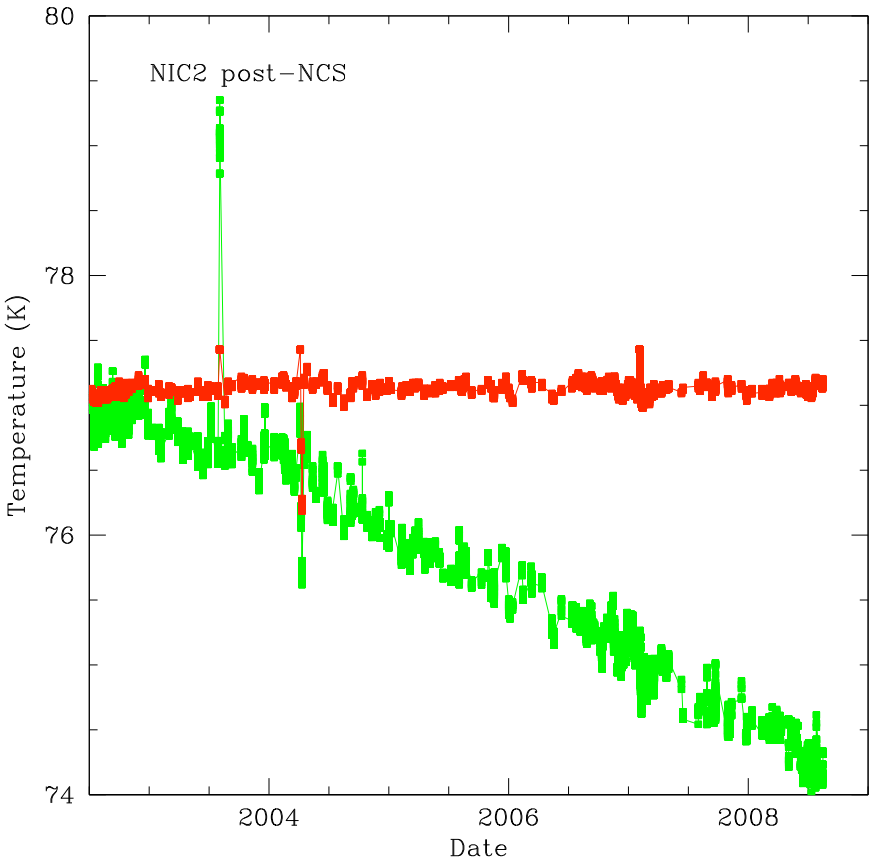


Figure 6: Mounting Cup temperature for NIC2 (red) and temperature-from-bias (green) for the post-NCS period of 2003 to 2008.

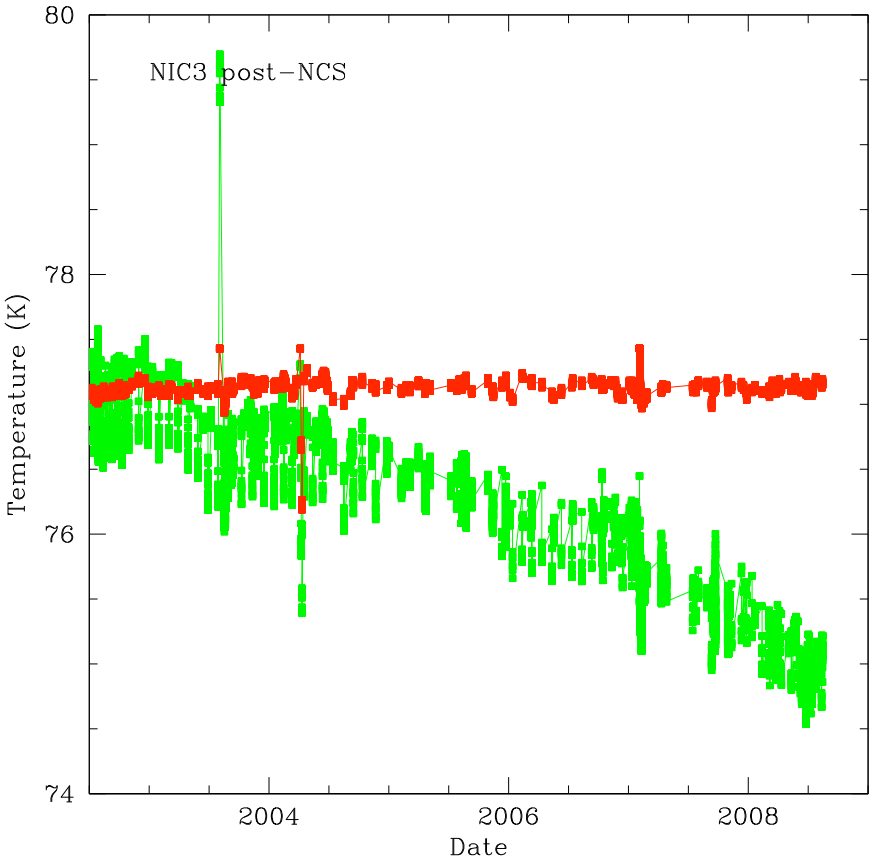


Figure 7: Mounting Cup temperature for NIC3 (red) and temperature-from-bias (green) for the post-NCS period of 2003 to 2008.