

# Removing Post-SAA Persistence in NICMOS Data

---

Elizabeth A. Barker, Victoria G. Laidler, and Anton M. Koekemoer  
May 15, 2007

---

## ABSTRACT

*A general concern with NICMOS images is the issue of ‘persistence’, or residual charge from bright sources and cosmic rays that remains trapped in pixels and is gradually released during subsequent exposures. This is particularly noticeable after HST transits of the South Atlantic Anomaly (SAA), where the cosmic ray rate is so high that their residual flux contributes a significant noise component to many subsequent exposures. We describe the ‘SAAclean’ task which can be run on post-SAA NICMOS images to identify the amount of charge in each pixel that can be attributed to persistent flux from SAA cosmic rays, and apply a correction based on an SAA persistence model to remove this flux from the data. This task has so far been tested on a variety of NICMOS data, and this represents the most recent update of this task in the PyRAF/STSDAS environment to the HST observer community.*

---

## Introduction

The data taken by the HST Near Infrared Camera and Multi-Object Spectrometer (NICMOS) is particularly susceptible to “image persistence” or latent charge. This residual charge is caused by electrons that become temporarily trapped in the NICMOS detectors. These electrons are gradually released from the detectors in an exponential decay, which produces a latent image of the charge, and can significantly reduce the signal-to-noise (S/N) in subsequent exposures (Bergeron & Dickinson 2003).

Approximately half of all HST orbits pass through the South Atlantic Anomaly (SAA), part of the Van Allen radiation belts. Passing through the SAA produces an extremely large cosmic ray hit-rate, so much so that NICMOS is powered down during the

SAA transit. Unfortunately, during this down time, charge accumulates and is trapped in the detectors, so upon powering up the instrument, the residual charge in the detectors is significant. This residual charge escapes slowly, introducing a decaying “persistence” in potentially many science exposures during the same orbit.

Clearly, we would like to recover the lost signal-to-noise in the SAA-impacted science exposures. During Cycle 7, a technique was created (Bergeron & Najita 1998) to produce an image of the persistence at the point when the persistence signal was strongest just after exiting the SAA. This “map” of the persistence could then be scaled and subtracted from the subsequent science exposures to remove the persistence and regain much of the lost signal-to-noise.

### **The SAAClean Script**

In an attempt to automate the process described above, E. Bergeron and M. Dickinson developed the SAAClean script, originally written in IDL. This script is described in detail in NICMOS Instrument Science Report 2003-010 (Bergeron & Dickinson 2003).

The SAAClean script will run through the procedure below if the time elapsed between the science image and the last SAA exit is less than 5600 sec (94 minutes, or 1 orbit) (Bergeron & Dickinson 2003). If the time elapsed is greater than this, the script exits without processing the science image. We presume that science images taken this long after the SAA exit are not affected by the SAA passage. If the time elapsed since the last SAA exit is less than 5600 sec, then the script proceeds as described below. If, after the script has run, the resultant correction to the science image is determined to be less than 1%, the script does not apply the correction and the science image remains unchanged.

### **The SAAClean Task**

The algorithm used by Bergeron’s original IDL script has been implemented in Python as a PyRAF task, now available in the `stsdas.hst_calib.nicmos` package. Some details of the numerical fitting were changed due to the use of Python/numpy/ScientificPython in place of IDL/ASTRON; additional diagnostics and task parameters were also implemented.

The Python task does not examine the elapsed time since the SAA exit, as the script does. Instead, the task checks to see if the `SAA_DARK` keyword is set to the name of the association table containing the names of the SAA dark images, if any were taken. If the keyword is not populated, the task ends without performing any calculations or changes.

It is recommended, but optional, to correct the input science image for the pedestal effect using either the **pedsky** or the **pedsub** IRAF task before processing with the SAAClean script. We believe performing the pedestal correction will improve the detection of the persistence due to the SAA passage. If a science image was impacted by the SAA passage, the `SAA_DARK` header keyword will be populated with the name of the association

table that identifies to the two post-SAA dark images correlated with the science image. Using the two post-SAA dark images, a decay scale factor is determined, and then the two images are averaged together with cosmic ray rejection. The resultant average image is then de-pedestal corrected, which provides an image of the persistence (see Figure 1).

**Figure 1:** Example of a NIC1 image before (calibrated input image), during (persistence model image) and after (SAA-persistence subtracted image) SAAclean.



This image of the persistence is then iteratively scaled by a scale factor between the two post-SAA dark images, fitting to the scale factor that provides the minimum total noise in the final output image. Then it is subtracted from the input science image. If desired, an additional and optional pedestal removal can be done at this point. We expect the results of this pedestal correction to be better than the first pedestal correction, since the persistence has now been removed from the science image.

The SAAclean task produces several intermediate and final statistics as output data (see the example output in Figure 2). This output includes the SAAclean input filename, persistence scale factor, threshold for determining the high S/N pixel regime versus the

low S/N pixel regime, the number of pixels in each regime, and best fit factors and noise values for each regime.

**Figure 2:** Example of output values determined in finding the SAA persistence.

```

saaclean version 0.99dev
Input files: n8xw12n8q_subisr.fits n8xw12n8q_subisr.fits
sci image : using DQ extension for badpix
postsaa dark #1 : using DQ extension for badpix
failing over to /data/cdbs5/nref//m9c1047pn_msk.fits
postsaa dark #2 : using DQ extension for badpix
failing over to /data/cdbs5/nref//m9c1047pn_msk.fits
Using scale factor of 0.54 to construct persistence image
flatfile : using DQ extension for badpix
median used in flatfielding: 0.0379231178101

Coefficients for gauss-poly fit to persistence model histogram:
Gaussian (low signal component) terms:
  Amplitude, Mean, Sigma: 1026.073392 17.503406 9.216971
Polynomial terms:
  Constant, Linear, Quadratic: 47.687849 0.422538 -0.003921

Threshold for hi/lo: 0.0995250821636
Npixels hi/lo: 9596 55940

Results summary for high domain:
chi2 for parabola fit = 0.000420518773132
  min-noise (best) scale factor is: 0.311645043762
  effective noise at this factor (electrons at gain 5.400000): 49.903919
  noise reduction (percent) : 38.5484443442

Results summary for low domain:
chi2 for parabola fit = 4.05079716407e-05
  min-noise (best) scale factor is: 0.189995821621
  effective noise at this factor (electrons at gain 5.400000): 36.192606
  noise reduction (percent) : 7.21719319318

Applying noise reduction in both domains

```

During our testing we found it informative to plot a histogram of the flat-fielded persistence image used in the determination of the high- and low-domain pixels. This image is produced when the `alldiags` parameter is set to ‘yes’. This allows the user to do a quick, visual sanity check on the calculated mean and threshold determined by the task (see Figures 4, 5 and 6). By default, the threshold delineating the two pixel domains is defined to be 3.5 sigma above the histogram mean (Bergeron & Dickinson 2003). The user can visually verify that the calculated threshold does in fact reasonably represent the division between high signal pixels and low signal pixels. If it does not, the user can determine a more appropriate threshold by hand and set the `fitthresh` parameter to “no” and the `thresh` parameter to the desired value. The task can then be rerun to produce a persistence-corrected image based on the input `thresh` value.

As noted in the original **SAAclean** ISR (Bergeron & Dickinson 2003), any correction determined by the **SAAclean** script was applied to the input science exposure. In the current task, the user has the option to apply the correction to the pedestal-corrected image or to another image, usually the original `*_cal.fits` file. Since the persistence is decaying with time, the user will want to ensure that the time since exiting the SAA passage is

the same for both of these images, if different images are selected. The examples in this document were produced from a case for which the SAA correction was applied to the pedestal-corrected (**pedsub**) image.

The **SAAclean** task also requires a dark file for dark-subtracting the post-SAA darks before they are used to compute the model of the persistence. These dark files are available on the NICMOS Reference File web site ([http://www.stsci.edu/hst/nicmos/tools/post\\_SAA\\_tools.html](http://www.stsci.edu/hst/nicmos/tools/post_SAA_tools.html)) or from the STScI CDBS web site (<http://www.stsci.edu/instruments/observatory/cdb/>) for each camera for the Post-NCS era. If the **SAADFILE** keyword is not populated with the complete path for these reference files, then the `saaref$` variable must be set to the directory containing them.

### *Using the SAAclean PyRAF Task*

In order to run **SAAclean**, the input image must be flat-fielded and have the **SAA\_DARK** header keyword populated with the name of the association table which contains the names of the post-SAA dark exposures taken immediately before the science exposure. The user must specify the input image filename, on which the computation will be performed; the target image filename, to which the correction will be applied; the output image filename, to which the result will be saved; and the clobber parameter, which will determine whether or not any existing output files should be overwritten. The remaining parameters are optional and have the default values shown in Figure 3. The values of these parameters can be changed by the user, but we recommend the default values for the initial run.

**Figure 3:** Parameter list and the default settings for the **SAAclean** task.

<code>calcimage =</code>	Input (usually ped) file to calculate correction
<code>targimage =</code>	Input (ped or cal) file to correct
<code>output =</code>	Output cleaned datafile (targimage with correction applied)
<code>clobber = no</code>	Overwrite output files if they already exist?
<code>(readsaaaper = no)</code>	Read SAA persistence image from file (If no, construct it.)
<code>saaperfile = saaper.fits</code>	Filename for SAA persistence image
<code>(writesaaaper = yes)</code>	Write SAA persistence image?
<code>(flatsaaaper = yes)</code>	Flat-field SAA persistence image before analysis
<code>(darkpath = saaref\$)</code>	Path to dark reference files
<code>(scale = 0.54)</code>	Scale factor for constructing persistence image
<code>(wf1 = 0.7)</code>	Weight for first SAA exposure
<code>(wf2 = 0.3)</code>	Weight for second SAA exposure
<code>(crthresh = 0.3)</code>	Threshold for CR rejection
<code>(noisethresh = 1.0)</code>	Noise reduction threshold (percent)
<code>(binsigfrac = 0.3)</code>	Stddev fraction for excluding narrow bins
<code>(stepsize = 0.008)</code>	Increment multiplier for SAA scale factor fitting
<code>(fitthresh = yes)</code>	Solve for threshold value? (If no, uses value of thresh)
<code>(thresh = )</code>	Threshold to separate high/low signal domain in SAA persistence image
<code>(histbinwidth = 0.001)</code>	Bin width for histogram in threshold fitting
<code>(nclip = 3)</code>	Number of clipping iterations for threshold fitting
<code>(hirange = 0.4)</code>	Maximum multiplier for high signal domain
<code>(lorange = 0.25)</code>	Maximum multiplier for low signal domain
<code>(fitmult = yes)</code>	Fit to determine multiplier for minimum noise?
<code>(applied = )</code>	Cleaning applied to which domains?
<code>(hi_nr = )</code>	Noise reduction in high signal domain
<code>(lo_nr = )</code>	Noise reduction in low signal domain
<code>alldiags = no</code>	Write out all possible diagnostic files?
<code>(diagroot = diag)</code>	Root filename for diagnostic files

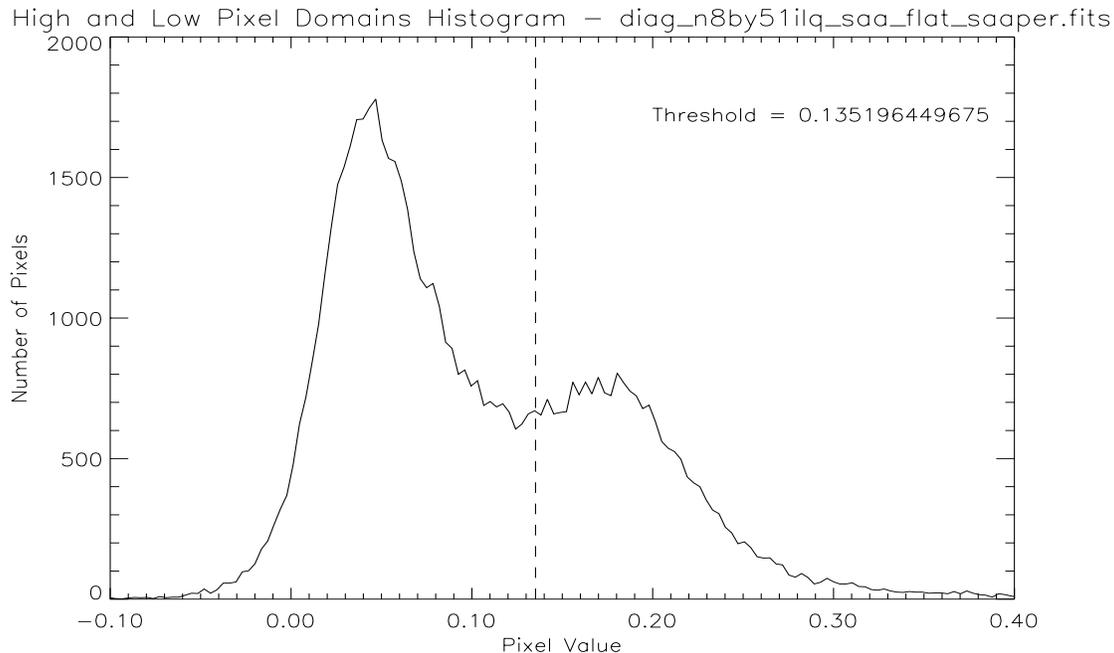
Users can find all the necessary information for running the task in the task help file. We recommend using the pedestal-corrected image as the input image for computing the ideal SAA persistence cleaning, but the correction can be applied to either the `*_cal.fits` file or the pedestal-corrected file, whichever the user feels is most appropriate to achieve the desired scientific result. We also recommend performing a final pedestal correction on the output image to remove any residual pedestal.

The SAA persistence is, by default, determined using the post-SAA dark exposures indicated in the association table given by the `SAA_DARK` keyword. Normally, the best files to use for this are the post-SAA darks taken immediately after the SAA exit during which the science exposures were taken. These SAA darks are delivered with the science exposures by the archive when the “Best Calibration Files” are requested. However, the user may specify a different set of dark exposures in the `SAA_DARK` keyword if desired.

In order to make the best compromise between correcting for as much persistence as possible and not unduly increasing the noise in the final image, high- and low-signal persistence pixels are treated separately (see Bergeron & Dickinson 2003). This is done by fitting a combination of a Gaussian and a polynomial to the persistent signal. A good fit to the histogram is critical to the determination of the high- and low-pixel populations, since the threshold is calculated as 3.5 times the standard deviation of the fitted Gaussian component. Separating the two populations properly produces a much better fit in scaling the persistence to achieve the best noise reduction as described below. If an initial determination produces poor results, the histogram of the output persistence image (produced when the `alldiags` parameter is set to ‘yes’), with the threshold, can be plotted and examined. If an anomalous threshold was found by the algorithm, the user has the flexibility to enter a value for the high/low domain threshold and reprocess the image. We found that the algorithm occasionally produces some questionable threshold values, when

objects are heavily saturated, prominent diffraction spikes are present, the entire array is filled with high-signal domain pixels, or other anomalous features are present.

**Figure 4:** Image histogram showing the high- and low-signal domains and the threshold defining the two regimes in the persistence image.

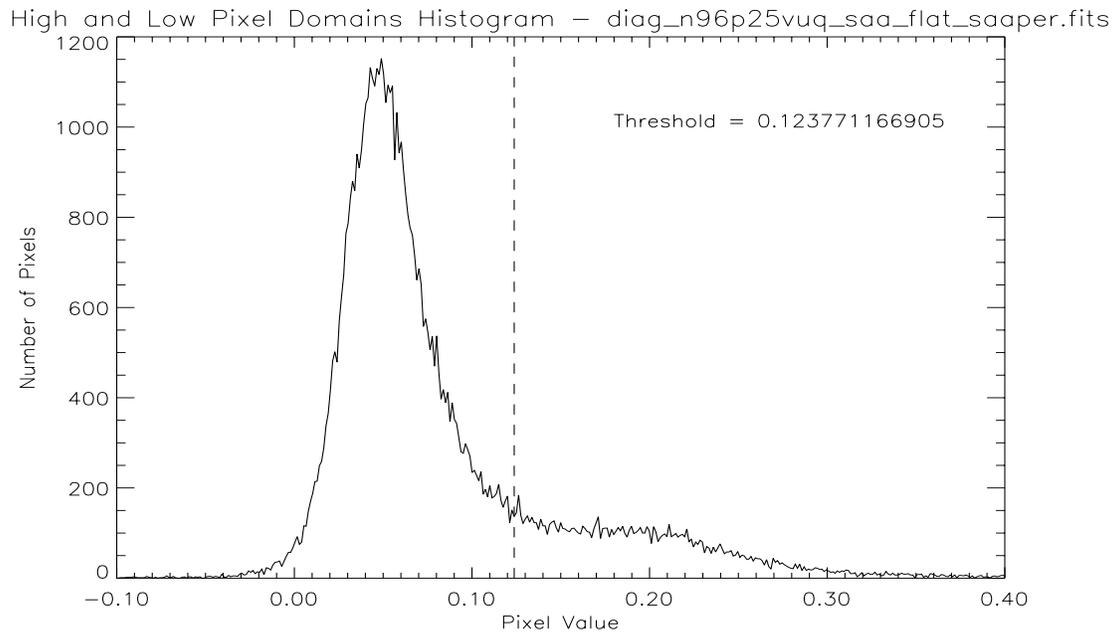


Once the high- and low-pixel domains have been determined, the noise in each domain is estimated separately. For each pixel domain, the noise reduction estimate must exceed the user-specified `noisethresh` value in order for the correction to be applied to pixels in that domain. Note that if the low-pixel domain exceeds the `noisethresh` value but the high-pixel domain does not, then the low domain correction is applied to the entire image.

Once the task has produced an output image and a correction was applied, the header keyword `SCNAPPLD` will indicate this with one of the following values: `both`, `low everywhere`, `high only`. The value will be `none` when the calculation is below the value of the `noisethresh` parameter, indicating that no correction was warranted.

The n/a value will indicate that the task determined that the input image was not SAA impacted, and then exited gracefully.

**Figure 5:** Another example of a histogram showing the two pixel domains.

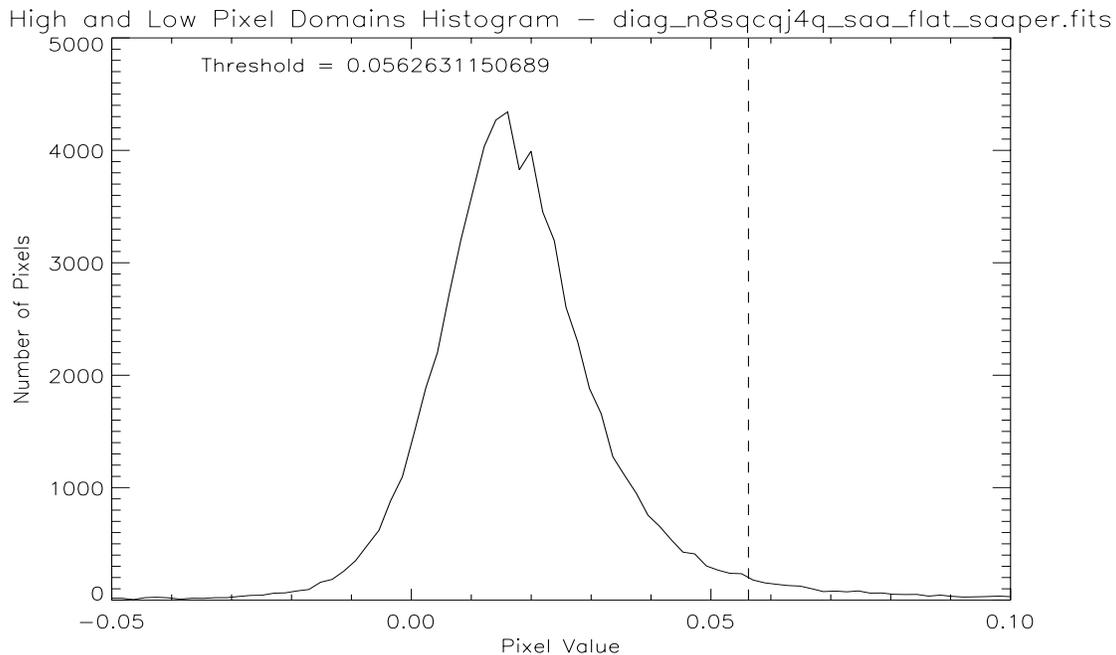


If the task encountered any difficulty during the calculations, it will populate the SCNAPPLD keyword with the value `abort` and then stop and exit with an informative error message. This can happen when an anomalous threshold was calculated, when the fitting process for the best scale factor has failed, when the task has already performed a correction on the `targimage`, or when a linear algebra error was encountered during the Gaussian-polynomial fitting process. In this way, the keyword is intended as an indicator to the user. The user can then reset the value of this keyword to blank if reprocessing the data is desired.

After the task has completed the calculations, many of the values determined throughout the process are written to the standard output (see Figure 2). The relevant keywords are populated with the final calculations and values, based on the input settings. These keywords are listed in Table 1 and can be found in Chapter 2 of the NICMOS Data Hand-

book (available on the NICMOS web site at <http://www.stsci.edu/hst/nicmos/documents/handbooks>).

**Figure 6:** Persistence image histogram and the threshold separating the pixel domains.



The persistence model image is output, along with the screen text, which lists the values calculated at various points through the task. The user can examine this image and these values to get some insight into the quality of the results and which parameters might be modified if better results are desired. Extremely high or low  $\chi^2$  values are an indication of a poor persistence model fit. If the number of high-domain pixels is more than the number of low domain pixels, or if there are fewer than a few thousand pixels in either one of the domains, one should investigate the calculated threshold by plotting it against the flat-fielded persistence image histogram (produced when the `alldiags` parameter is set to 'yes'). The threshold should fall between the two domains in the histogram (Figures 4, 5, and 6 show typical threshold value determinations). If necessary, the `thresh` parameter can be used to set to a visual estimate of the threshold, the `fitthresh` parameter set to "no" and the task rerun on the exposure.

**Table 1.** New **SAAclean** header keywords and descriptions.

<b>Keyword</b>	<b>Description</b>
SAAPERS	SAA persistence image
SCNPSCCL	Scale factor used to construct persistence image
SCNPMMDN	Median used in flatfielding persistence image
SCNTHRSH	Threshold dividing high and low signal domains
SCNHNPIX	Number of pixels in high signal domain (HSD)
SCNLNPIX	Number of pixels in low signal domain (LSD)
SCNHCHI2	HSD chi squared for parabola fit
SCNHSCCL	HSD scale factor for min noise
SCNHEFFN	HSD effective noise at SCNGAIN
SCNHNRED	HSD noise reduction (percent)
SCNLCHI2	LSD chi squared for parabola fit
SCNLSCL	LSD scale factor for min noise
SCNLEFFN	LSD effective noise at SCNGAIN
SCNLNRED	LSD noise reduction (percent)
SCNAPPLD	To which domain(s) was SAA cleaning applied
SAACNTAB	Reference table (with task parameters)
SAACPDGR	Pedigree of reference table
SAADFILE	SAA dark reference image file
SAADPDGR	Pedigree of reference image
SAACORR	Correct for SAA signature
SAADONE	Status of SAA signature correction

## Testing the SAAclean Task

We have tested the **SAAclean** task on a vast sample of existing NICMOS data (8840 images), utilizing all three cameras (NIC1, NIC2, NIC3). Specifically, we ran it on 766, 3013 and 5061 images for NIC1, NIC2 and NIC3, respectively (see Table 2). It appears that the data from NIC1 is the least impacted by SAA-induced persistence (least amount of correction required, if needed at all), and NIC3 the most. We determined this by examining the number of exposures utilizing the **SAAclean** correction and evaluating how strongly these exposures are affected.

### *Testing Data Sample*

We included data from each camera and all filters for which data existed in the archive at the time the testing began (see Table 2). We did not test any images that were not flat-fielded, which excluded all grism images. We required a pedestal correction to be done before running **SAAclean**, which excluded some data sets that did not produce successfully pedestal-corrected images. Our testing exclusively used post-NCS era data, since only these datasets have post-SAA darks delivered with the science exposures.

**Table 2.** Camera/filter combinations and total number of observations used in the testing.

	NIC1	NIC2	NIC3
F090M	F110W	F108N	
F095N	F160W	F110W	
F108N	F165M	F113N	
F110M	F171M	F160W	
F110W	F180M	F164N	
F145M	F187N	F166N	
F160W	F187W	F187N	
F164N	F190N	F190N	
F165M	F204M	F196N	
F166N	F205W	F200N	
F170M	F212N	F212N	
F187N	F215N	F215N	
F190N	F222M	F222M	
POL0S	F237M	F240M	
POL120S	-	-	
POL240S	-	-	
<b>Total # of Images:</b>	<b>766</b>	<b>3013</b>	<b>5061</b>

This sample of data included many different objects. Planets, stellar clusters, galaxies, sparse fields, bright extended objects, bright point sources, faint point sources, and faint extended sources were all included.

### *SAA Dark Reference Files*

Before the persistence model can be determined, **SAAclean** must calibrate both SAA dark files through dark-subtraction. This is done with suitable dark files, one for each camera (available on the NICMOS web site at [http://www.stsci.edu/hst/nicmos/tools/post\\_SAA\\_tools.html](http://www.stsci.edu/hst/nicmos/tools/post_SAA_tools.html) and the STScI CDBS web site at <http://www.stsci.edu/instruments/observatory/cdbs/>).

### *Population of Keywords*

In order for **SAAclean** to run, several header keywords must be populated in the input science exposure header. The `FLATFILE` keyword must be populated with the name of the corresponding flat field image filename. The `SAA_DARK` keyword must be populated with the name of the association table containing the names of SAA dark images corresponding to the science exposure. These keywords are typically populated automatically by the OTFR processing when the data is requested from the archive.

### *Functionality Modifications*

We have made slight modifications the **SAAclean** task since its original release in an attempt to improve functionality over a broad range of data types and to make it slightly more user friendly. We have modified the original script behavior from using an input parameter to determine the gain of the science image to now extracting the gain from the

science image header. Several new parameters have been added, allowing the user to have more complete control over the histogram generation used for determining the threshold for the high- and low-pixel domains (see Figures 4, 5 and 6).

During our testing we discovered many poorly corrected exposures may have been related to the non-existence of a separate “bad pixel image”. We believe these problem cases may have been caused by no bad pixel image being found. The original IDL task used a bad pixel mask to ignore bad pixels throughout the **SAAclean** processing. We have removed this dependency by using the science exposure data quality image ([DQ] extension) to achieve the same result.

The first stand-alone release of the PyRAF implementation of the task used a faulty method for determining the histogram for determining the high- and low-pixel populations, which has now been corrected.

### ***Pedestal Correction Selection***

We tested the robustness of the two available pedestal-correcting tasks, since the output of one of these would be required as the pipeline input to **SAAclean**. These two tasks, **pedsky** and **pedsub**, are currently available to the NICMOS community in IRAF/STSDAS. We ran an exhaustive set of NICMOS data through both tasks and compared the output pedestal corrections by the following method: We divided each output pedestal-corrected image by the calibrated input image and subtracted the one pedestal-corrected output from the other, for each original input image. We determined that **pedsub** provided better and more reliable results, without regard to the nature of the input exposure.

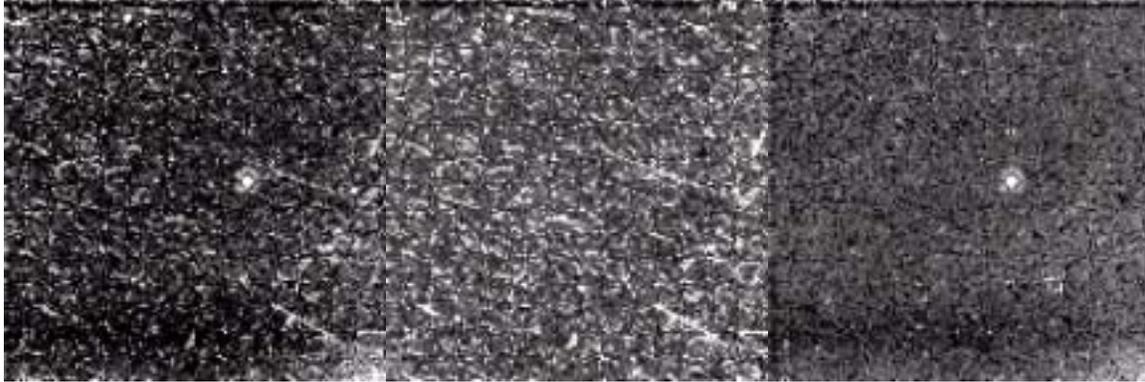
### ***Representative Examples***

#### *Good example*

Through our testing we found that NIC1 was far less impacted by SAA persistence, and therefore required far less dramatic correction, when necessary. Here we present an exam-

ple of a correction done on a NIC1 image of the 2MASS object WJ152322+3014 in the F108N filter.

**Figure 7:** A good example of when **SAAclean** works well, zoomed in to show detail. From left to right, before (calibrated input image), during (persistence model image) and after (SAA-persistence subtracted image) running the **SAAclean** task.



### *Problem Cases*

During our testing, we discovered that some images produce anomalous results when processed through **SAAclean**. Images containing heavily saturated objects (especially with diffraction spikes) and images of Jupiter did not produce reliable results. The images have statistics that are not handled well when the best Gaussian-polynomial fit is attempted. We recommend caution when trying to use **SAAclean** on such observations.

## **Conclusion**

NICMOS images are strongly impacted by SAA passages through latent charge remaining in the detectors, which causes persistence in subsequent exposures. We have tested a new PyRAF task, based on the **SAAclean** script developed by Bergeron and Dickinson, which can remove much of the SAA-induced persistence and regain otherwise lost signal.

After testing the new **SAAclean** task on essentially all existing NICMOS archived data, we see that in most cases, **SAAclean** improves the signal-to-noise of objects of interest and produces good results. The large number of task parameters provides the user tremendous flexibility in tailoring the results to achieve the best possible results for his or her data.

Work is underway to add **SAAclean** functionality to the standard calibration pipeline for NICMOS by wrapping it in a pipeline-compliant script. This script would be performed after **calnica** and before **calnicb**. We expect the pipeline version of the **SAAclean** task to be available to the community in the future.

We encourage users to send questions or comments to the STScI Help Desk at [help@stsci.edu](mailto:help@stsci.edu).

## **Acknowledgements**

We would also like to thank Eddie Bergeron for helpful discussions throughout the conversion process.

## **References**

- Bergeron, L. E. & Dickinson, M. E. 2003, NICMOS ISR 2003-010, STScI  
Bergeron, L. E. & Najita, J. R. 1998, BAAS, 30, 1299