

Removal of Cosmic Ray Persistence From Science Data using the Post-SAA Darks

Louis E. Bergeron and Mark E. Dickinson
September 25, 2003

Abstract

Latent or “persistent” images of cosmic ray hits following HST transits of the South Atlantic Anomaly (SAA) can significantly increase the noise in subsequent NICMOS science images. By taking a pair of DARK exposures immediately following the exit of the SAA, a map of the persistent signal can be made and then used to subtract this signal from the impacted images in that orbit, thereby recovering much of the original S/N. We here describe an algorithm to do this.

Introduction

Latent charge or “image persistence” is caused by photon or charged-particle produced electrons which become temporarily trapped in flaws in the crystalline material of the NICMOS HgCdTe and CdTe bulk material, as well as at the surfaces and interfaces of these materials. These trapped electrons can not make their way through the depletion region to be detected in a particular pixel in the same timescale as all of the other (mostly desirable) radiation-generated electrons, which show up as signal in a normal exposure. Instead, these electrons are eventually released in a logarithmic fashion over time, due primarily to thermal excitations which knock the electrons loose and allow them to be detected at the pixels. The distribution of trapped electrons is proportional to the intensity, spatial distribution of, and time since a given radiation source filled those traps, and the physical distribution and number-density of the traps themselves in the device. As a result, some memory of previous exposures is observed in subsequent images, whether they be faint images of bright stars, or random salty blips from the many cosmic-ray hits the detectors receive in low

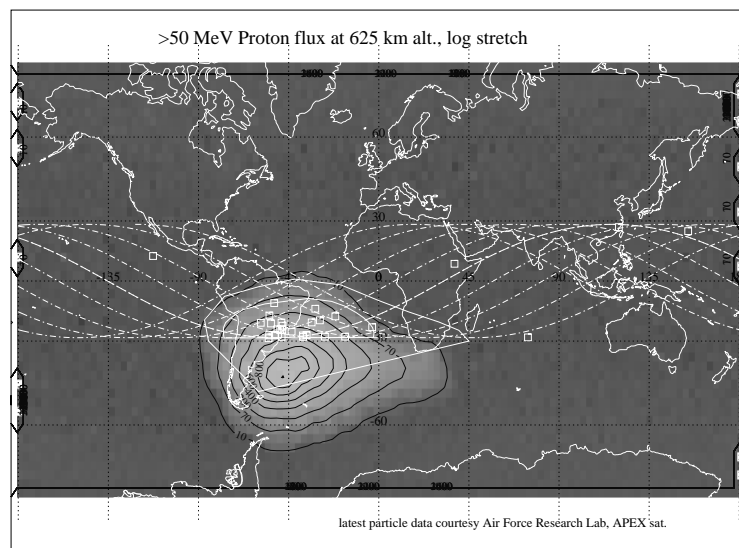


Figure 1. Nine sequential HST orbital ground tracks through the SAA. The underlying image shows the relative particle intensity, with black contours. The white contours are the HST operational SAA contours 05 (outer) and 02 (inner). Square boxes are a few SEU locations, shown here for reference only. (Bergeron and Najita, 1998)

Earth orbit. These latent images can seriously degrade the signal-to-noise (S/N) in subsequent exposures and can corrupt photometric measurements of faint sources.

A particularly nasty manifestation of this trap-and-release phenomenon occurs when HST passes through a low dip in the Van Allen radiation belts over the Atlantic ocean called the South Atlantic Anomaly (SAA). The charged particle density increases dramatically in this region, and HST makes several (8-9) passes through it each day - about half of all HST orbits (figure 1). The total dosage of cosmic rays depends on how deep a passage is, and how much time is spent in the traverse. The practical hit-rate is so large in fact, that most HST science activities are suspended while in the SAA. NICMOS is powered

down during this time to prevent Single Event Upsets (SEUs) in the opto-isolators from accidentally triggering bit flips that lead to undesirable actions in the electronics. However, the traps are still there and they still grab unwary electrons from all those cosmic-ray hits (which in a deep SAA passage can hit every pixel on the detector several times). The result of all this is that when HST emerges from the SAA and the NICMOS is powered back up and begins taking exposures again, those electrons start to drop out of the traps and show up in the images in the random, but spatially structured patterns in which the cosmic rays had populated them. The spatial distribution of the CR hits leaves a random-looking noise pattern of varying intensity all over the image. Since the rate that the trapped electrons escape from the traps is small and decaying exponentially with time (figure 2), short exposures are relatively immune to the effect, as there is not enough persistent signal to add to the total noise. Long exposures however allow the persistent signal to accumulate to a point where it becomes the dominant noise source in the image (figure 3). This “noise” is in fact worse than a pure white noise signal from the point-of-view of its impact on science data, since it has spatial coherence, is non-Gaussian and persists from one image to the next. Of course one could just wait until the traps are depleted before starting an exposure, but with timescales of minutes to hours HST has already gone around the earth and is entering the SAA again before the persistent signal has decayed enough to make this practicable. The only option would be to not take data in SAA-impacted orbits. Since half of all HST orbits are impacted to some degree or another, this would incur a huge efficiency hit on HST science operations.

In Cycle-7, a technique was developed (Bergeron and Najita, 1998) whereby a pair of images with the BLANK filter in place (i.e. dark exposures) are taken to make an image of the persistent cosmic rays at the point where the persistent signal was strongest just after exiting the SAA. This CR-persistence map could then be multiplied by some scale factor (less than 1.0, assuming everything is normalized to counts per second) and subtracted from the subsequent science data to remove the persistence and nearly recover the original S/N. In Cycle 11 (following

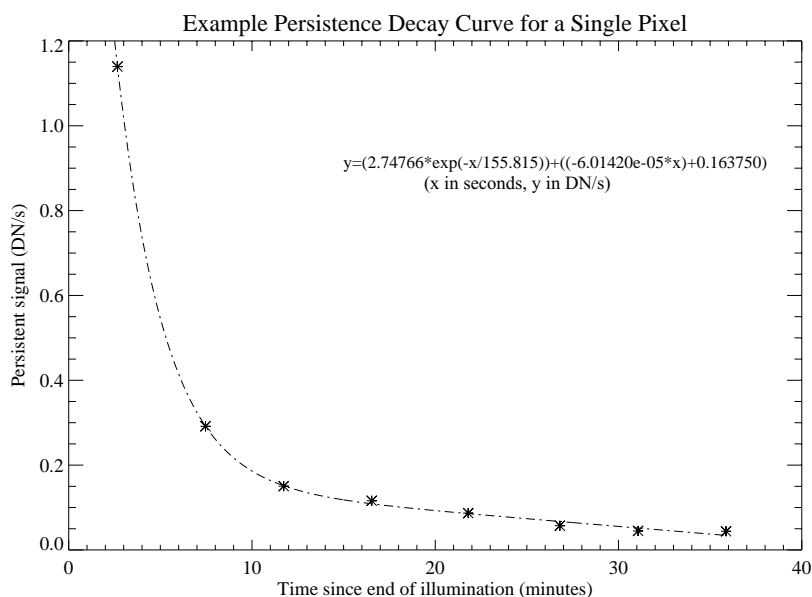


Figure 2. Example of a persistence decay curve after a very bright illumination.

the installation of the NCS cryocooler in Servicing mission 3B), a scheduling change was implemented that automatically takes these post-SAA darks upon exit from the SAA whenever NICMOS exposures are to be taken during that orbit. This document describes an algorithm by which these post-SAA darks can be used to correct impacted NICMOS science data.

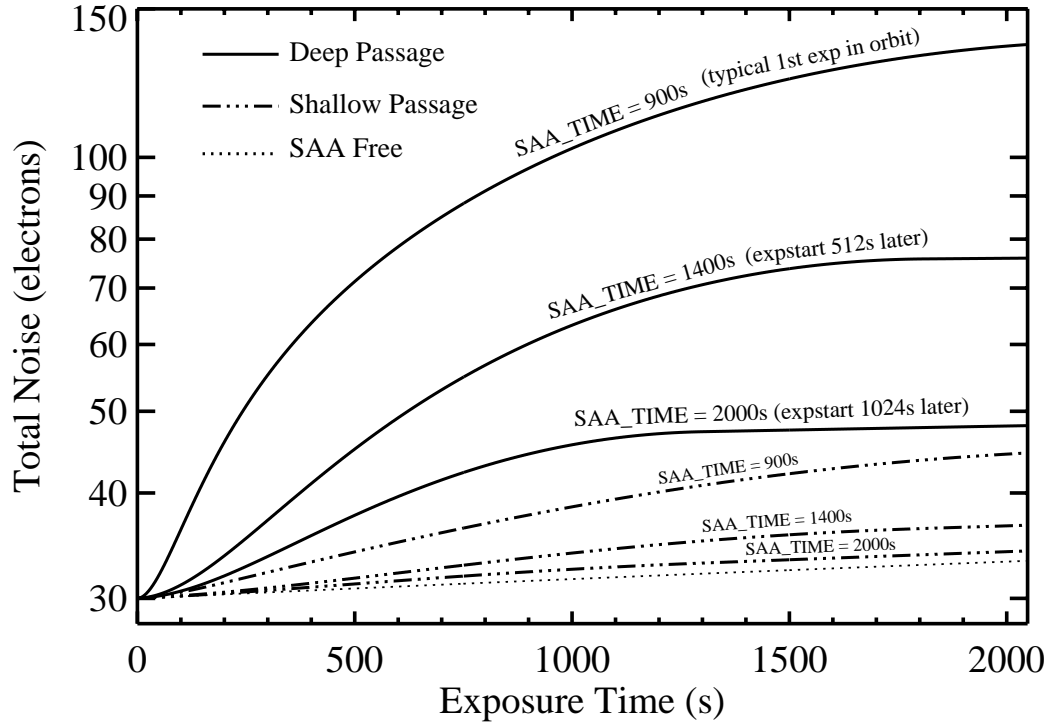


Figure 3. Noise induced by cosmic-ray persistence following an SAA passage as a function of time since SAA exit and exposure time. Curves are shown for both deep and shallow passes. The “SAA Free” curve is read noise + shot noise from dark current at 77.1 K.

The Algorithm

The following outline describes the steps involved in removing post-SAA cosmic-ray persistence from an image. Each of these steps is described in detail in subsequent sections of this document. Optional (but recommended) steps are shown in *italics*.

- First determine if the image (a calibrated NICMOS image in DN/s) was taken during an SAA-impacted orbit by checking the header keyword SAA_DARK and considering your exposure times. If so, make sure you have both the *_asn.fits table that this keyword points to, as well as the two post-SAA dark images that are called out in this table.
- *Pedestal correct your calibrated science image to aid in the correction process.*
- Read the two post-SAA dark images, determine the scale factor of the decay between them (or use the default), average the two together with some cosmic-ray rejection, and de-pedestal to make an image of the persistence.
- Iteratively scale and subtract the persistence image from your science image, fitting for the scale factor that gives the minimum total noise in the final output image.
- *If desired, repeat the pedestal removal which should work better in the absence of the persistent signal.*

1) Does Your Science Image Have Cosmic Ray Persistence in it?

Before attempting to correct any dataset for post-SAA persistence, it is useful to check a few things to make sure it is indeed worth correcting. There is a group of header keywords in all NICMOS science headers called the *POST-SAA DARK KEYWORDS*:

```
SAA_EXIT= '2003.054:11:09:17' / time of last exit from SAA contour level 23
SAA_TIME=          916 / seconds since last exit from SAA contour 23
SAA_DARK= 'N6273I020' / association name for post-SAA dark exposures
SAACRMAP= 'n6273i020_saa.fits' / SAA cosmic ray map file
```

If the SAA_DARK keyword is “N/A” then this data was not taken in an SAA-impacted orbit and no correction is necessary. In this case, the SAACRMAP keyword will also be “N/A,” and the SAA_TIME keyword will usually have a value larger than 5600s (94 minutes, or 1 orbit). If the SAA-DARK keyword contains a filename, this is the name of the association table (a fits table) which lists the rootnames of the two post-SAA dark images that were taken at the beginning of the orbit. These two images will be the ones used to generate the image of the SAA cosmic ray map file (SAACRMAP). The SAACRMAP keyword is currently just a placeholder, but this will point to the image containing the CR-persistence map that will be automatically generated as part of the pipeline processing when the algorithm described in this document is implemented. The table called out in the SAA_DARK keyword, and the two post-SAA dark images listed in it will be automatically extracted from the HST Data Archive with your science data as part of your retrieval request. If not, you can request them explicitly from the archive using STARVIEW.

Another way to check where your science exposures took place relative to the SAA crossings is to use the web-based SAA Crossing Calculator tool on the NICMOS website:

<http://stdas.stsci.edu/hst/nicmos>

(follow the “Software Tools” link in the task bar on the left, then click on “SAA Crossing Calculator”)

Just enter the value(s) of the EXP-START keyword from your science image headers and it will generate a plot showing the family of SAA passages (typically 8/day) along with a vertical line for each of your observations (figure 4). The width of the vertical black bars indicates the time HST spent in the SAA during each of the passes. If your observation occurs immediately after (to the right of) one of the black bars on the plot, then it is probably impacted. If its more than the spacing between the bars to the right, then the observation was

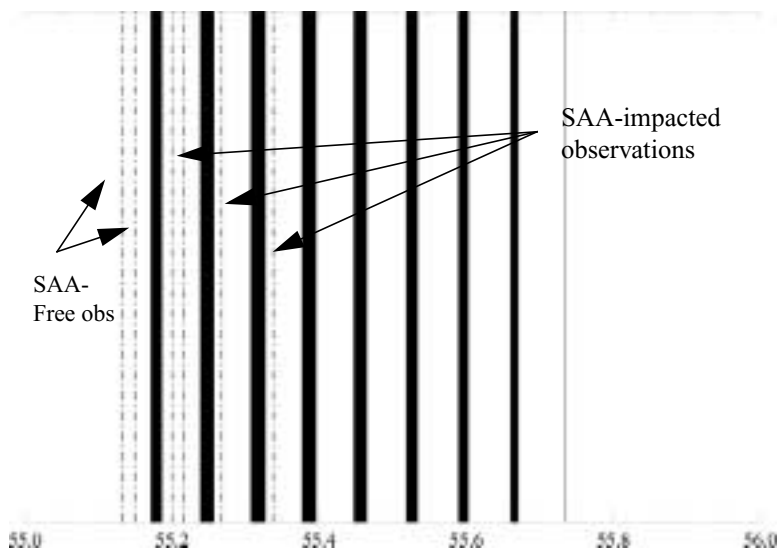


Figure 4. Example output from the SAA Crossing Calculator tool. The horizontal axis is time (days). Vertical black bars are the times when HST is traversing the SAA. Vertical dashed lines are the EXPSTART times of science exposures

obtained in an SAA-free orbit. Generally the first and last SAA crossings of each day are only mildly affected.

One last check is to simply look at the images and measure a few statistics. If there are some parts of the image that are free of astronomical and background sources, you can measure the sigma in the sky background. A typical low-background MULTIACCUM sequence results in a `_cal.fits` calibrated science image with a 25-30e- total noise. An equivalent SAA-impacted exposure can have up to 150e- of noise (figure 3). Impacted images have a “mottled” or “salty” appearance, and often it is difficult to even see sources (figure 5). The mean-value of the persistent signal will also tend to show up as an elevated “sky” background in images that shouldn’t normally have much background, but it is the noise that really stands out to the eye.

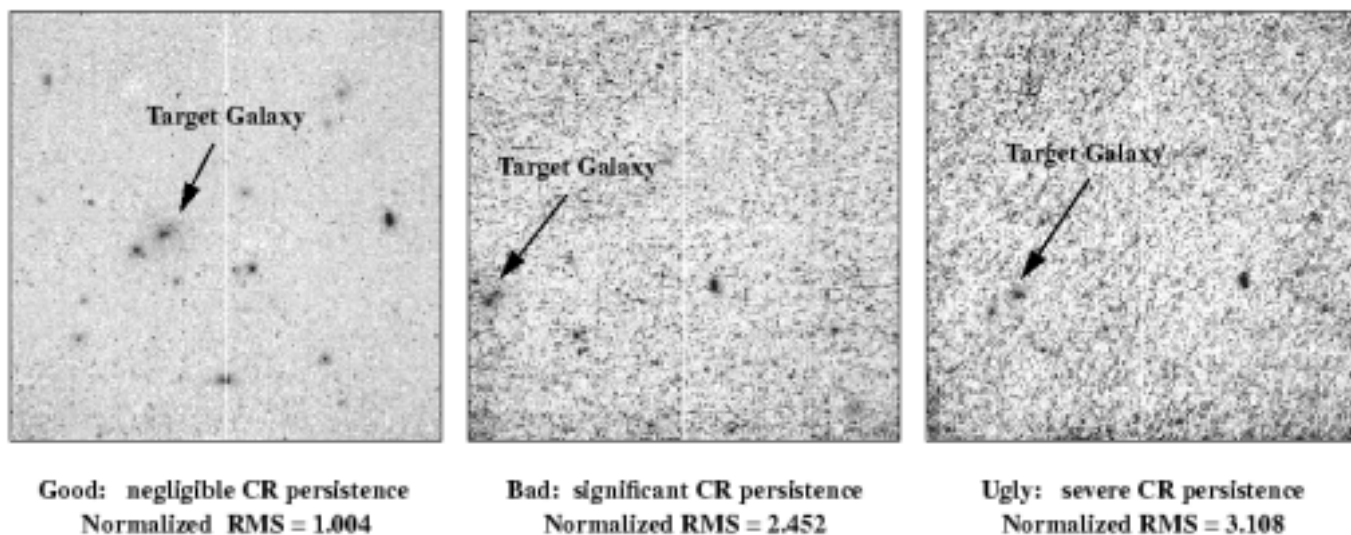


Figure 5. Examples of varying degrees of post-SAA cosmic ray persistence in a science image.

2) Pedestal Correction

After calibration with the standard NICMOS calibration pipeline, nearly all science images still contain some residual “pedestal.” The pedestal is a time-variable quadrant bias, usually constant across a quad but different from quad-to-quad, which is electronically induced. It is mostly likely due to the very lowest frequency of the $1/f$ (Johnson) noise in the output FET spectrum, although there could be other sources. For a more thorough discussion of the effect, see section 4.1.2 of the HST Data Handbook (<http://www.stsci.edu/hst/nicmos/documents/handbooks/>). Removal of any pedestal in the science image before running the post-SAA persistence removal algorithm will usually improve the noise-rejection performance, although it is not strictly necessary. Also,

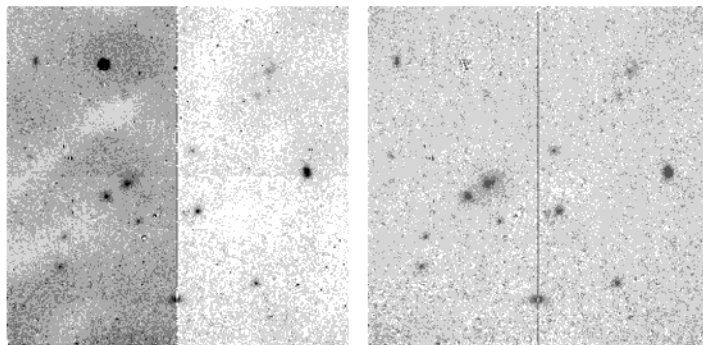


Figure 6. Data affected by variable quadrant bias. Left: image processed normally with `calnica`; note the quadrant intensity offsets, and also the residual flat field pattern imprinted on the data, due to the unremoved bias being multiplied by the inverse flat. Right: image after processing through `pedsky`.

re-running the pedestal removal after post-SAA persistence correction is useful as well (figure 6). Most pedestal-removers work by iteratively subtracting quad-mean biases and trying to minimize the overall noise. Persistent signal can limit the sensitivity of such algorithms. The STSDAS routine **pedsky** in the `hst_calib.nicmos` package is fast, effective and easy to use. We recommend running it on science images both before and after post-SAA correction. See the **pedsky** help file in STSDAS for more details.

SAA_CLEAN.PRO

The following two sections describe steps that are performed by an IDL routine called `saa_clean.pro`. A complete description of this IDL code and the code itself are described in Appendix A. `saa_clean.pro` is being converted into a stand-alone STSDAS task, which might eventually be added to the **calnica** calibration pipeline.

3) Make an Image of the Persistence (the SAA Cosmic Ray Map)

Once you know you have SAA-impacted data, the next step is to make an image of the cosmic ray persistence using the two post-SAA dark images. Taking as an example the image used in section 1, we see that the keyword `SAA_DARK= 'N6273I020'`. Using the stsdas task **tprint**, have a look at the contents of that fits table:

```
ni> tprint n6273i020_asn.fits
# Table n6273i020_asn.fits[1] Tue 14:30:09 24-Jun-2003

# row MEMNAME      MEMTYPE      MEMPRSNT
#
  1 N6273IGXQ      EXP-TARG      yes
  2 N6273IH0Q      EXP-TARG      yes
  3 N6273I020      PROD-TARG      yes
```

The two post-SAA dark images in this case are `n6273kheq_raw.fits` and `n6273khhq_raw.fits`. These are both ACCUM mode `NREAD=25`, `EXPTIME=256s` darks (`FILTER = BLANK`), taken in parallel in all 3 cameras just after NICMOS is transitioned from SAAOPER to OPERATE mode. Two images are taken so that cosmic ray hits incurred during the exposures can be removed. The timing of the start of these two exposures is fairly consistent from orbit-to-orbit, with the first post-SAA dark beginning 174 seconds after the SAA exit, and the second beginning 444 seconds after the SAA exit (i.e. ~256s after the first one begins). Occasionally this timing differs from the norm due to higher priority activities on the spacecraft, but 99% of the time it is 174s and 444s. This repeated timing has good repercussions, which we will describe later. The goal here is to make an image of the decaying persistent image of the cosmic rays. First, the post-SAA darks must be calibrated themselves, to remove all of the instrumental signal (amp-glow, bias/shading, linear dark current, pedestal, etc) (figure 7). This is accomplished by subtracting an appropriate dark reference file image from each, then converting to countrates (DN/s). If desired, make some form of pedestal correction as well. The dark reference file should match the camera

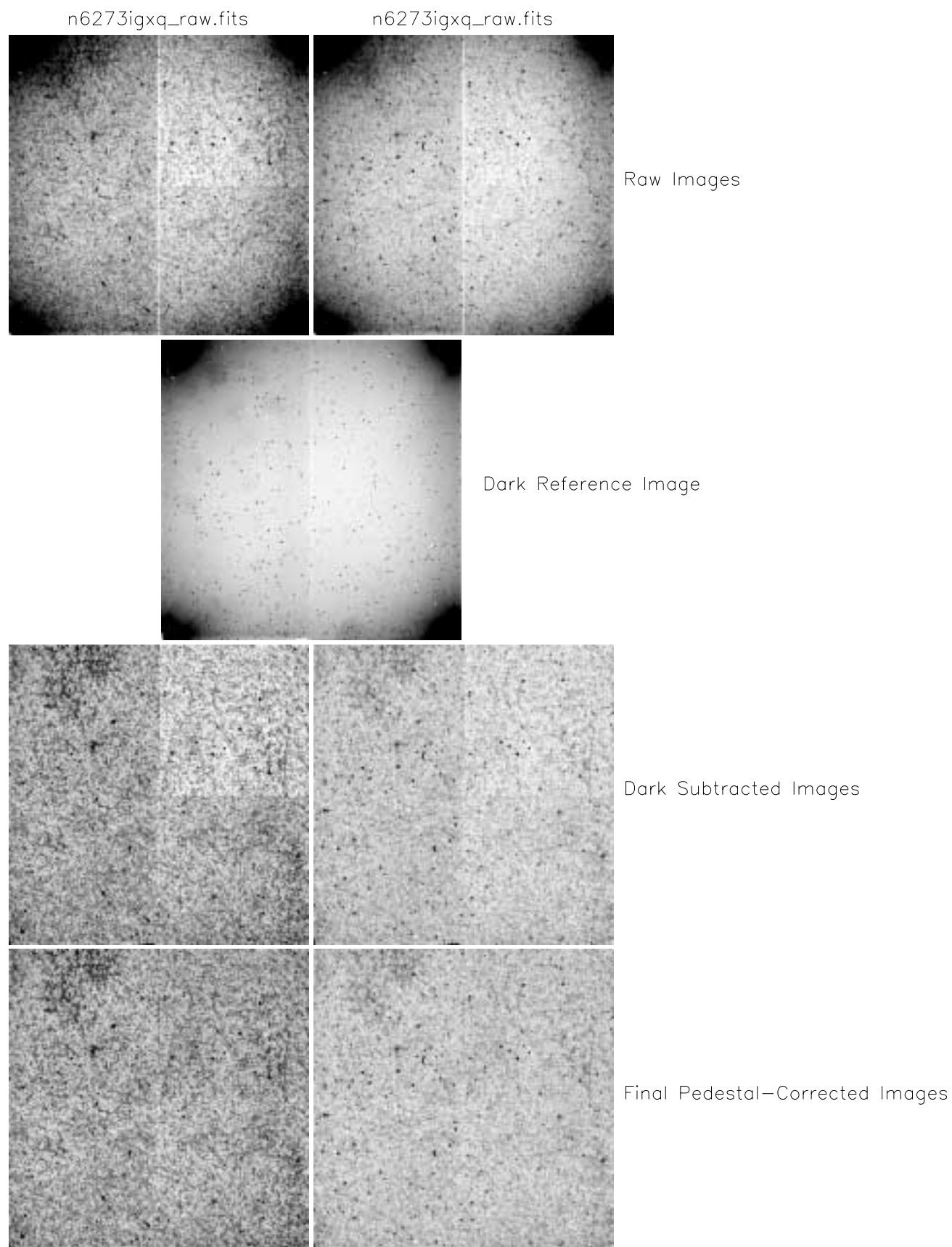


Figure 7. Two post-SAA darks at various stages of the calibration process.

and be the correct one for an ACCUM mode NREAD=25 exposure. Reference files are available in the NICMOS reference file database and can be extracted from the HST Data Archive or downloaded from the NICMOS web page. An upcoming release of calnica will have the ability to make ACCUM mode temperature-dependent darks from the *_tdd.fits reference files. The IDL routine saa_clean.pro already has this capability built in. You can also make your own reference file from dark data in the archive. It is absolutely critical that the two post-SAA darks be as low-noise as possible, and the dark subtraction is one place where noise is added. In fact, the reason the post-SAA darks are taken as NREAD=25 ACCUM mode exposures is to get the benefit of the Fowler-sampling for noise reduction (*note that the realized reduction is not $\sqrt{25}=5$ due to the added poisson from the extra amp glow; its more like a factor of 2 at the center and 1.5 in the corners...every little bit helps here.*)

Once the two post-SAA dark images have been calibrated we need to combine the two - rejecting cosmic rays - to make an average image of the persistent signal at the start of the orbit. Since the persistent signal is decaying exponentially with time (figure 2), the second image must be multiplied by a scale factor to match the signal from the first before averaging. This exploits the property of an exponential that two points separated by an equal time on the curve differ by the same multiplicative constant *regardless* of where along the curve those two points are located - early or late in the decay. In other words, it doesn't matter if the persistent signal at a particular pixel is coming from a hit near the very end of the SAA passage or whether it was an early hit. The scale factor for those two incidents will be the same in the two post-SAA images because the time between the first and second image is the same for both the early and late hits. It is this very property which will be exploited later when using the SAA persistence map to remove this signal from the impacted science images.

The scale-factor between the two images can be found by plotting the signal in the first image against the signal in the second image (figure 8). The slope of the linear fit to this is the scale factor. Of course this is essentially the same as just taking the mean of the ratio of the two images, but the fit also allows for an easy measurement of the inherent noise in each of the images and use of those sigmas to flag the cosmic rays in each image. It also shows graphically that indeed one scale factor works over the wide range of persistent signal across the image. In addition, the slope-fit provides some immunity to any leftover global pedestal in either image. A zero-point shift along either axis drops out in a fit, but not in a mean ratio.

Recall that the timing between the first and second images relative to each other and to the SAA exit is the

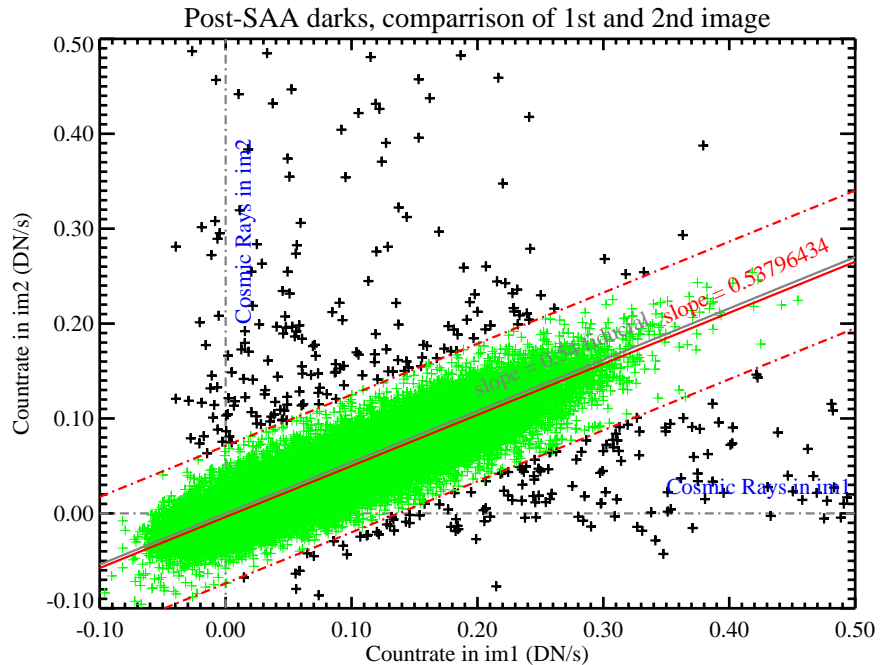


Figure 8. Plot of post-SAA dark #1 vs. dark #2, with linear fit, nominal slope and parallel sigma bars. A firm 1.5-sigma clip is used in the algorithm.

Post-SAA Dark Image1 to Image2 Measured Scale-Factor Histograms

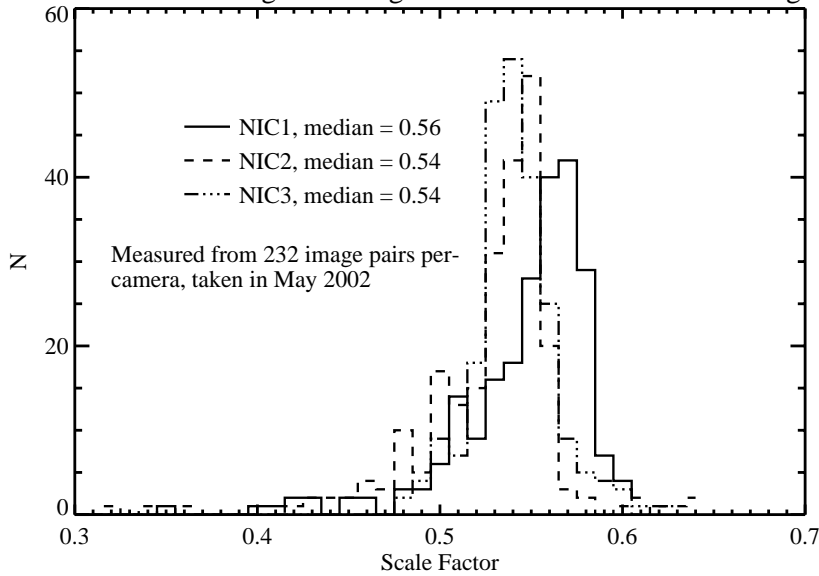


Figure 9. Histograms of im1 or im2 scale factors as a function of camera for a large number of post-SAA darks. All image pairs here had identical post-SAA timing - i.e. 174s and 444s after SAA exit.

same 99% of the time. Because of this, the scale factor between the two images is also the same every time (figure 9). Since the persistent signal is lower in shallower SAA passes, fitting for the scale factor (or even just taking the ratio) can become questionable. Using a default scale factor is useful in those cases, and in the saa_clean.pro IDL tool the default (one factor per-camera) is the normal mode of operation, with fitting and user-supplied values as options.

Once the scale factor is known, just scale and combine (with optimal weighting and rejection of any CRs) to make the SAA cosmic ray map image SAACRMAP (figure 10). Intu-

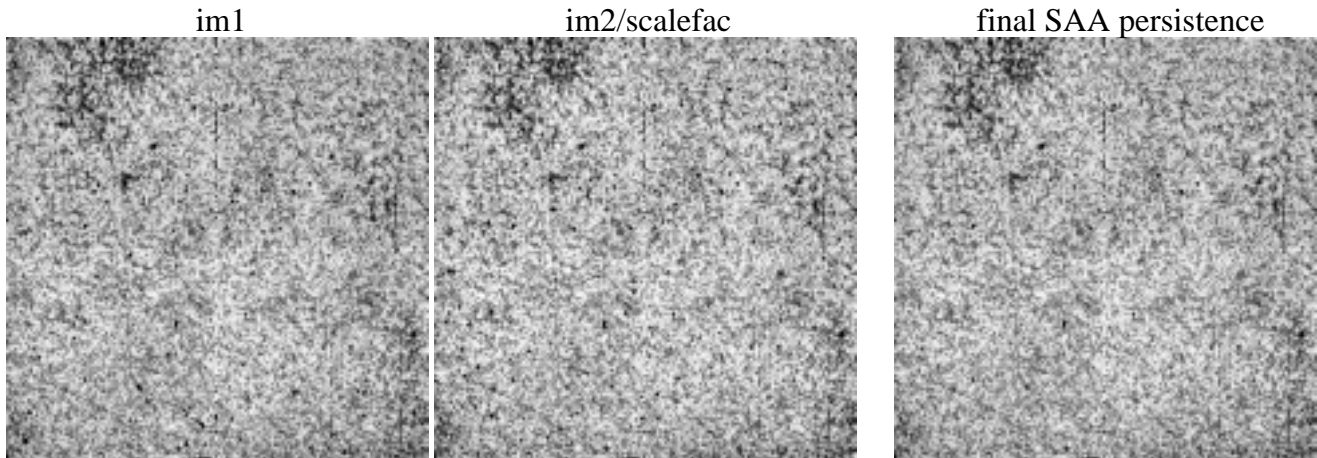


Figure 10. Calibrated post-SAA dark images, scaled to match, and the combination of the two. This “SAACRMAP” is the model image that is then iteratively scaled and subtracted from the science data to remove the persistent signal

itively the optimal weighting should be proportional to the scale factor between the two images, since the noise should be the same in each - only the signal has changed. For a scale factor of 0.54, the weights would be 0.77 and 0.23 respectively. Just to be sure, we made a sweep of noise-reduction vs. image weighting using saa_clean.pro on a sample of impacted science images. Based on this (figure 11) the optimal weights appear to be 0.7 and 0.3 for image1 and image2 respectively.

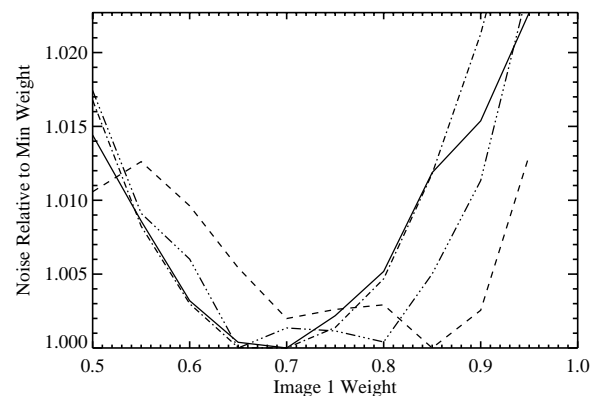


Figure 11. Relative total noise in various science images as a function of the weight scaling used for the post-SAA darks. (curves for shallower crossings are a bit noisier than the deep crossings)

4) Scale and Subtract the Persistent Signal

Now that we have an optimal image of the cosmic ray persistence signal we can subtract it from the science image. The amplitude of the persistent signal in a given image will depend on how long after the SAA-EXIT the exposure started, the depth of the passage itself and the integration time (figure 3). Remember that we are integrating over an exponentially decaying signal. Therefore the persistence image will have to be multiplied by some scale factor to match the persistent signal in the data. This scale factor will be less than 1.0 (assuming the science images have units of DN/s), since the science images were taken *after* the post-SAA darks. The `saa_clean.pro` algorithm determines the best scale factor by iteratively multiplying the persistent image by small scale factors, subtracting that from the science image and then measuring the sigma in the result (by measuring the half-width of a Gaussian fit to the resulting image histogram). The best scale factor will result in the lowest sigma. This produces a curve of scale-factor vs. sigma (figure 12).

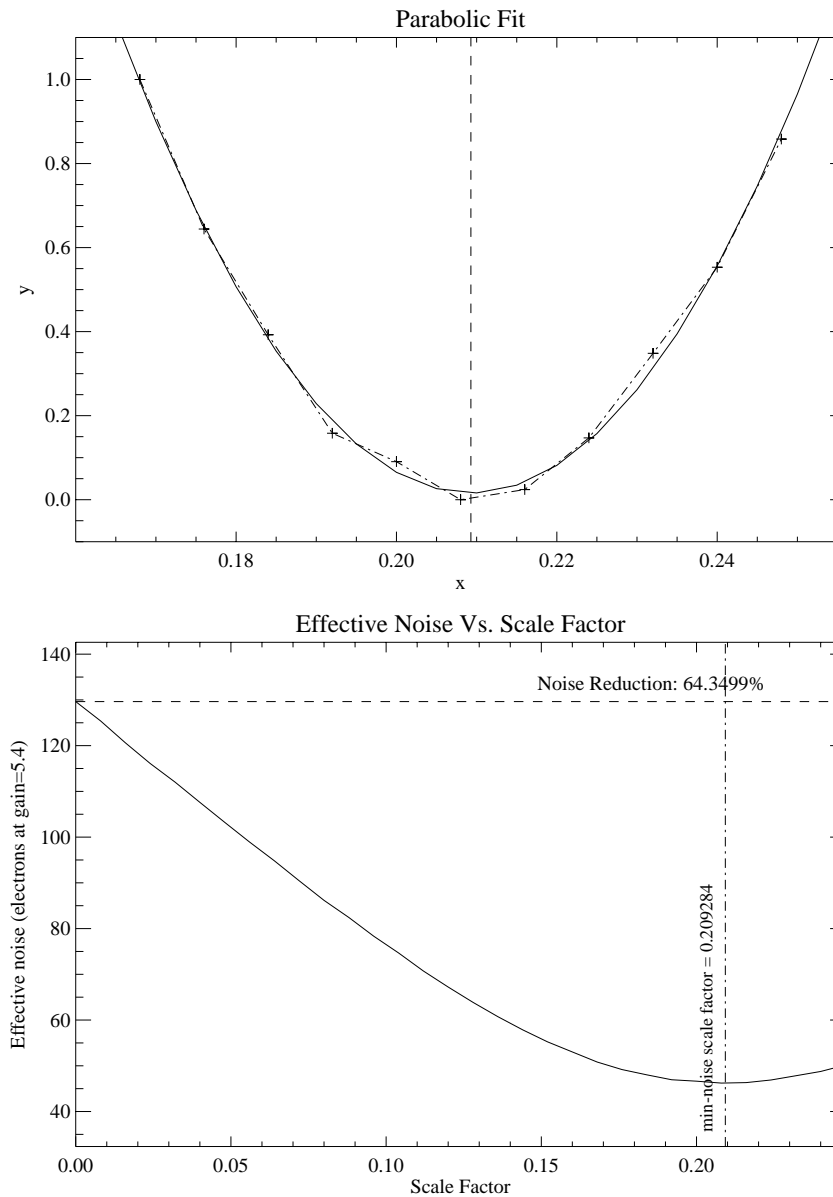


Figure 12. Bottom panel shows image noise (half-width of gaussian fit to image histogram) as a function of scale factor. Top panel shows parabolic fit to the minimum. The effective noise reduction is also shown (64% in this case)

The minimum scale factor is determined by fitting a parabola to this curve. One additional step is performed on the persistence image before the iteration starts: it is multiplied by the same flatfield reference file that was applied to the science image during the pipeline processing. This is done because the persistent signal in the science image has been multiplied by the flatfield pattern and we want the model persistence image to be as good a match to the persistent signal in the data as possible. The flatfield pattern varies dramatically with wavelength, so this should be done for each image individually. A bad pixel mask is also used to prevent known bad pixels from biasing the sigmas.

In principle, only one scale factor should be necessary to correct the entire image. However, there is a wide range of S/N in the persistence image. Some CR hits during the SAA passage have higher energy than others, or occurred later in the passage. Any inherent noise (mostly readnoise) in the persistence image is multiplied by the scale factor and gets added in quadrature to the science image. The higher the scale factor, the larger this proportion of added noise that gets added to the resulting image, limiting the overall effectiveness of the correction. If instead of using a single scale factor for all the pixels in the image we separate the high signal pixels from the lower signal ones, we can utilize the higher S/N of the high signal ones to get away with a larger scale factor for those. If we don't do this, the highest persistent signal pixels will be undersubtracted, because the overall readnoise will limit the optimal scale factor overall. In reality there is a whole range of scale factors running continuously from the low signal to the high signal pixels, but after experimenting it was determined that a simple threshold and 2-signal regime worked best in preactical application (determining a good scale factor requires a certain number of pixels, and if the bins are divided too small the fit is poor). The two-stage approach gains a few percent in overall noise reduction as compared to using just a single scale factor, and the spatially localized gains are even higher.

The `saa_clean.pro` routine determines the threshold between the two signal regimes by locating the peak and width of the histogram of the input science image. The threshold is taken to be 3.5 sigma above the mean. A sample of science images was used to experimentally determine where the optimal threshold should be, and it turned out to be 3.5 on average (figure 13). The optimal threshold varies somewhat depending on how badly a given image is impacted, but

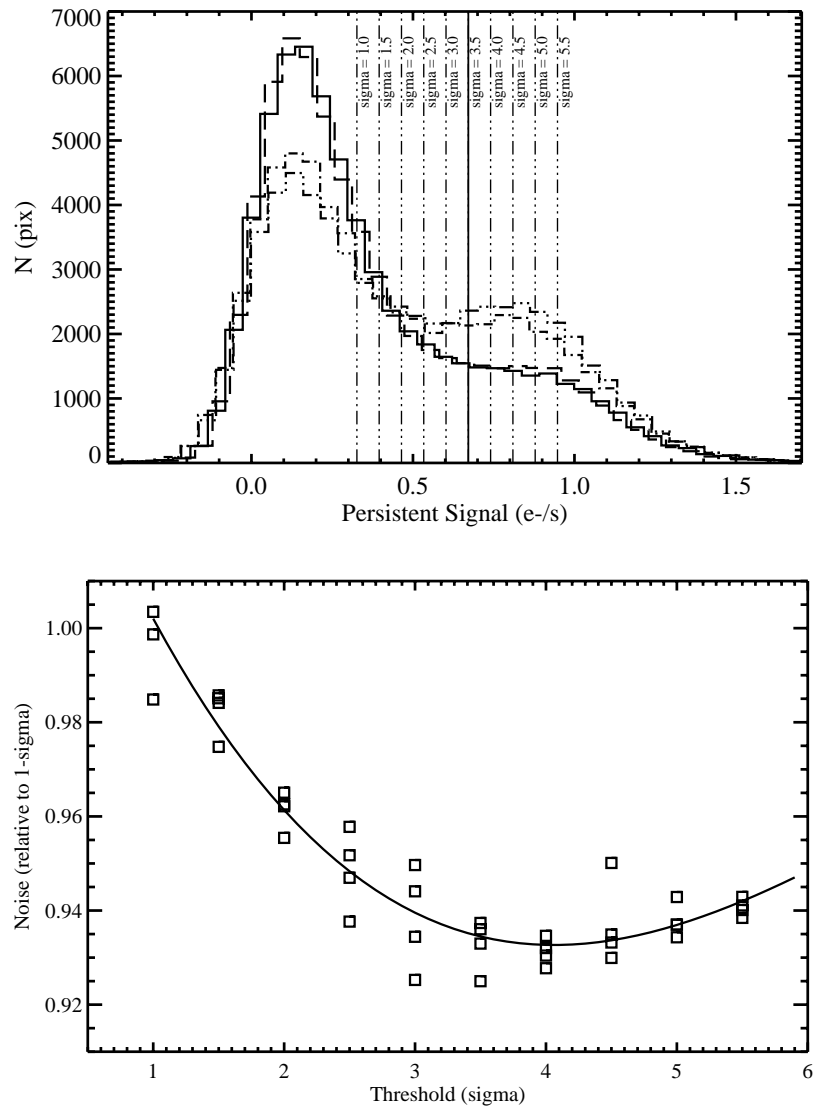


Figure 13. Bottom panel shows relative noise as a function of threshold in sigma. Top panel shows a range of thresholds overplotted on a few impacted image histograms

the relative noise reduction in the final product is negligible over the 3-5 sigma range. As can be seen in the top panel of figure 13, the best threshold location roughly corresponds to the inflection point between the histogram of the persistent signal (which is always positive) and the mean inherent image noise.

As the iteration proceeds for each of the steps, the routine prints a number of statistics about each of the 2 iteration passes. For example:

```

Iterating for SAA-to-data scale factor...
Using hi/low threshold of 0.136 DN/s in the saaper image...
Percent Done:
 10
 20
 30
 40
 50
 60
 70
 80
Parabola minimum at x= 0.214039
Parabola minimum at x= 0.214039
zero-mode scale factor is: 0.0960000

min-noise (best) scale factor is: 0.214039
effective noise at this factor (electrons at gain=5.4): 49.7772
noise reduction (percent): 27.5926
Percent Done:
 10
 20
 30
 40
 50
 60
 70
 80
Parabola minimum at x= 0.182049
Parabola minimum at x= 0.182049
zero-mode scale factor is: 0.000000

min-noise (best) scale factor is: 0.182049
effective noise at this factor (electrons at gain=5.4): 42.7149
noise reduction (percent): 37.4282

```

If the noise reduction is less than 1% then no correction is applied and the output image is the same as the input image. This threshold is arbitrary and can be overridden. If the high signal level reduction is less than 1% but the low signal reduction is better, the low scale factor is applied to the entire image. The “noise reduction (percent)” given here is for each of the two signal ranges separately. Total noise of the final product can be best measured on the final image after execution. The cleaned product is displayed side-by-side with the input image and is written out as a fits image (figure 14). Its best to run **pedsky** on the output image one last time to clean up any remaining quad-based pedestal (figure 15) (see section “2” for details).

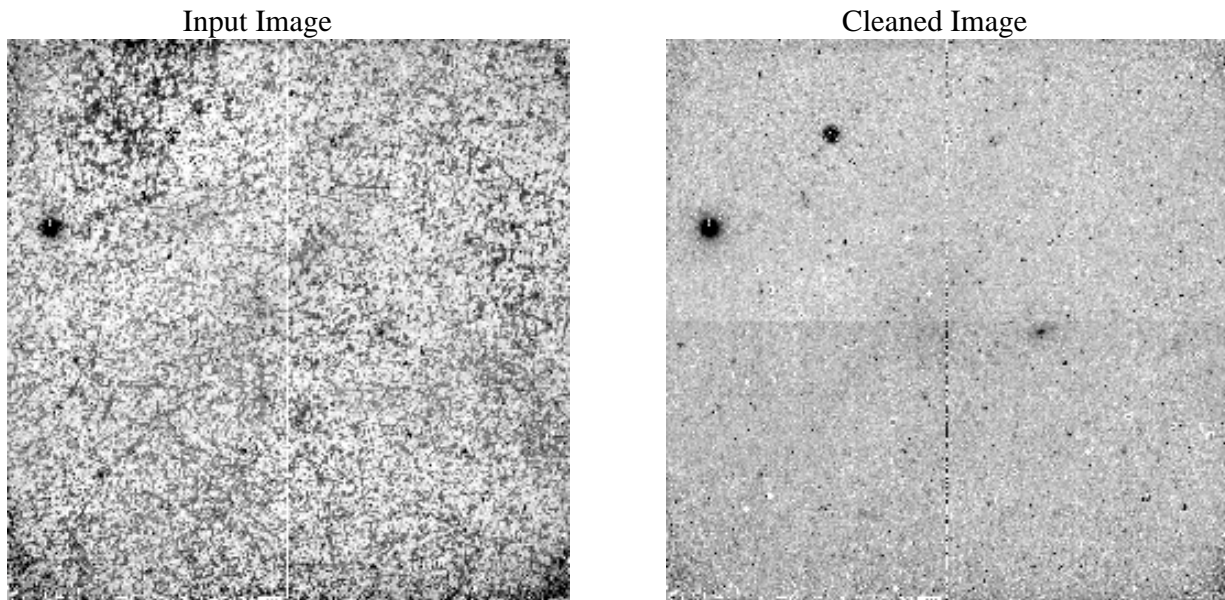


Figure 14. Comparison of input science image (left) and SAA CR-persistence removed result (right). Note that the bright spot that appears above center of the upper left quad of the cleaned image is an imprint of the coronagraphic hole that is in the flatfield reference file. This region should be masked whether the post-saa correction is applied or not as part of any image combination or drizzling that is done to make a final science product.

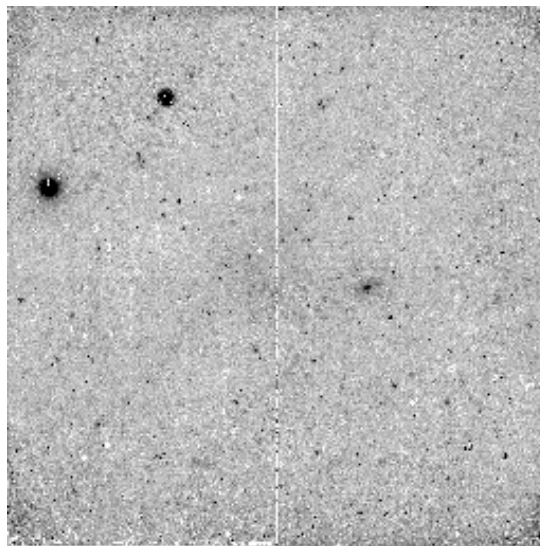


Figure 15. Final cleaned science image after a final pass through **pedsky**.

Some Example Datasets

Presented here are a few example datasets which show the relative effectiveness of the saa_clean algorithm.

Example 1:

The first example is a set of 8, NIC2 F160W, 1350s exposures, taken one or two per orbit, heavily CR- persistence impacted, of a faint galaxy. These same observations were then repeated

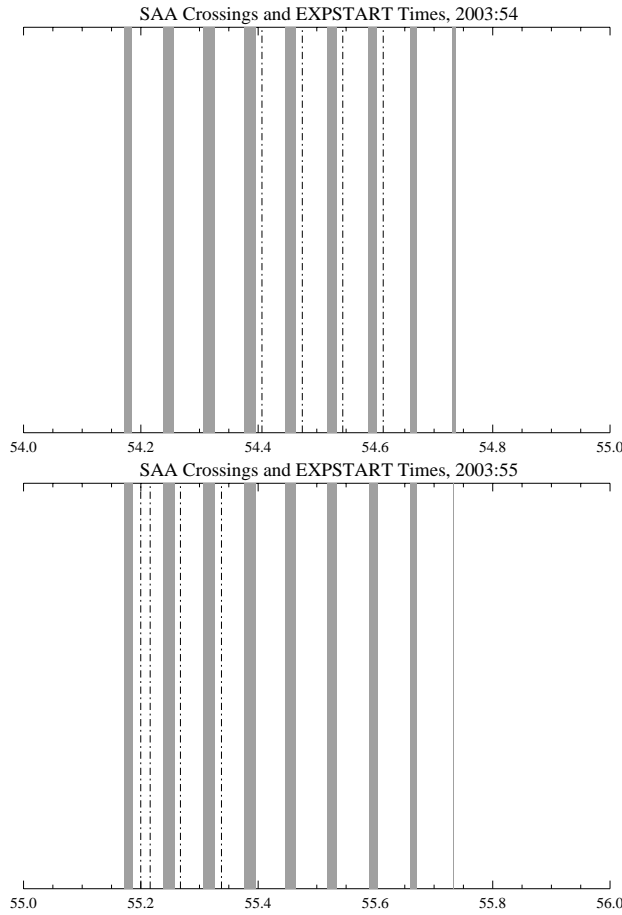


Figure 16. Observations (dashed lines) relative to SAA crossings (vertical gray bars). The heavily impacted epoch 1 observations are on the left, and the SAA-free epoch 2 observations are above. Note that the first two observations in the epoch 2 set were left out so that an equal exposure time comparison could be made between the two epochs.

2 weeks later in entirely SAA-free orbits, providing a unique control set to look at the algorithm effectiveness. Figure 16 shows the two sets of observations and their timing relative to the SAA passages. There were two additional exposures in the epoch 2 set (non-impacted), which will be left out of this set in order to match the total exposure times of both epochs for direct comparison. All of the epoch 1 images were run through the persistence cleaning algorithm, following all of the steps outlined in the previous sections. In addition, a “sky” subtraction was performed on both epochs to remove any remaining dark-current structure before drizzling. Figure 17 shows the before and after images of the epoch 1 data, and the histograms of each. Figure 18 shows the raw and “sky” subtracted epoch 2 (non-impacted) data. In this case (1350s exposure taken immediately after SAA exit), the total noise in an image can be reduced by up to 300% in a deep passage. The “sky” subtraction itself helps an addition few percent, mostly from the correct subtraction of thermally induced dark current leftover in the corners of the image. Note in figure 17 that there is a gradation in depth of passage from left to right. This shows up in the histogram widths and the intensity of the persistent signal in the images across the top. The 6th image from the left shows almost no persistent signal. Looking at the left side of figure 16, you can see that this image was the second one taken in an orbit following the first passage of HST through the SAA on day 55. The first passage of a given day tends to be a shallow, grazing passage, and although the length of the passage (width of the gray bar) appears as long as some of the others, the particle density reached on that passage was lower than some of the middle passes of a given day. Note also that the excess corner dark is not visible in this image at the top of figure 17. This is also evident in every other image in figure 18 (epoch 2 data). Once again, it is the second image in each orbit that is free of this extra dark. It is believed that the source of this signal is thermal heating of the bulk material by the output FETs on each corner while the device is being read rapidly.

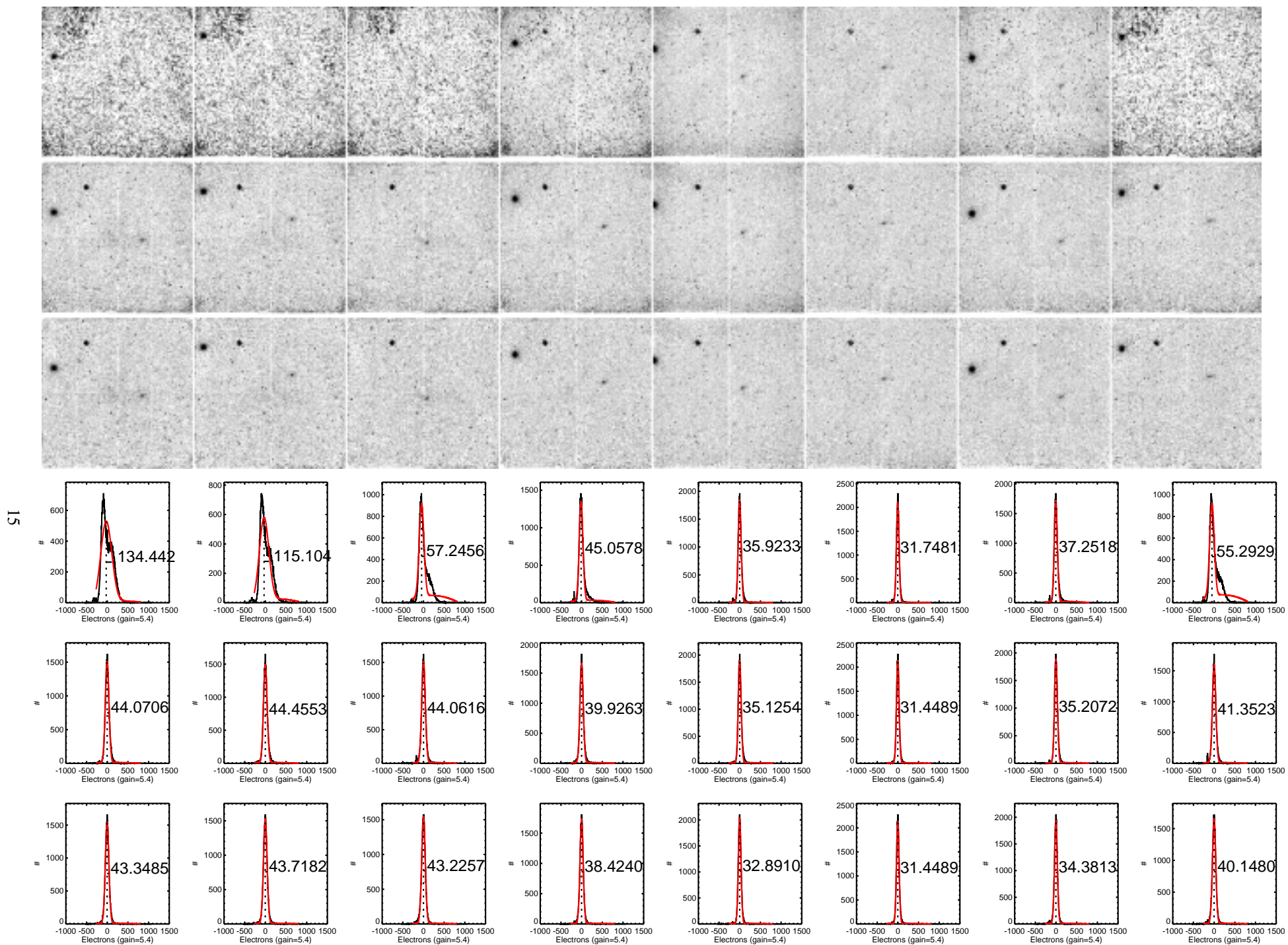


Figure 17. Epoch 1 (impacted). Top row: input images. Middle row: after SAA persistence removal. Bottom row: after "sky" subtraction.

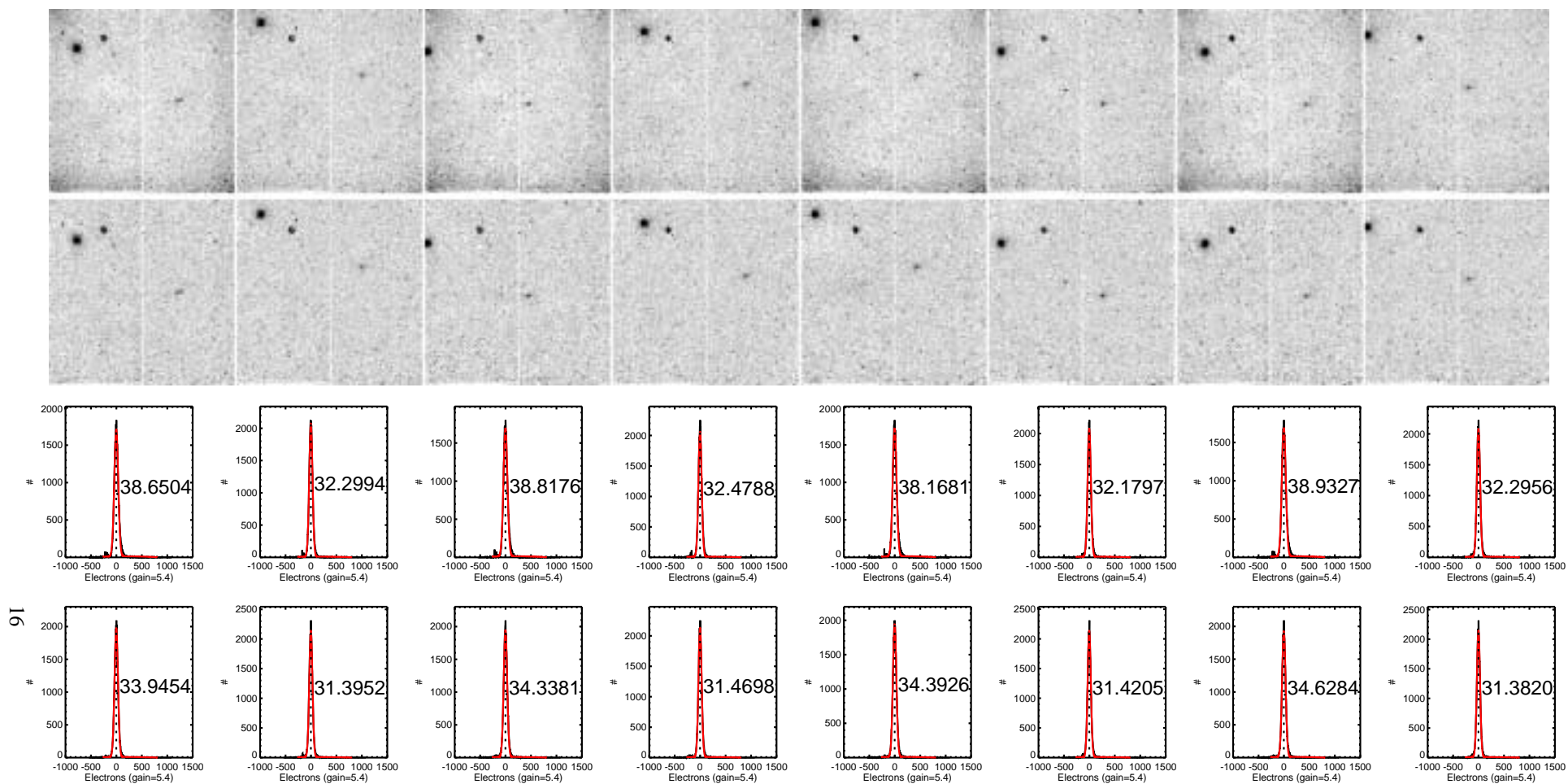


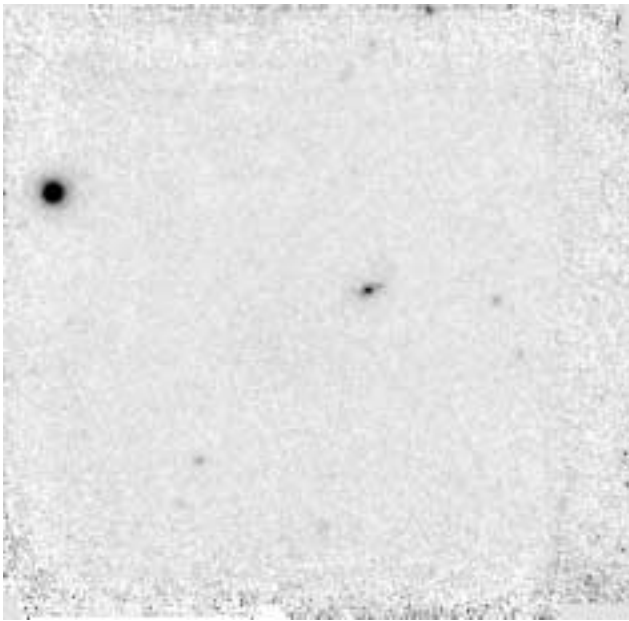
Figure 18. Epoch 2 (non-impacted). Top row: input images. Bottom row: after “sky” subtraction.

This occurs during autoflush while the detectors are idling, and also when the detector is read rapidly as in the Fowler sampling that is done in the post-SAA darks themselves at the beginning of an impacted orbit. This effect is a topic of its own and will be discussed in a future ISR.

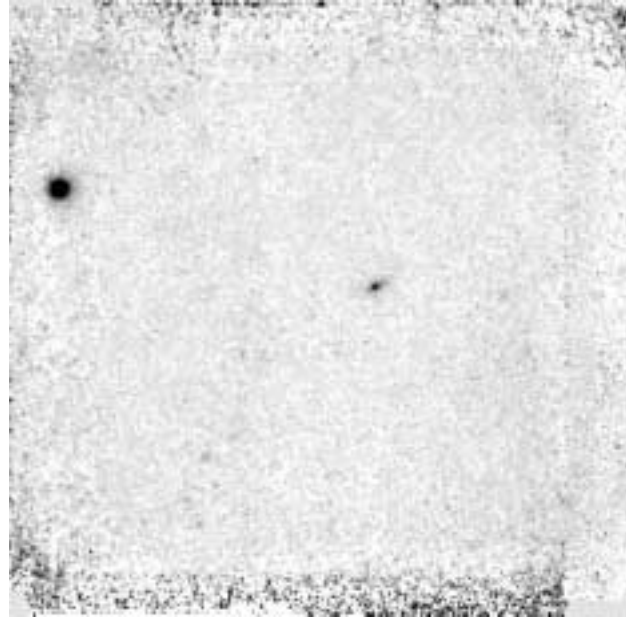
Comparing the final cleaned images in figure 17 with the non-impacted images in figure 18, on average, the cleaned images have ~25-30% higher noise than equivalent non-SAA impacted images. This varies somewhat depending on the depth of the passage. This extra noise comes from a combination of the scaled readnoise of the SAACRMAP image and the photon noise of the persistent signal itself, which cannot be removed.

Now we can drizzle all 8 image at each epoch into final science products and see what the noise reduction looks like collectively. Information about the STSDAS **dither** package can be found in the “HST Dither Handbook” (Koekemoer et al., 2002). We won’t go into the details of the drizzling here except to say that optimal weighting using the QE maps (flatfields) and appropriate bad pixel masks were used. Three versions of the epoch 1 (impacted) data were produced (figure 19). The image on the left is a drizzle of the pipeline-calibrated only data (top row of images in figure 17). The bad pixel mask used here was the same one used in the image on the right, which is a drizzle of the persistence-removed data (bottom row of images in figure 17). The middle image is a “best effort” drizzle using the pipeline-calibrated only data. In this case, bad pixel masking in the “blot/driz_cr” step is done with a tight threshold, allowing much of the CR persistence to be flagged and rejected during the drizzling process. This is the best one could do without a persistence removal tool, but suffers from the rejection of much of the exposure time at each pixel. Figure 20 shows the optimal drizzle combination of the epoch 2 (non-impacted) data side-by-side with the SAA persistence cleaned data. The impacted image set, after persistence removal, has 22% higher noise in the final drizzled product than that of the non-impacted image set.

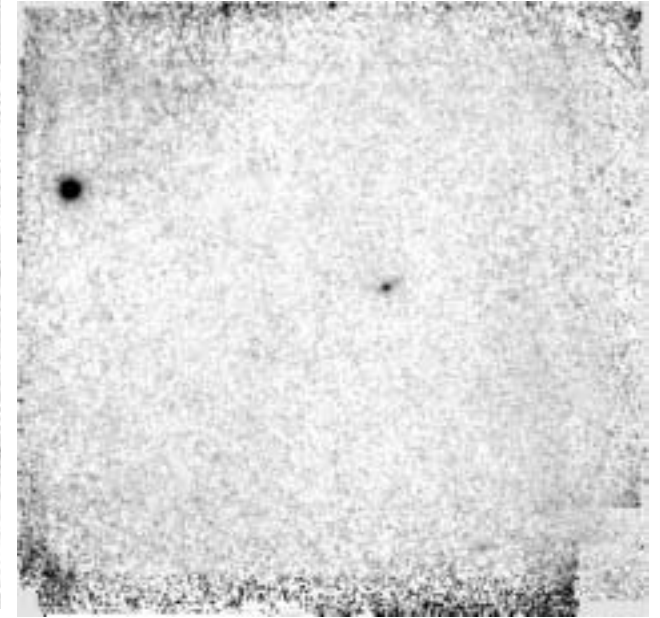
The results for this example are very much a worst-case scenario. In the impacted epoch 1 set here, 7 of the 8 images were the first image taken in an orbit immediately following exit from the SAA (the second half of each orbit was used for a different filter in this case). Operationally it would have been better to take all the exposures in a given filter, then switch to the other filter, taking at least 2 exposures per-orbit. This would mean that half of the exposures - the ones at the beginning of the orbit - would be heavily impacted, and the other half would be significantly less impacted because of the time elapsed since SAA exit to the start of the exposure. From figure 3, assuming two 1350s exposures per orbit, the second exposure would have less than half the induced noise of the first exposure for a deep passage. Less induced noise from persistence will result in less noise in the final combined product and a closer match to a non-impacted equivalent dataset. In any case, some additional exposure time could be added as a hedge to reach targeted S/N levels in the event of SAA-impacted orbits.



The Good



The Bad



The Ugly

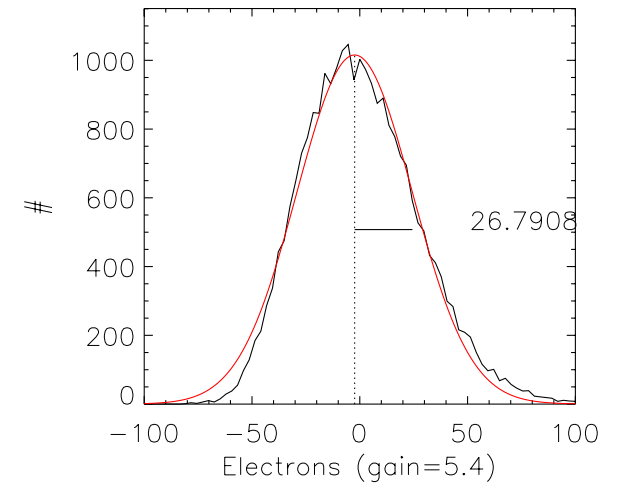
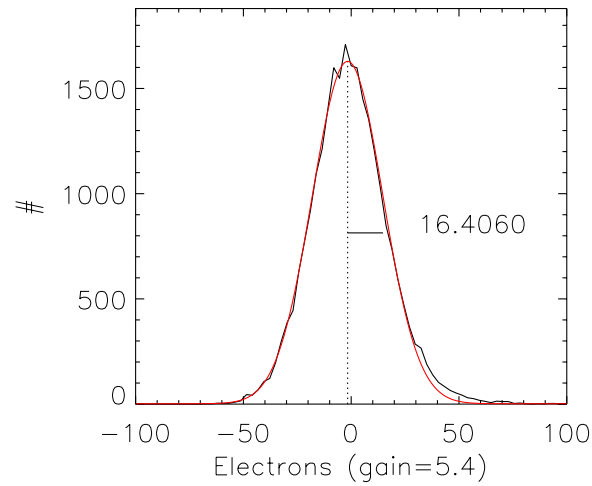
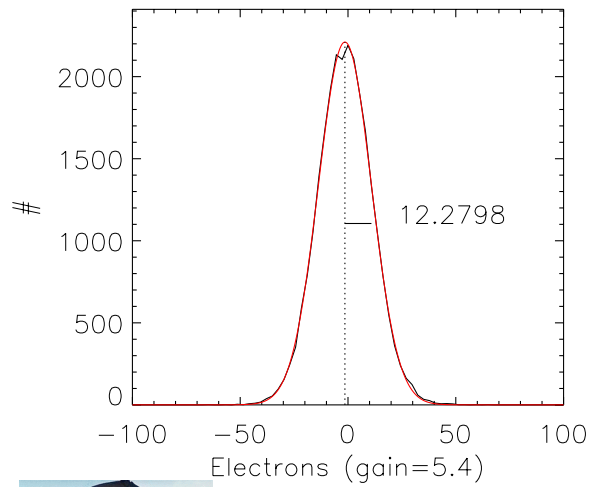


Figure 19. Drizzled composite images of the 8 1350s SAA-impacted images. Left: SAA persistence cleaned. Middle: uncleaned data, but drizzled with hard blot/driz_cr rejection threshold (i.e. “best effort”). Right: uncleaned data drizzled with same weightmap as the cleaned data Histograms are in electrons equivalent at 1350s.

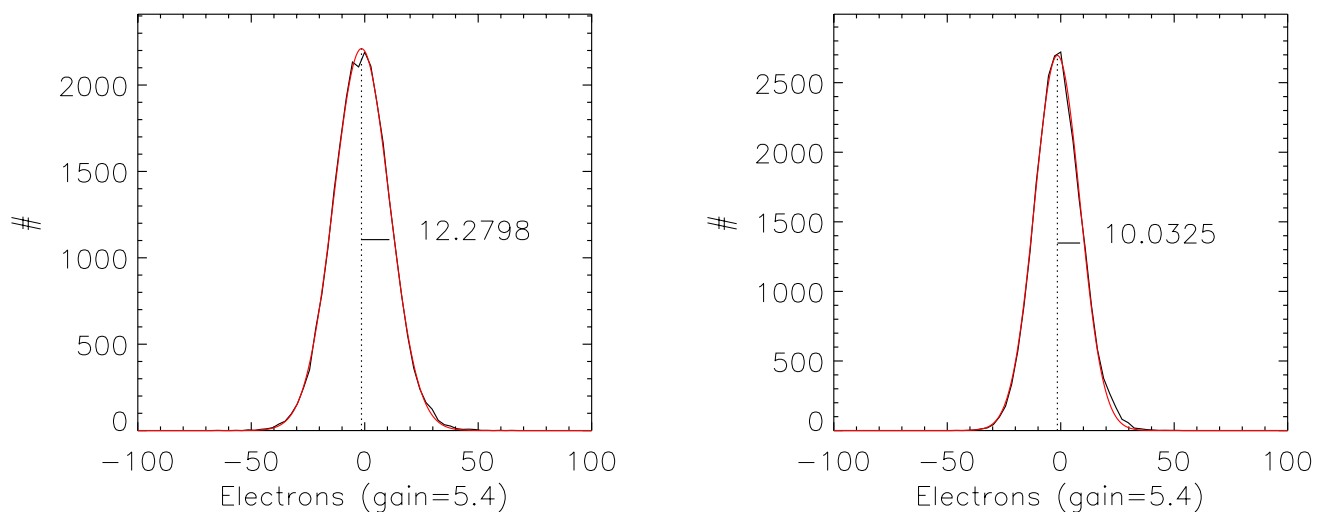
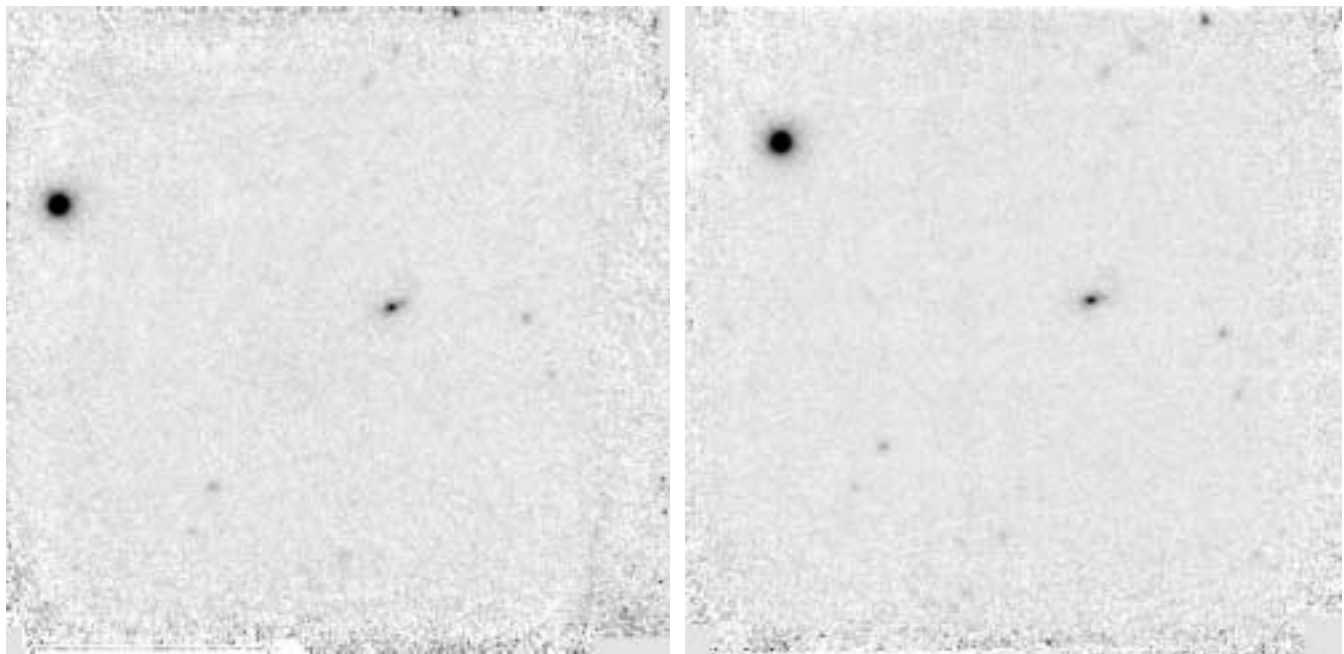
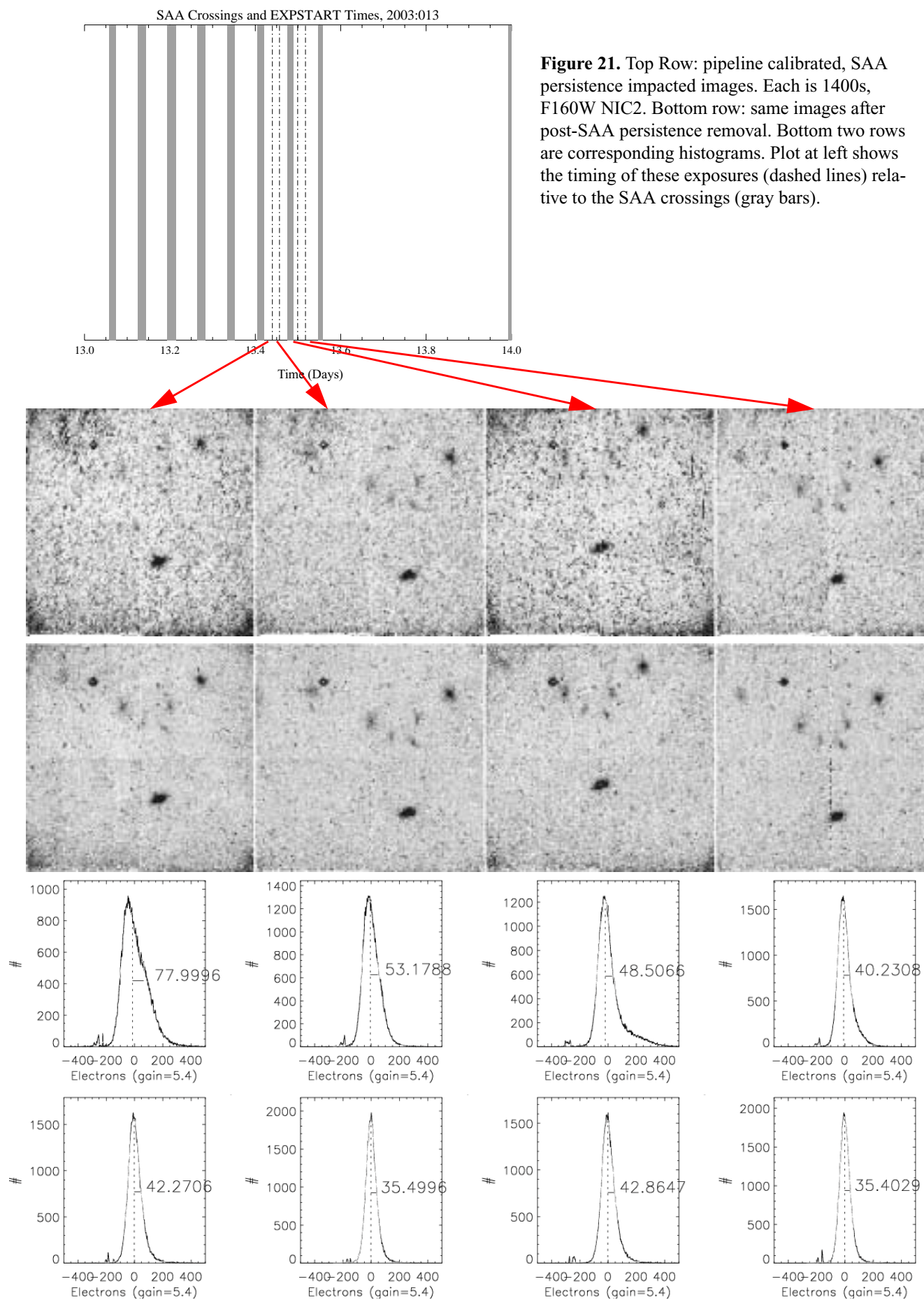
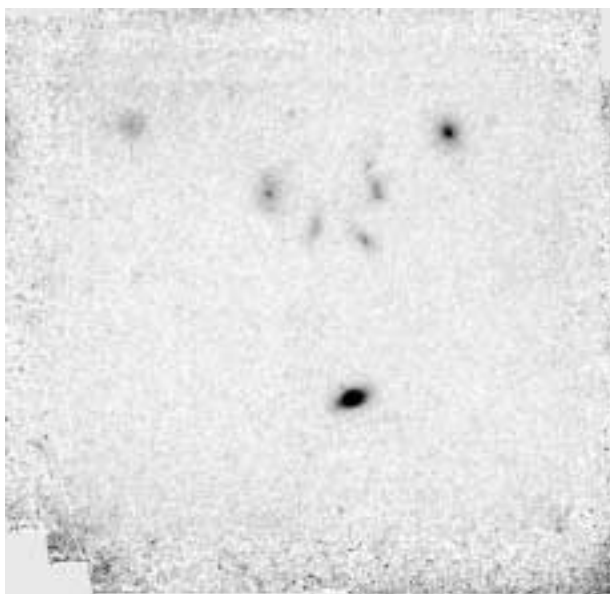


Figure 20. Side-by-side comparison of SAA persistence removed (epoch 1) final image (left) with the SAA-free (epoch 2) final image (right). Histograms are in electrons equivalent at 1350s. SAA cleaned image has 22% higher noise than SAA free image.

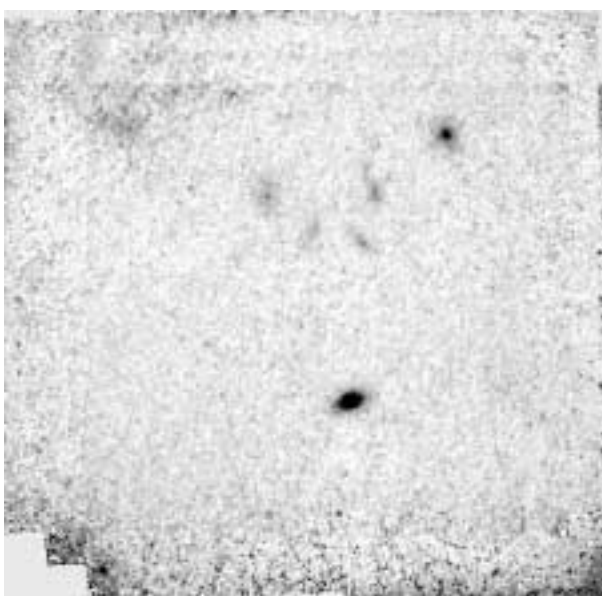
Example 2:

The second example is a set of four 1400s, NIC2 F160W exposures of a small group of galaxies. We don't have an SAA-free control group for this set, so this is strictly a before-and-after comparison of the efficacy of the persistence removal algorithm. The four input images and their cleaned versions are shown in figure 21. The final drizzled products are shown in figure 22. Note that this dataset was used to determine a number of options in the `saa_clean.pro` routine such as how to best normalize and apply the flatfield to the persistence map and what effect pedestal removal had on the results. Some of these figures of merit are shown in appendix B.

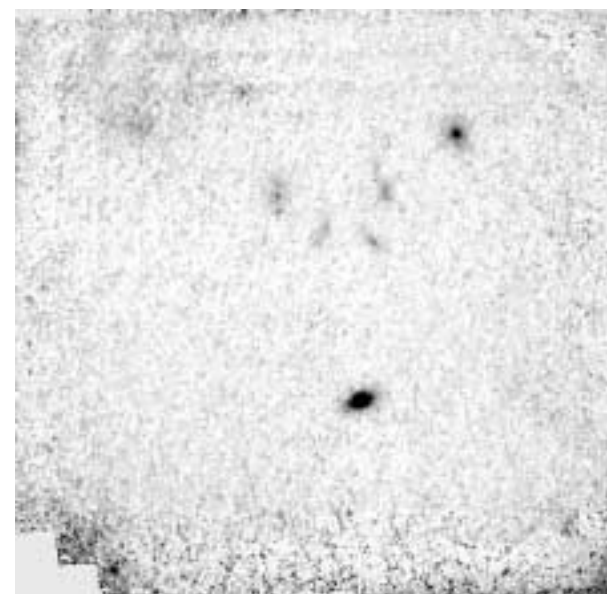




The Good



The Bad



The Ugly

21

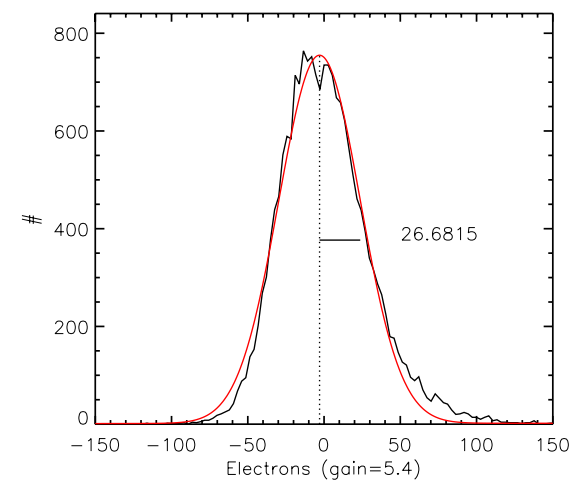
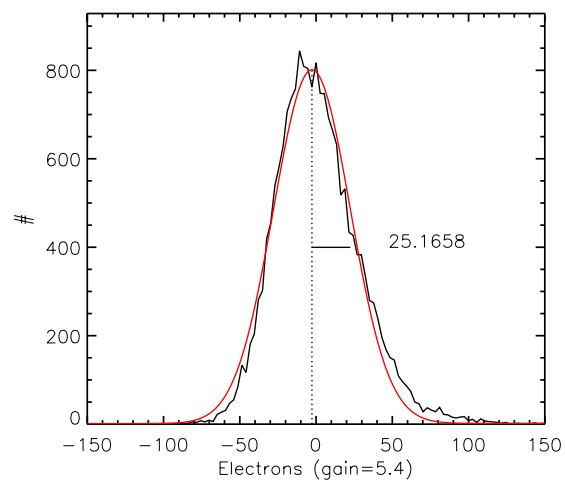
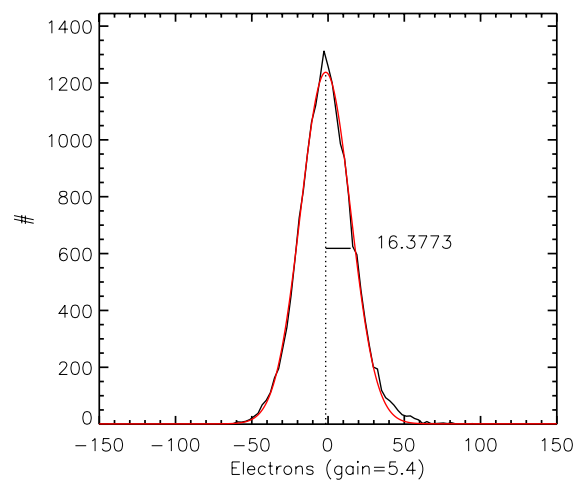


Figure 22. Drizzled composite images of the 4 1400s SAA-impacted images. Left: SAA persistence cleaned. Middle: uncleaned data, but drizzled with hard blot/drizz_cr rejection threshold (i.e. “best effort”). Right: uncleaned data drizzled with same weightmap as the cleaned data Histograms are in electrons equivalent at 1400s.

Example 3:

The third example is just a 4-panel image provided by Dave Thompson of Cal Tech. No drizzling was performed here. These images are strictly shift-and-average, using integer pixel shifts and the **imcombine** task in iraf. In each of the four cases, 12 NIC2 images, each 704s, were combined.

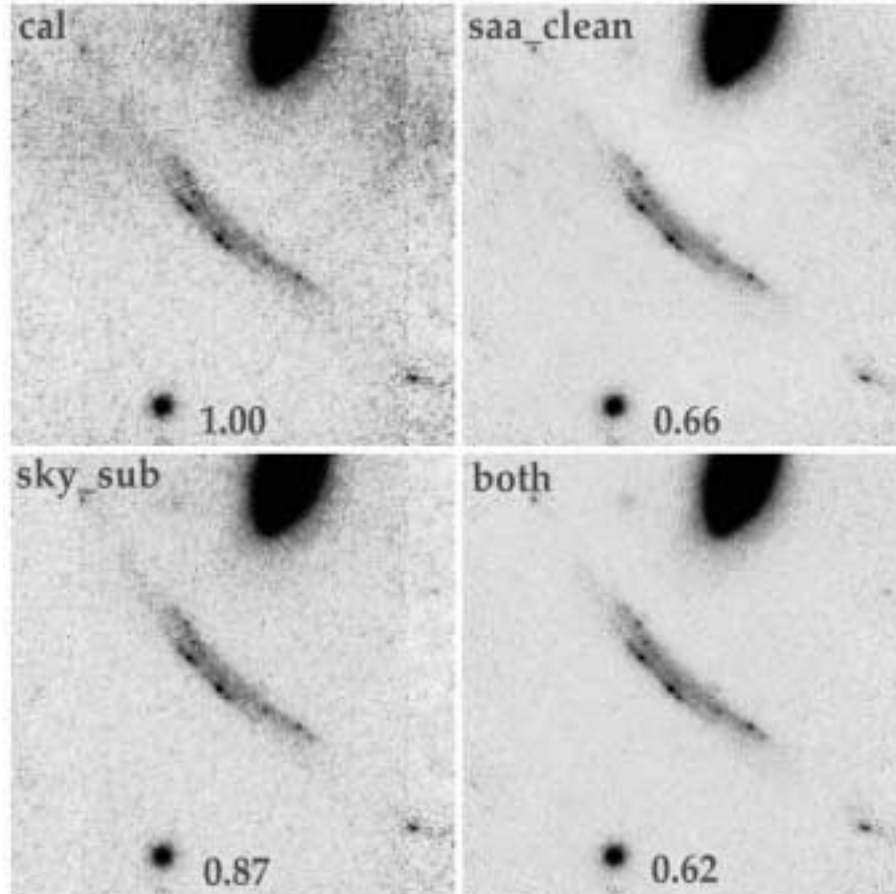


Figure 28. upper left: average of the calibrated data straight out of the calnica data pipeline. Lower left: average of calibrated data with background (sky) subtraction. Upper right: average of calibrated data with post-SAA CR persistence removal. Lower right: average of calibrated data with both background subtraction and post-SAA CR persistence removal. The numbers in each panel are the ratio of the measured background RMS as compared to that of the image in the upper left panel.

Conclusion

Persistent or latent images of cosmic ray hits on the NICMOS detectors as HST passes through the SAA can result in an increase in the total effective noise of an image by up to factors of 3 to 4 over data taken in non-impacted orbits. An algorithm has been presented to utilize a pair of post-SAA darks to iteratively scale and subtract this signal from science data, significantly reducing the injected noise. The algorithm has been optimized for a number of variables, and tests have shown that a factor of 4 reduction in the noise is possible in many cases. Badly impacted individual datasets can have their noise reduced to within 25-30% (down from 300-400%) of what it would have been if taken in SAA-free orbits. Combined datasets (via drizzling with some rejection) can get to within 22% of their non-impacted equivalents. This technique, perhaps combined with a small amount of additional exposure time, can be used to recover the desired S/N of a given science program when executed during even the deepest of SAA passages. The algorithm has been written as an IDL routine, and work is underway to include it as an STSDAS task in the **hst_calib.nicmos** package, and possibly as part of the normal OTFC calibration pipeline.

Acknowledgements

We would like to thank Lou Strolger, Adam Riess and Dave Thompson for permission to use their science data in this document as well as for the development and shakedown of the algorithm itself. Eddie would also like to mention that Bill Sparks was the first person in the NICMOS group at STScI who thought about ways of characterizing and removing the CR-persistence when it was first observed back in 1997. His original thoughts led to the solutions outlined here.

Index to Appendices

Appendix A: Description of the saa_clean.pro IDL implementation of the algorithm, and a print out of the main function source code.

Appendix B: Figures of merit for various methods of applying the flatfield to the SAACRMAP image and the effects of pedestal removal before an after execution on the final noise reduction.

Appendix C: Derived trap density maps for each of the NICMOS detectors.

Appendix D: Noise statistics of SAA persistence and operational SAA contours for NICMOS.

References

- Bergeron, L. E. and Najita, J. R., 1998, "Cosmic Ray and Photon Image Persistence and Ways to Reduce its Effects," American Astronomical Society, 193rd AAS Meeting, #36.06; Bulletin of the American Astronomical Society, Vol. 30, p.1299
- Dickinson, Mark, et al., 2002, "HST Data Handbook," v5.0, http://www.stsci.edu/hst/nicmos/documents/handbooks/DataHandbookv5/NICMOS_longdhbcover.html (Baltimore: STScI)
- Koekemoer, A. M., et al. 2002, "HST Dither Handbook," Version 2.0 (Baltimore: STScI).
- Najita, Joan, Dickinson, Mark and Holfeltz, Sherie, 1998, "Cosmic Ray Persistence in NICMOS Data," Instrument Science Report NICMOS-98-001 (Baltimore: STScI)

Appendix A:

The algorithm described in this document has been implemented as an IDL function called `saa_clean.pro`. This function calls a number of other IDL subroutines, some of which are part of the main *astron* library and some of which were written by the authors. All IDL code and necessary subroutines are included in a tarball available locally at stsci here:

/data/joecool7/eddie/saa_clean.tar

This file can be requested through the stsci helpdesk (help@stsci.edu), and might eventually be posted on the NICMOS website, although it is expected to be superseded by an STSDAS task. The README file from the tarball is included here, along with the actual source code for the `saa_clean.pro` function itself for completeness.

README file from tarball:

To clean the persistence from an saa-impacted `_cal.fits` NICMOS file, do the following (using the file `n8g6cnh3q_cal.fits` in the `demo_data` dir as an example):

```
[Copy the demo files into the current dir]
cp demo_data/*.fits .
```

First, run the `stdas.hst_calib.nicmos` task `pedsky` on the `_cal.fits` file

```
ni> pedsky n8g6cnh3q_cal.fits n8g6cnh3q_cal1.fits salgori="quick"
```

Note that `pedsky` needs the flatfield that was used on this data, as does the `saa_clean.pro` code. You will also need an `_tdd.fits` dark reference file. I have included these with the demo data for all 3 cams. If you don't have these files, you can download them quickly from the NICMOS website:

<http://www.stsci.edu/hst/nicmos/calibration/reffiles>

Then, start `idl` and put all the necessary routines in your `idl` path (these are in the `lib` dir):

```
IDL> !path = 'lib:' + !path
```

Then, run the task like this:

```
IDL> saa_clean, 'n8g6cnh3q_cal1.fits', 'n8g6cnh3q_cal2.fits', /tv, /fit
```

This will produce lots of text output and a bunch of plots on the screen. Use `"/ps"` to make postscript versions of these instead, or leave both off for quiet ops. For most files you will also want to leave off the `/fit` option as well and just use the default scale factor. You should also see a nice side-by-side of the input image and the corrected image.

Now run `pedsky` again on the output like this:

```
ni> pedsky n8g6cnh3q_cal2.fits n8g6cnh3q_cal3.fits salgori="quick"
```

The final end product is now called `n8g6cnh3q_cal3.fits`.

You can compare the stats of the two with various `iraf` routines, or use the `gaussfithist.pro` routine in the `lib` dir. Say, something like this:

```
IDL> fits_read, 'n8g6cnh3q_cal.fits', cal
IDL> fits_read, 'n8g6cnh3q_cal3.fits', cal3
IDL> exptime=float(getkeyval('n8g6cnh3q_cal.fits', 'EXPTIME'))
IDL> a=gaussfithist(cal*exptime, gain=5.4)
IDL> b=gaussfithist(cal3*exptime, gain=5.4)
IDL> print, 'total noise reduction in percent ', (1.0-b(2)/a(2))*100.
```


Note that you might have a negative sky background after running these tasks. This is just a zeropoint offset in the NREAD=25 ACCUM mode dark used on the post-saa darks...something I will fix as soon as the tool is done. A simple constant sky subtraction on the final product should take care of this before any drizzling.

There is a much more info in the doc/post-saa.ps file. This is an Instrument Science Report describing the algorithm and the tool.

good luck!

Eddie Bergeron (8/14/03)

IDL source code for saa_clean.pro:

```
pro saa_clean,infile,outfile,fit=fit,tv=tv,saainage=saainage,scalefac=scalefac,noflat=noflat,modeflat=modeflat,ps=ps
```

```
final=(-1)
saaper=(-1)
```

```
outroot=strmid(outfile,0,9)
```

```
; read in the image
orig=getset(infile)
cal=reform(orig(*,*,0,0))
```

```
calexptime=float(getkeyval(infile,'EXPTIME'))
```

```
; get the name of the saa_asn table, if there is one
saa_asn=strcompress(getkeyval(infile,'SAA_DARK'))
; if table is N/A, then exit
if strpos(saa_asn,'N/A') ne -1 then begin
print,'This data was not taken in an SAA-impacted orbit. No correction needed. Exiting.'
goto,n1
endif
```

```
print
; read table to get the names of the two SAA dark images
saa_asn=strlowercase(strcompress(saa_asn/rem))+ '_asn.fits'
print,'Name of SAA _asn table is: '+saa_asn
fits_read,saa_asn,crud
ss=strcompress(strlowercase(string(crud)))
safiles=ss(0:1)
for i=0,1 do begin
; fix a bug with tedit if it was used
if strpos(safiles(i),'exp-targ') ne -1 then safiles(i)=strmid(safiles(i),0,strpos(safiles(i),'exp-targ')-1)
safiles(i)=safiles(i)+'_raw.fits'
endfor
print,'SAA dark files: '
print,safiles
```

```
n=n_elements(safiles)
exptime=fltarr(n)
cam=fltarr(n)
saa_time=fltarr(n)
im=fltarr(256,256,2,n)
```

```
; get some keywords
for i=0,n-1 do begin
exptime(i)=float(getkeyval(safiles(i),'EXPTIME'))
cam(i)=float(getkeyval(safiles(i),'CAMERA'))
saa_time(i)=float(getkeyval(safiles(i),'SAA_TIME'))
endfor
```

```
;if cam(0) eq 1 then saadark='/data/joecool7/eddie/saa_clean/c1_saadarkref_drk.fits'
;if cam(0) eq 2 then saadark='/data/joecool7/eddie/saa_clean/c2_saadarkref_drk.fits'
;if cam(0) eq 3 then saadark='/data/joecool7/eddie/saa_clean/c3_saadarkref_drk.fits'
;fits_read,saadark,dark,exten=1 ; This is the dark for an SAA image. Its just an ACCUM-mode dark.
; ; The first sci exten has the matching darktime by definition here.
; ; see /home/bergeron/idl/nicmos/make_accum_ref.pro for details
```

```
; Or try using the _tdd.fits file from the image header to make a dark (aka synthetic ACCUM dark)
```

```

; seems to work pretty well - better than the observed ACCUM darks actually
if cam(0) eq 1 then tddfile='m9b12115n_tdd.fits'
if cam(0) eq 2 then tddfile='m9b12116n_tdd.fits'
if cam(0) eq 3 then tddfile='m9b12117n_tdd.fits'

nreads=25.0; for the post-SAA case
shade=makedark_from_tdd(tddfile,'step256',77.1,/shadeonly)
fits_read,tddfile,ampglow,exten=2
fits_read,tddfile,lindark,exten=1
dark=(shade(*,*,0)/nreads)+(ampglow*nreads*0.86)+(lindark*256.0)

; dark subtract the two SAA dark images
for i=0,n-1 do begin
fits_read,saafiles(i),t
im(*,*,i)=(t-dark)/exptime(i); dark subtracted, and in DN/s
endfor

; read the bad pix map so these can be excluded from the fit
badfile=strcompress(string(getkeyval(saafiles(0),'MASKFILE'),/rem)
if strmid(badfile,0,5) eq 'nref$' then badfile=strmid(badfile,5)
fits_read,badfile,bad,exten=3

im1=im(*,*,0)
im2=im(*,*,1)

off1=0.07
off2=0.075

if cam(0) eq 1 then deffac=0.56
if cam(0) eq 2 then deffac=0.54
if cam(0) eq 3 then deffac=0.54

if not keyword_set(scalefac) then scalefac=deffac
print,'Default scale factor is '+strmid(strcompress(string(deffac),/rem),0,4)
if keyword_set(scalefac) then print,'User has specified a scale factor of: '+strcompress(string(scalefac),/rem)

; if requested, fit for the scale factor. Otherwise just use the default or the specified scalefac
if keyword_set(fit) then begin

;; first pass *****
u=where(im1 gt -0.1 and im2 gt -0.1 and im1 lt 0.5 and im2 lt 0.5)
p=poly_fit(im1(u),im2(u),1,/double)
uu=where(im2(u) lt poly(im1(u),p)+off2 and im2(u) gt poly(im1(u),p)-off1)

; instead of first pass, use deffac to clip outliers
u=where(im1 gt -0.1 and im2 gt -0.1 and im1 lt 0.5 and im2 lt 0.5 and bad eq 0)
uu=where(im2(u) lt poly(im1(u),[0.0,deffac])+off2 and im2(u) gt poly(im1(u),[0.0,deffac])-off1)

; second pass *****
; additional weighting to give the high countrate pixels a chance in shallow passes
pp=poly_fit(im1(u(u)),im2(u(u)),1,/double,measure_err=1./sqrt(im1(u(u))^2+im2(u(u))^2))
uu=where(im2(u) lt poly(im1(u),pp)+off2 and im2(u) gt poly(im1(u),pp)-off1)

if keyword_set(tv) or keyword_set(ps) then begin
!p.multi=0
device,decomp=0
loadct,0
tvlct,r,g,b,/get
g(254)=0
b(254)=0
b(253)=0
r(253)=0
r(252)=0
g(252)=0
tvlct,r,g,b

if not keyword_set(ps) then window,0

if keyword_set(ps) then begin
set_plot,'ps'
device,/color,bits=8,xoff=0.25,yoff=4.5,xs=8.0,ys=6.0,/inches,filename=outroot+'_compfac.ps'
!p.charsize=1.2
!p.charthick=2
!p.thick=5
!x.thick=6
!y.thick=6
!p.font=0

```

```

endif

plot,im1(u),im2(u),psym=3,xst=1,yst=1,xra=[-0.1,0.5],yra=[-0.1,0.5],xtitle='Countrate in im1 (DN/s)',ytitle='Countrate in im2 (DN/s)',title='Post-SAA darks, com-
parrison of 1st and 2nd image',/nodata
plots,[0,0],[-0.1,0.5],lines=3,color=130
plots,[-0.1,0.5],[0,0],lines=3,color=130
u1=where(im2(u) gt poly(im1(u),pp)+off2)
u2=where(im2(u) lt poly(im1(u),pp)-off1)
oplot,im1(u(u1)),im2(u(u1)),psym=1
oplot,im1(u(u2)),im2(u(u2)),psym=1

oplot,im1(u(uu)),im2(u(uu)),psym=1,color=253,thick=1
oplot,(findgen(100)*0.06)-0.1,poly((findgen(100)*0.06)-0.1,pp),color=254
oplot,(findgen(100)*0.06)-0.1,poly((findgen(100)*0.06)-0.1,pp)-off1,color=254,lines=3
oplot,(findgen(100)*0.06)-0.1,poly((findgen(100)*0.06)-0.1,pp)+off2,color=254,lines=3
xyouts,0.35,0.02,'Cosmic Rays in im1',/data,color=252
xyouts,0.02,0.2,'Cosmic Rays in im2',/data,orient=90,color=252
xyouts,0.35,poly(0.35,pp)+0.015,'slope = '+strcompress(string(pp(1)),/rem),orient=pp(1)*45.,color=254,/data
oplot,(findgen(100)*0.06)-0.1,poly((findgen(100)*0.06)-0.1,[0.0,deffac]),color=130
xyouts,0.2,poly(0.2,[0.0,deffac])+0.015,'slope = '+strmid(strcompress(string(deffac),/rem),0,4)+' fiducial',color=130,orient=45/2.,/data

if keyword_set(ps) then begin
device,/close
set_plot,'x'
!p.charsize=1
!p.charthick=1
!p.thick=1
!x.thick=1
!y.thick=1
!p.font=(-1)
endif

endif

scalefac=pp(1)
print,'Scale factor from fit is: '+strcompress(string(scalefac),/rem)+' , using this value....'

; end of fit loop
endif
print

; pedskyish...

m=fltarr(4)
m(0)=median(im1(10:117,10:117))
m(1)=median(im1(10+128:117+128,10:117))
m(2)=median(im1(10+128:117+128,10+128:117+128))
m(3)=median(im1(10:117,10+128:117+128))
m=m-median(m); subtract off the median countrate
im1(0:127,0:127)=im1(0:127,0:127)-m(0)
im1(0+128:127+128,0:127)=im1(0+128:127+128,0:127)-m(1)
im1(0+128:127+128,0+128:127+128)=im1(0+128:127+128,0+128:127+128)-m(2)
im1(0:127,0+128:127+128)=im1(0:127,0+128:127+128)-m(3)

m=fltarr(4)
m(0)=median(im2(10:117,10:117))
m(1)=median(im2(10+128:117+128,10:117))
m(2)=median(im2(10+128:117+128,10+128:117+128))
m(3)=median(im2(10:117,10+128:117+128))
m=m-median(m); subtract off the median countrate
im2(0:127,0:127)=im2(0:127,0:127)-m(0)
im2(0+128:127+128,0:127)=im2(0+128:127+128,0:127)-m(1)
im2(0+128:127+128,0+128:127+128)=im2(0+128:127+128,0+128:127+128)-m(2)
im2(0:127,0+128:127+128)=im2(0:127,0+128:127+128)-m(3)

; special handling of middle col/row
if cam(0) lt 3 then begin
im1(127,*)=im1(127,*)-median(im1(127,*)-im1(126,*))
im2(127,*)=im2(127,*)-median(im2(127,*)-im2(126,*))
endif
if cam(0) eq 3 then begin
im1(*,127)=im1(*,127)-median(im1(*,127)-im1(*,126))
im2(*,127)=im2(*,127)-median(im2(*,127)-im2(*,126))
endif

; combine the two images using the scale factor
;saaper=( im1+(im2/scalefac) )/2.; average

```

```

; combine with weight in proportion to the scale factor
;wf1=1.0/scalefac
;wf2=scalefac
;sum=wf1+wf2
;wf1=wf1/sum
;wf2=wf2/sum
wf1=0.7
wf2=0.3
saaper=( (im1*wf1) + ((im2/scalefac)*wf2) ); weighted average

; cr thresholding
a=im1-(im2/scalefac)
u1=where(a gt 0.2)
saaper(u1)=im2(u1)/scalefac
a=(im2/scalefac)-im1
u2=where(a gt 0.2)
saaper(u2)=im1(u2)

if keyword_set(tv) or keyword_set(ps) then begin
if not keyword_set(ps) then window,1,xs=800,ys=280
!p.charsize=1.8

if keyword_set(ps) then begin
set_plot,'ps'
device,/color,bits=8,xoff=0.25,yoff=4.5,xs=8.0,ys=8.0/2.85714,/inches,filename=outroot+'_saamap.ps'
!p.charsize=1.2
!p.charthick=2
!p.thick=5
!x.thick=6
!y.thick=6
!p.font=0
loadct,0
tvlct,r,g,b,/get
r=reverse(r)
g=reverse(g)
b=reverse(b)
tvlct,r,g,b
endif

plot,findgen(100),/nodata,xra=[0,799],yra=[0,279],xmar=[0,0],ymar=[0,0],xst=5,yst=5
tv,bytsc1(im1,-0.01,0.3),0,0,/data,xs=256,ys=256
tv,bytsc1(im2/scalefac,-0.01,0.3),260,0,/data,xs=256,ys=256
tv,bytsc1(saaper,-0.01,0.3),540,0,/data,xs=256,ys=256
xyouts,115,260,'im1',color=250
xyouts,80+260,260,'im2/scalefac',color=250
xyouts,590,260,'final SAA persistence',color=250

!p.charsize=1
if keyword_set(ps) then begin
device,/close
set_plot,'x'
!p.charsize=1
!p.charthick=1
!p.thick=1
!x.thick=1
!y.thick=1
!p.font=(-1)
endif
endif

; saaper is the persistence image
if keyword_set(saaimage) then writefits,saaimage,saaper

; now, correct the input image using the new saa persistence image...
print,'Iterating for SAA-to-data scale factor...'

; If desired, flatfield the saa persistence image using the same flat applied to
; the image being corrected.

if not keyword_set(noflat) then begin
flatfile=strcompress(getkeyval(infile,'FLATFILE'),/rem)
flatdone=strcompress(getkeyval(infile,'FLATDONE'),/rem)
if strpos(flatfile,'nref$') ne -1 then flatfile=strmid(flatfile,strpos(flatfile,'$')+1)

if flatdone eq "PERFORMED" then begin
fits_read,flatfile,flat

```

```

        mm=median(saaper)
; use this for the mode - or peak of the histogram - instead of the median. Mode is a little better in the final result after a second pedsky, but leaves a little more pedestal
in the output of this routine.
if keyword_set(modeflat) then begin
a=gaussfithist(saaper*exptime(0),sigma=100)
mm=a(1)/exptime(0)
endif
        saaper=((saaper-mm)*flat)+mm
    endif
endif

; best to mask bad pix if possible. 0=good, nonzero=bad
badfile=strcompress(getkeyval(infile,'MASKFILE'),/rem)
if strpos(badfile,'nref$') ne -1 then badfile=strmid(badfile,strpos(badfile,'$')+1)
fits_read,badfile,bad,exten=3
mask=fltarr(256,256)
badmask=fltarr(256,256)+1.0
u=where(bad ne 0)
mask(u)=1
badmask(u)=0; for use later so we don't apply any correction using bad pixels
; cross
mask(127,*)=1
mask(*,127)=1
; borders
mask(0:3,*)=1
mask(253:255,*)=1
mask(*,0:15)=1
mask(*,253:255)=1

;save,mask,filename='mask.sav

if keyword_set(tv) or keyword_set(ps) then begin

if not keyword_set(ps) then window,2,xs=512,ys=700

; this is where it gets interesting

; the 500. here is just to get the data in range for the gaussfithist routine.
; notice it gets divided back out afterwards.
aaa=gaussfithist(saaper*500.,gain=1) ; fit for the peak and width of the saaper image to select the hi/lo threshold
tsigma=3.5
thresh = (aaa(1)+(aaa(2)*tsigma))/500.
print,'Using hi/low threshold of '+strmid(strcompress(string(thresh),/rem),0,5)+' DN/s in the saaper image...'
; thresh=0.1

; clip high guys
u_hi=where(saaper gt thresh)

; clip low guys
u_lo=where(saaper le thresh)

; now iterate twice - once for hi and once for lo

fitmask=mask
fitmask(u_lo)=1; mask out low guys to fit the hi ones
if keyword_set(ps) then begin
loadct,0
p1_hi=piterate(cal*calexptime,saaper*calexptime,gain=5.4,step=0.008,/fit,mask=fitmask,range=0.25,psfile=outroot+'_scalefit_hi.ps',/plot)
endif
if not keyword_set(ps) then p1_hi=piterate(cal*calexptime,saaper*calexptime,/plot,gain=5.4,step=0.008,/fit,mask=fitmask,range=0.25)

wait,1

fitmask=mask
fitmask(u_hi)=1; now mask out hi guys to fit the lo ones
if keyword_set(ps) then begin
loadct,0
p1_lo=piterate(cal*calexptime,saaper*calexptime,gain=5.4,step=0.008,/fit,mask=fitmask,range=0.25,psfile=outroot+'_scalefit_lo.ps',/plot)
endif
if not keyword_set(ps) then p1_lo=piterate(cal*calexptime,saaper*calexptime,/plot,gain=5.4,step=0.008,/fit,mask=fitmask,range=0.25)

; if noise reduction it gt 1% then apply, otherwise don't
final=cal

```

```

if p1_hi(1) ge 1.0 then final(u_hi)=cal(u_hi)-(saaper(u_hi)*p1_hi(0)*badmask(u_hi))
if p1_lo(1) ge 1.0 then final(u_lo)=cal(u_lo)-(saaper(u_lo)*p1_lo(0)*badmask(u_lo))

if p1_hi(1) lt 1.0 then begin
print,'***** Noise reduction less than 1%, no correction applied to hi pixels *****'
final(u_hi)=cal(u_hi)
endif
if p1_lo(1) lt 1.0 then begin
print,'***** Noise reduction less than 1%, no correction applied to lo pixels *****'
final(u_lo)=cal(u_lo)
endif

; if the hi reduction is less than 1%, but the lo reduction was more than 1%, at least apply
; the low reduction to the high pixels too
if p1_hi(1) lt 1.0 and p1_lo(1) ge 1.0 then final(u_hi)=cal(u_hi)-(saaper(u_hi)*p1_lo(0)*badmask(u_hi))

if not keyword_set(ps) then window,3,xs=560,ys=300

!p.charsize=1.8

if keyword_set(ps) then begin
set_plot,'ps'
device,/color,bits=8,xoff=0.25,yoff=4.5,xs=8.0,ys=8.0/(560/300.)/inches,filename=outroot+'_before_after.ps'
!p.charsize=1.2
!p.charthick=2
!p.thick=5
!x.thick=6
!y.thick=6
!p.font=0
loadct,0
tv!ct,r,g,b,/get
r=reverse(r)
g=reverse(g)
b=reverse(b)
tv!ct,r,g,b
endif

plot,findgen(100)/nodata,xra=[0,559],yra=[0,299],xmar=[0,0],ymar=[0,0],xst=5,yst=5
tv,bytsc!cal-median(cal),-0.02,0.06,0,0,xs=256,ys=256,/data
tv,bytsc!final-median(final),-0.02,0.06,300,0,xs=256,ys=256,/data
xyouts,80,260,'Input Image',color=250
xyouts,380,260,'Cleaned Image',color=250

!p.charsize=1
if keyword_set(ps) then begin
device,/close
set_plot,'x'
!p.charsize=1
!p.charthick=1
!p.thick=1
!x.thick=1
!y.thick=1
!p.font=(-1)
endif

endif

if not keyword_set(tv) and not keyword_set(ps) then begin
p1=piterate(cal*calxptime,saaper*calxptime,gain=5.4,step=0.008,/fit,mask=mask,range=0.25)

; if noise reduction it gt 1% then apply, otherwise don't
if p1(1) ge 1.0 then final=cal-(saaper*p1(0)*badmask)
if p1(1) lt 1.0 then begin
print,'***** Noise reduction less than 1%, no correction applied *****'
final=cal
endif

endif

; write the output to a fits image, keeping the header and all other
; extensions from the input image.

orig(*,*,0,0)=final
write_multi,outfile,orig,header_from=infile

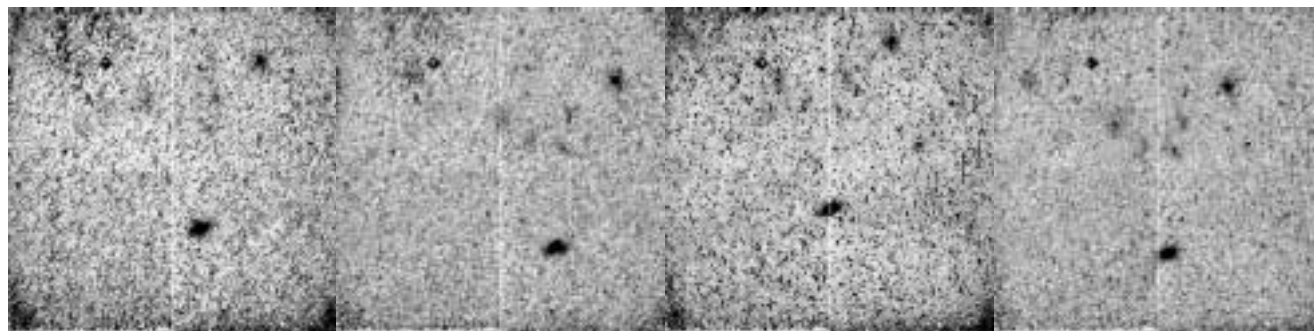
n1:

end

```

Appendix B

During shakeout of the `saa_clean.pro` algorithm, a number of different techniques were tested to decide which provided the optimal noise rejections and if there was any dependence on situation. Presented here are a number of figures using the data from Example 2 in the text. Below each image in the figures is the measured noise and noise reduction as a percentage of the noise measured in the input dataset. There are histograms with gaussian fits included for each of the images, which is where the noise numbers come from in the image figures. The nominal techniques decided upon for the `saa_clean.pro` code are to do a **pedsky** subtraction both before and after running (technically not part of the `saa_clean.pro` routine itself), and to use the median value of the SAACRMAP for normalization before applying the flatfield. In the tests here it was found that the mode provided a very slightly better result globally than the median, however it induced a larger fraction of systematic pattern from the flat (like a pedestal). This induced signal is well removed after the fact by the second run through **pedsky**, but it was decided that in cases where a second (or even first) run of **pedsky** is not performed that it was desirable to minimize any induced pedestalling. A “`/mode`” switch is included in the code in case you want to use it that way. The first four figures (one per page) that follow use the mode (really the location of the peak of a gaussian fit to the histogram), and the second four use the median.



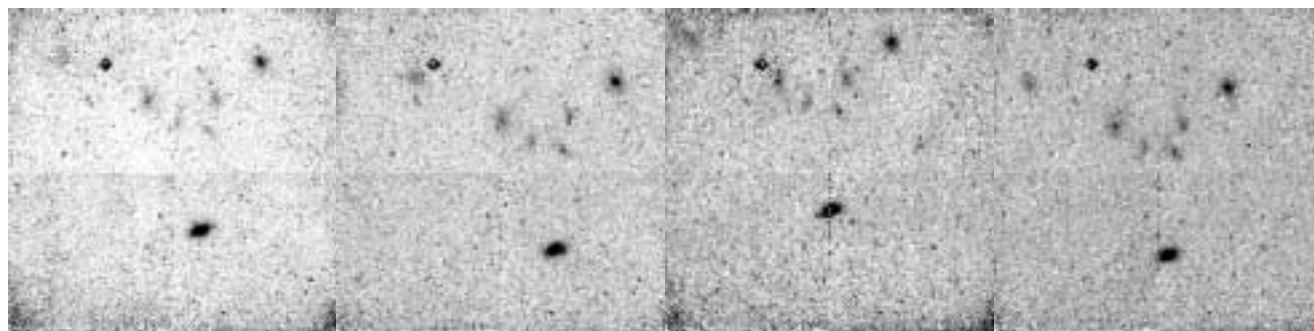
UNCLEAN
WITH PRE-PEDSKY

Noise: 79.4e-

Noise: 55.1e-

Noise: 49.5e-

Noise: 43.5e-



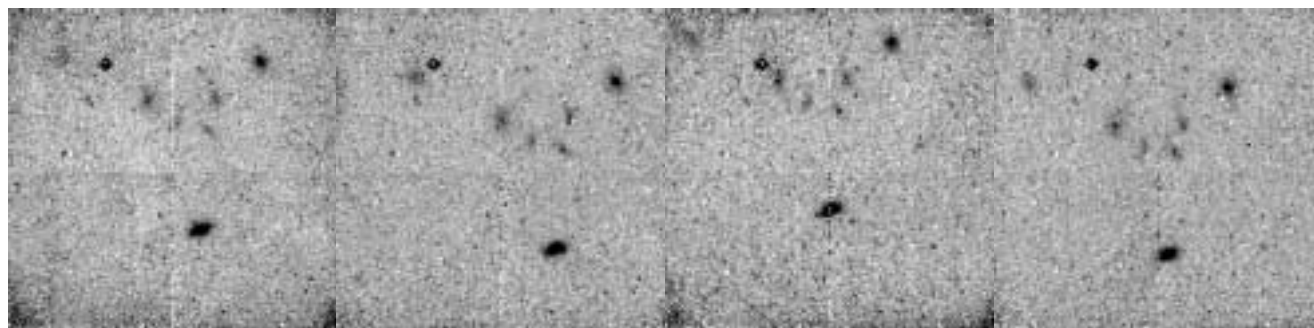
no flat
no post-pedsky

Noise: 43.7e-, Red: 44.8%

Noise: 36.2e-, Red: 34.1%

Noise: 43.0e-, Red: 13.0%

Noise: 35.6e-, Red: 18.2%



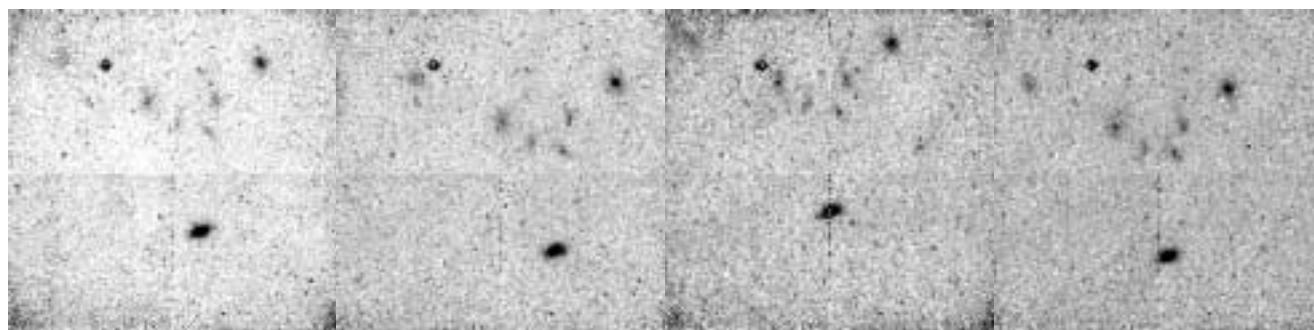
no flat
with post-pedsky

Noise: 42.3e-, Red: 46.6%

Noise: 35.7e-, Red: 35.1%

Noise: 42.8e-, Red: 13.4%

Noise: 35.5e-, Red: 18.4%



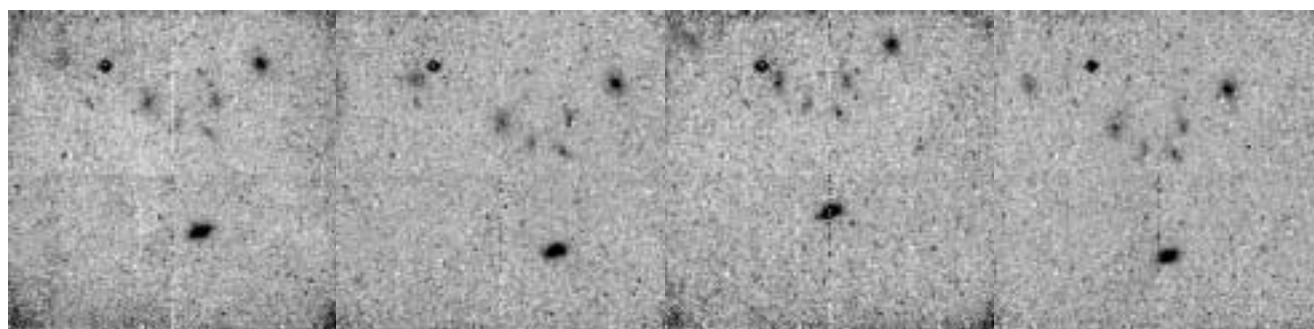
with flat
no post-pedsky

Noise: 43.3e-, Red: 45.4%

Noise: 36.1e-, Red: 34.4%

Noise: 43.0e-, Red: 13.1%

Noise: 35.6e-, Red: 18.1%



with flat
with post-pedsky

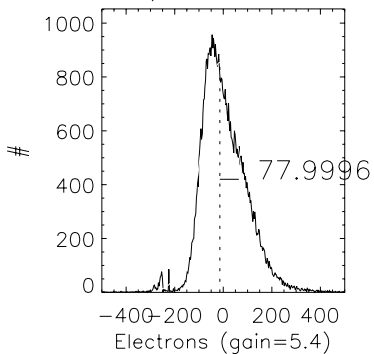
Noise: 42.2e-, Red: 46.8%

Noise: 35.4e-, Red: 35.6%

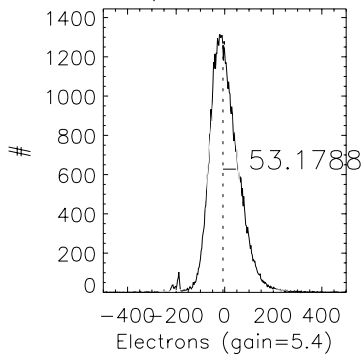
Noise: 42.7e-, Red: 13.5%

Noise: 35.4e-, Red: 18.7%

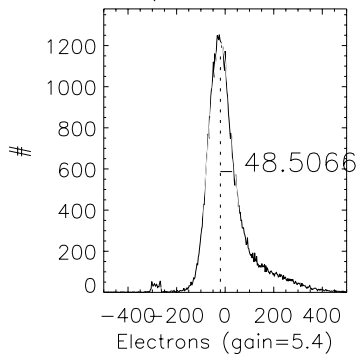
UNCLEAN, WITH PRE-PEDSKY



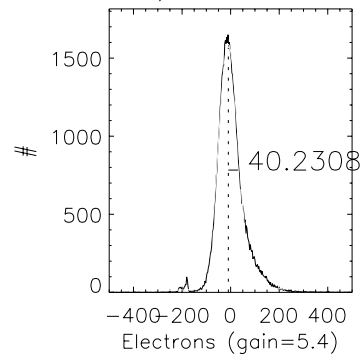
UNCLEAN, WITH PRE-PEDSKY



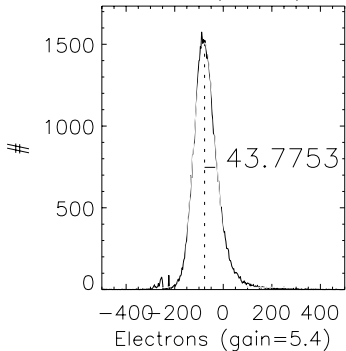
UNCLEAN, WITH PRE-PEDSKY



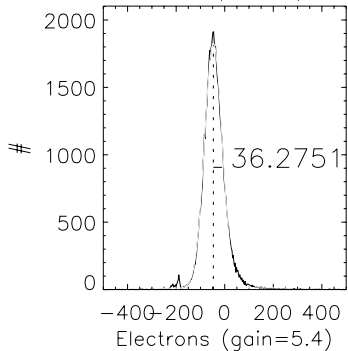
UNCLEAN, WITH PRE-PEDSKY



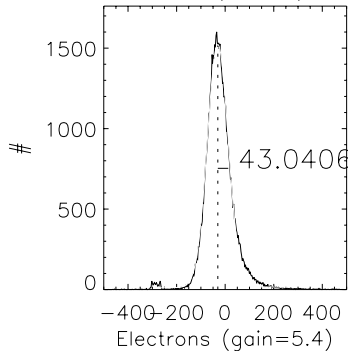
no flat, no post-pedsky



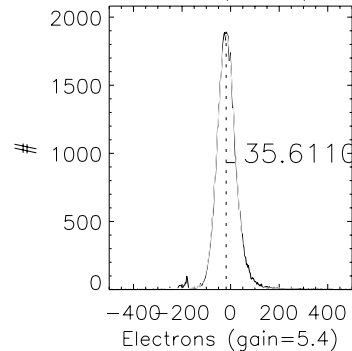
no flat, no post-pedsky



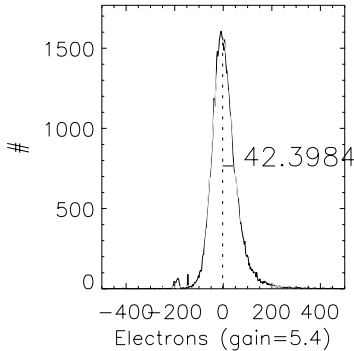
no flat, no post-pedsky



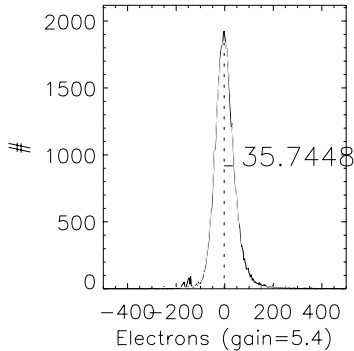
no flat, no post-pedsky



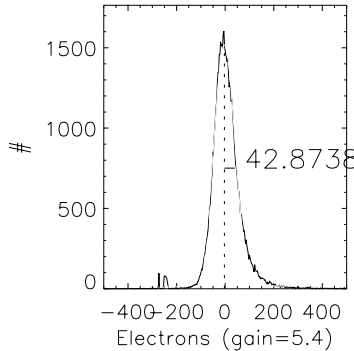
no flat, with post-pedsky



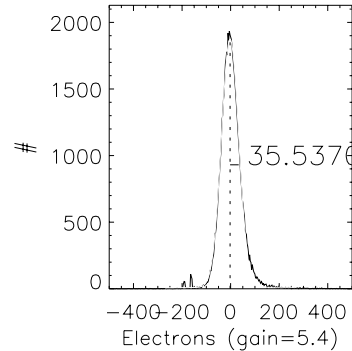
no flat, with post-pedsky



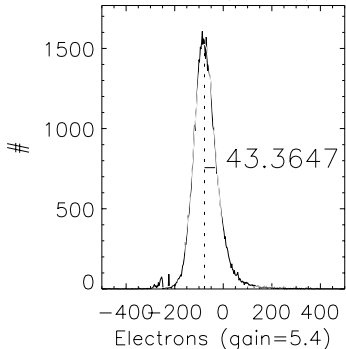
no flat, with post-pedsky



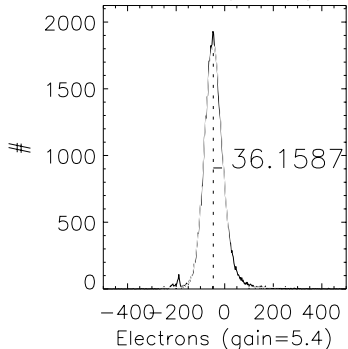
no flat, with post-pedsky



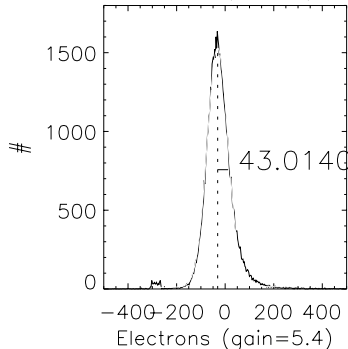
with flat, no post-pedsky



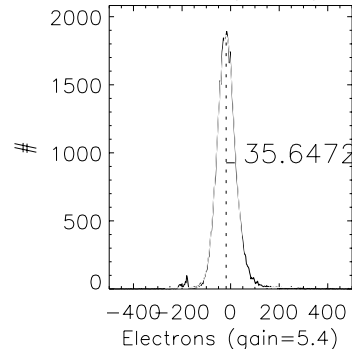
with flat, no post-pedsky



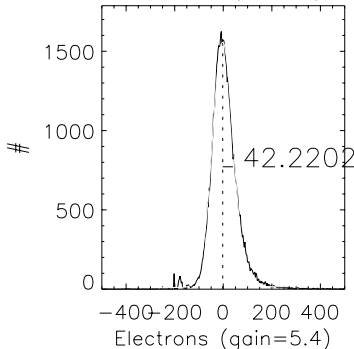
with flat, no post-pedsky



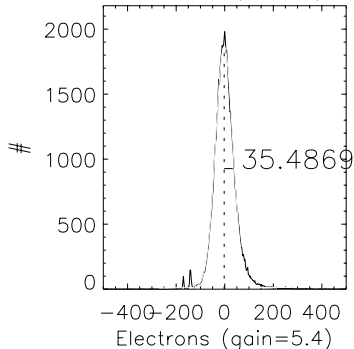
with flat, no post-pedsky



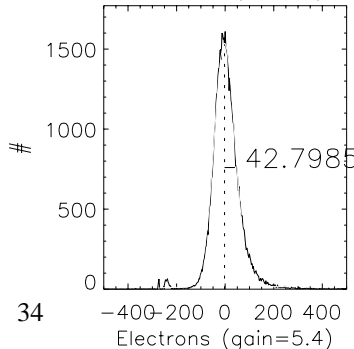
with flat, with post-pedsky



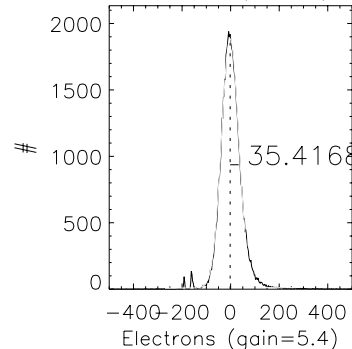
with flat, with post-pedsky

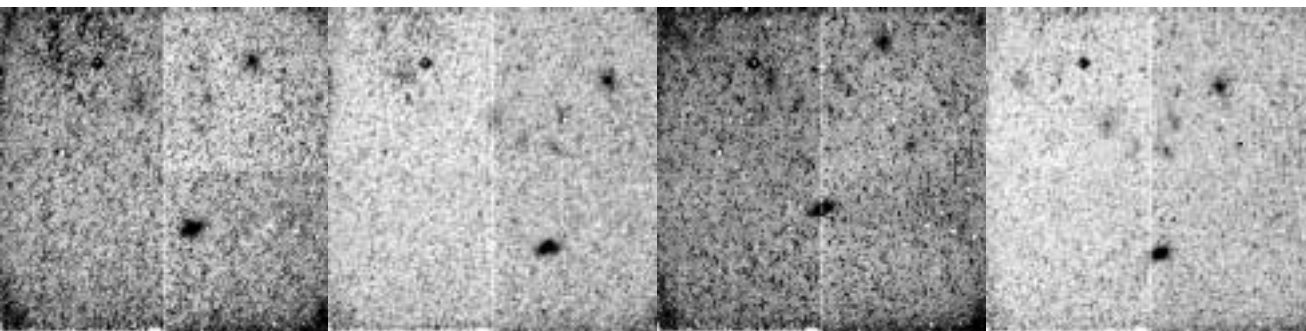


with flat, with post-pedsky



with flat, with post-pedsky





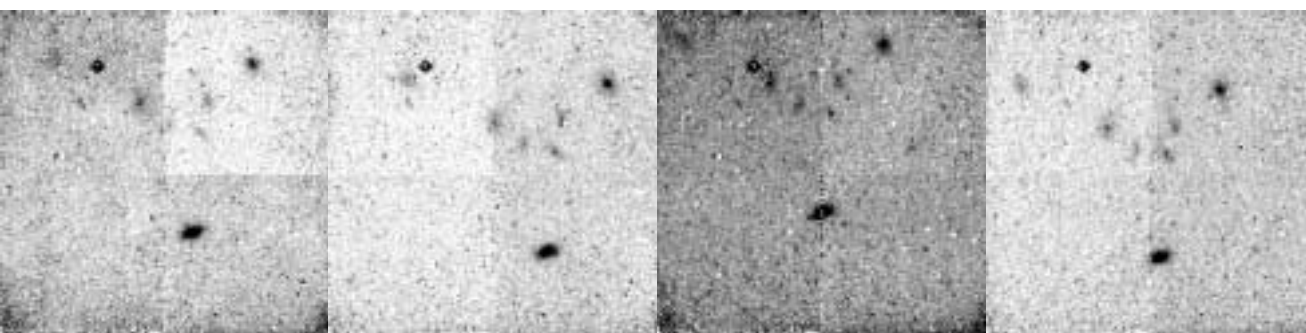
UNCLEAN
NO PRE-PEDSKY

Noise: 77.9e-

Noise: 53.1e-

Noise: 48.5e-

Noise: 40.2e-



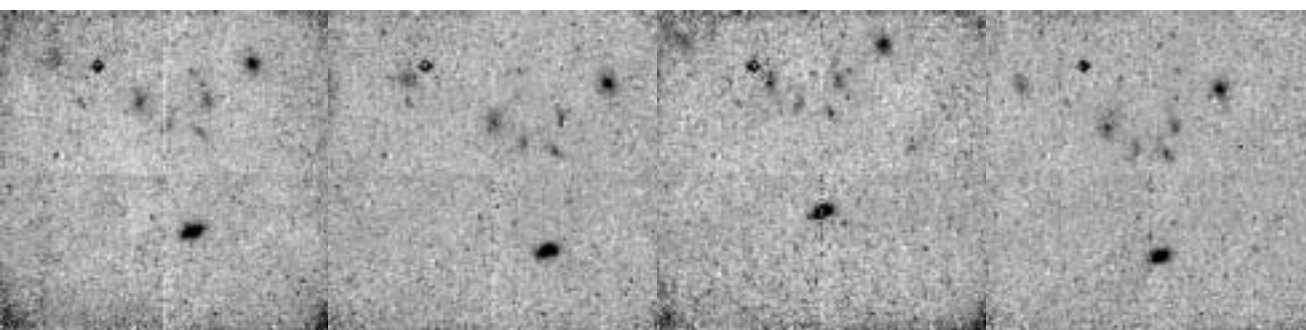
no flat
no post-pedsky

Noise: 48.4e-, Red: 37.9%

Noise: 39.3e-, Red: 25.9%

Noise: 44.5e-, Red: 8.14%

Noise: 38.9e-, Red: 3.21%



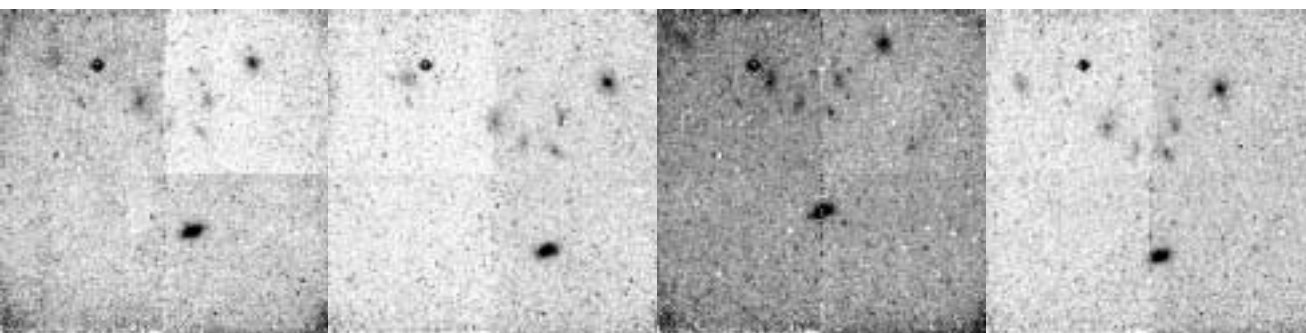
no flat
with post-pedsky

Noise: 42.6e-, Red: 45.3%

Noise: 36.1e-, Red: 32.0%

Noise: 43.2e-, Red: 10.8%

Noise: 35.8e-, Red: 10.9%



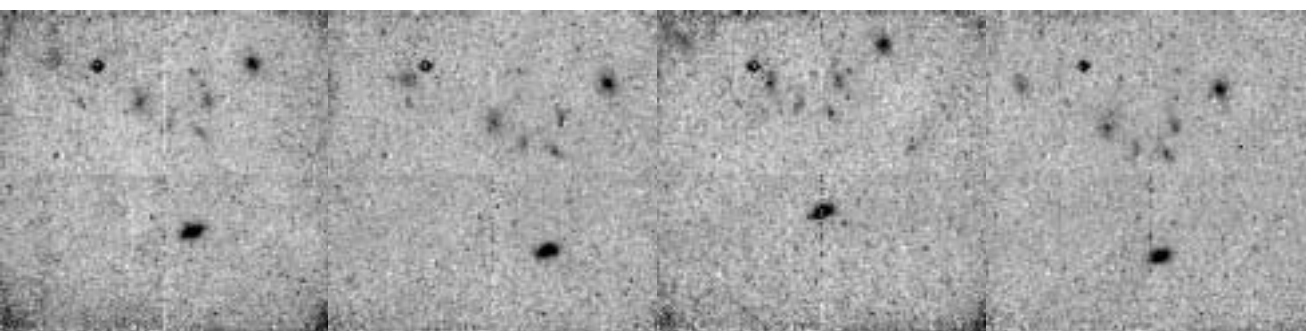
with flat
no post-pedsky

Noise: 47.5e-, Red: 39.0%

Noise: 38.9e-, Red: 26.8%

Noise: 44.4e-, Red: 8.29%

Noise: 38.6e-, Red: 3.92%



with flat
with post-pedsky

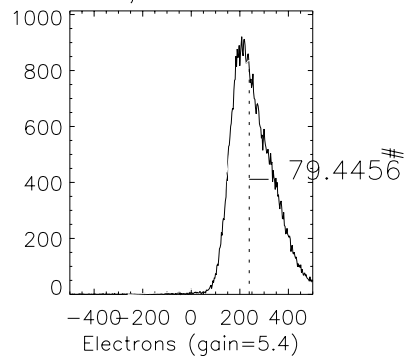
Noise: 42.1e-, Red: 45.9%

Noise: 35.7e-, Red: 32.8%

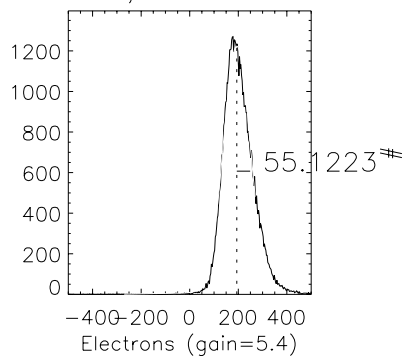
Noise: 43.2e-, Red: 10.9%

Noise: 35.7e-, Red: 11.1%

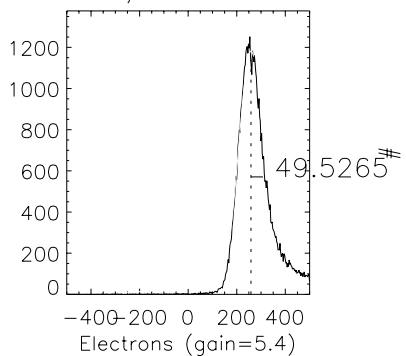
UNCLEAN, NO PRE-PEDSKY



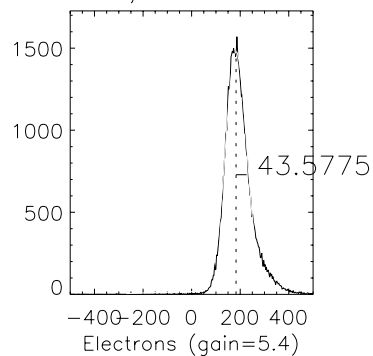
UNCLEAN, NO PRE-PEDSKY



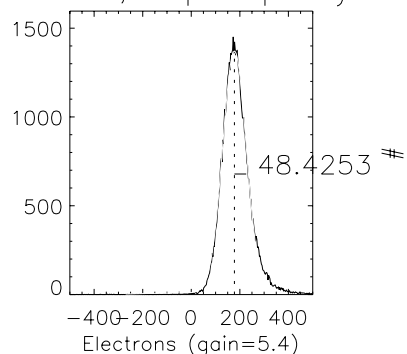
UNCLEAN, NO PRE-PEDSKY



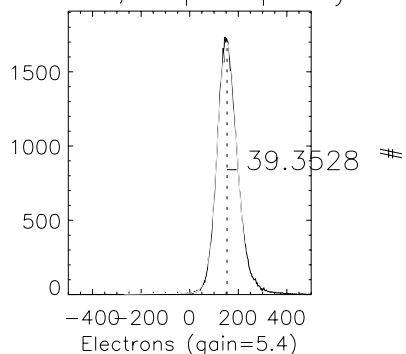
UNCLEAN, NO PRE-PEDSKY



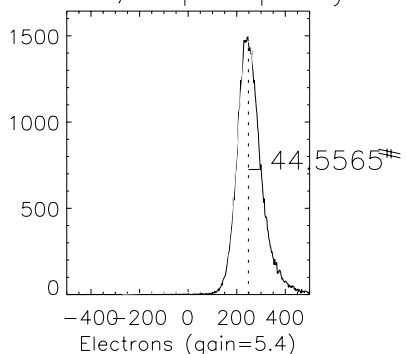
no flat, no post-pedsky



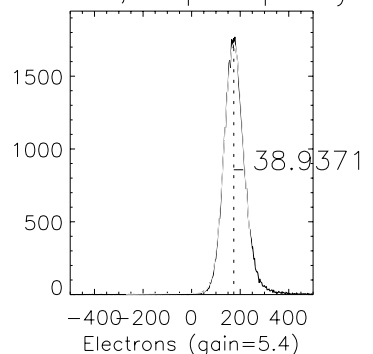
no flat, no post-pedsky



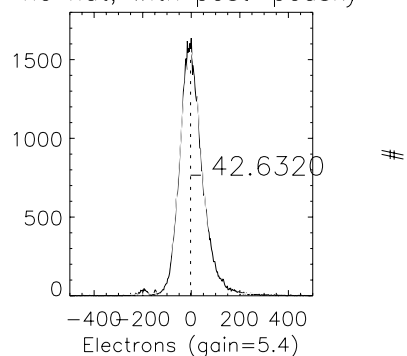
no flat, no post-pedsky



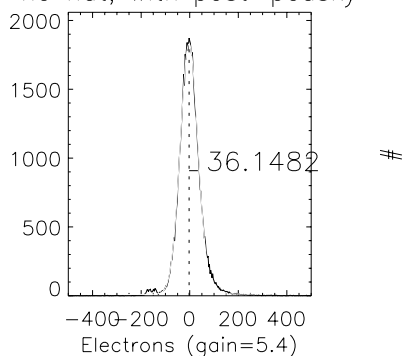
no flat, no post-pedsky



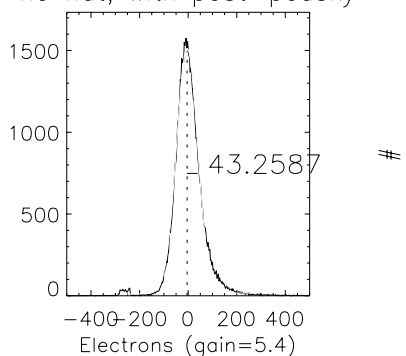
no flat, with post-pedsky



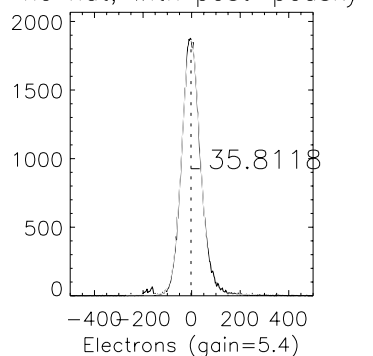
no flat, with post-pedsky



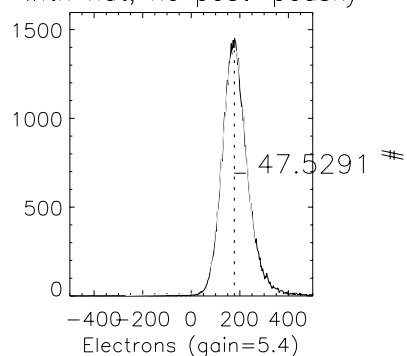
no flat, with post-pedsky



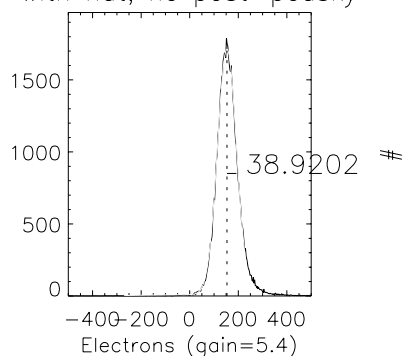
no flat, with post-pedsky



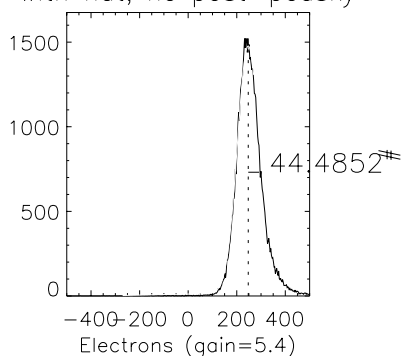
with flat, no post-pedsky



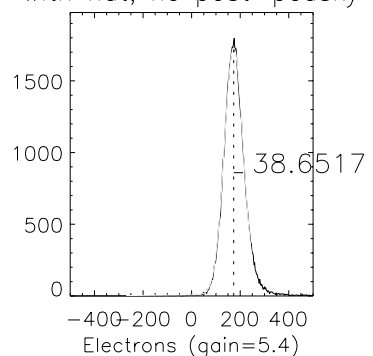
with flat, no post-pedsky



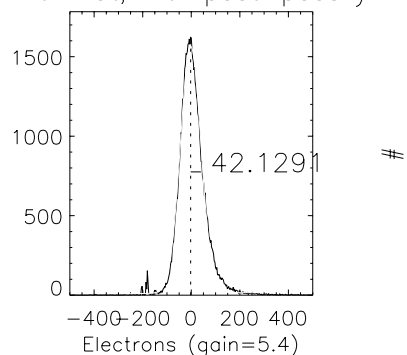
with flat, no post-pedsky



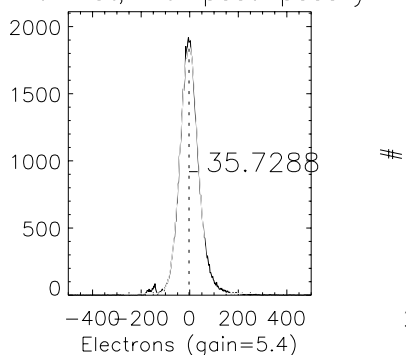
with flat, no post-pedsky



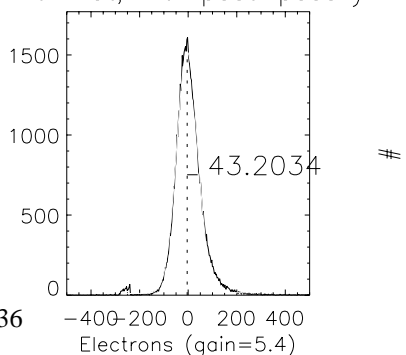
with flat, with post-pedsky



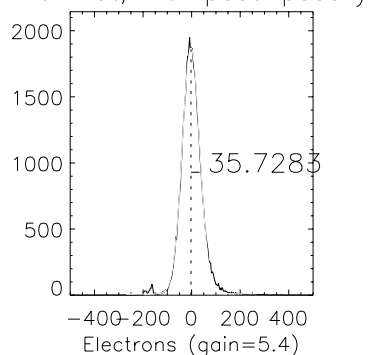
with flat, with post-pedsky

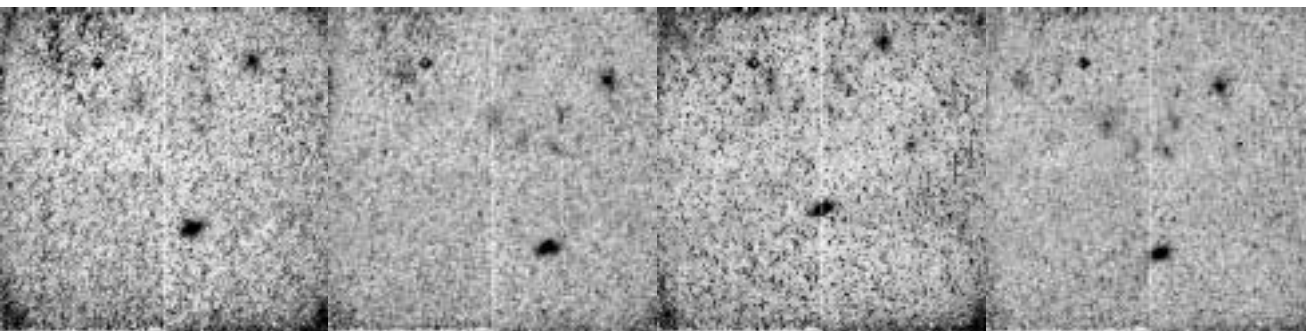


with flat, with post-pedsky



with flat, with post-pedsky





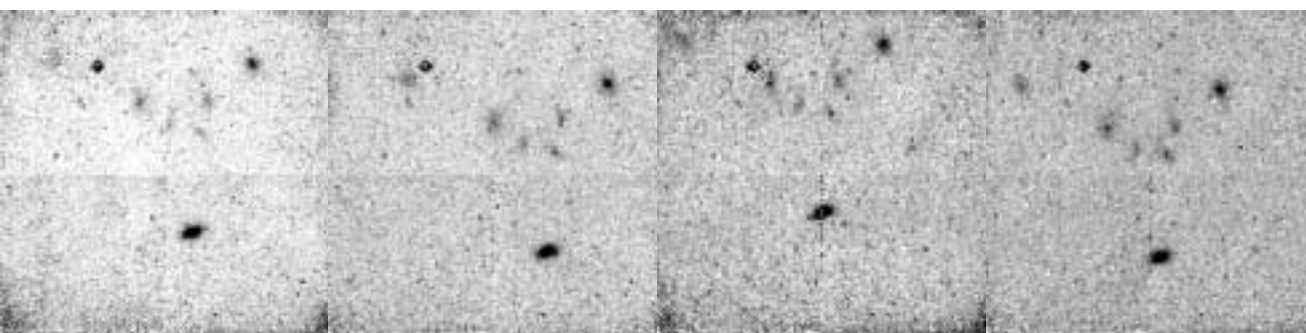
UNCLEAN
WITH PRE-PEDSKY

Noise: 79.4e-

Noise: 55.1e-

Noise: 49.5e-

Noise: 43.5e-



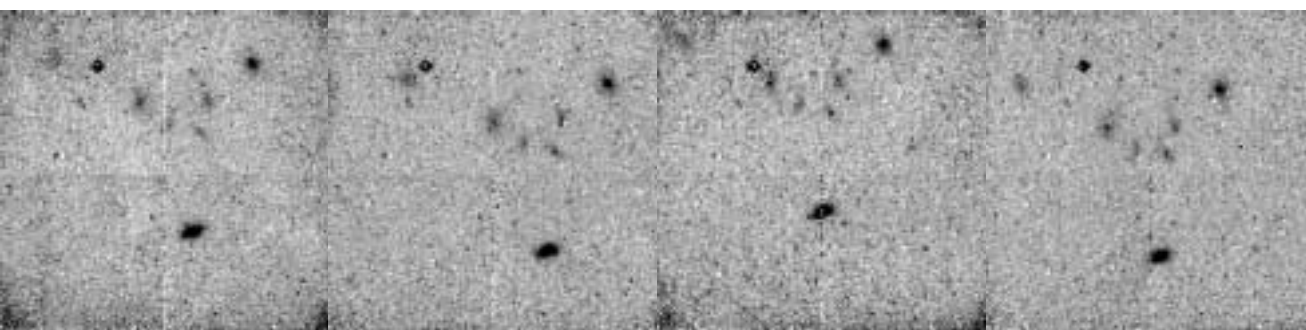
no flat
no post-pedsky

Noise: 43.7e-, Red: 44.8%

Noise: 36.2e-, Red: 34.1%

Noise: 43.0e-, Red: 13.0%

Noise: 35.6e-, Red: 18.2%



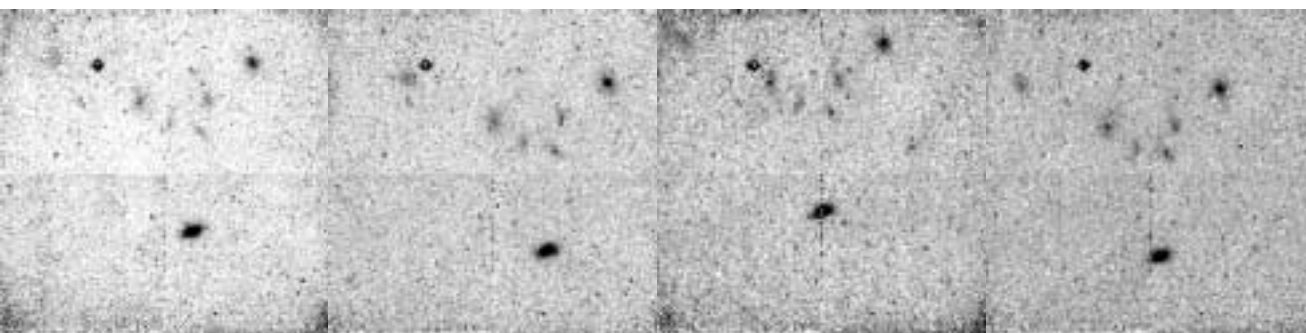
no flat
with post-pedsky

Noise: 42.3e-, Red: 46.6%

Noise: 35.7e-, Red: 35.1%

Noise: 42.8e-, Red: 13.4%

Noise: 35.5e-, Red: 18.4%



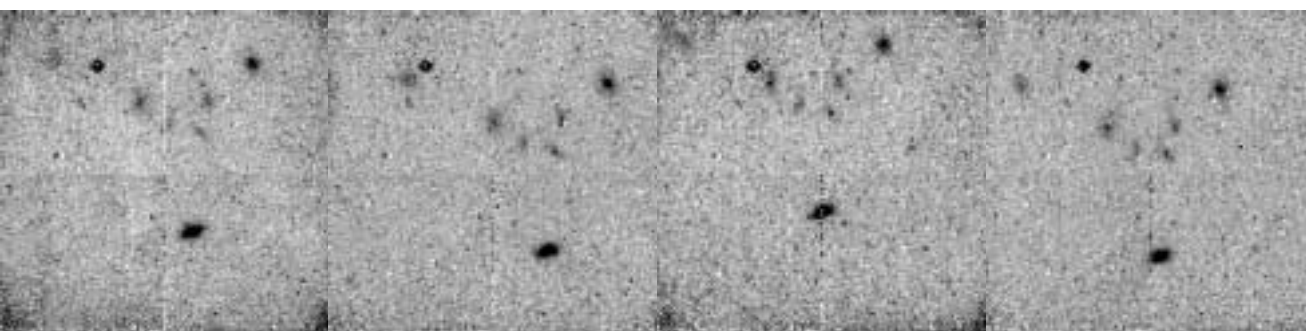
with flat
no post-pedsky

Noise: 43.5e-, Red: 45.2%

Noise: 36.1e-, Red: 34.4%

Noise: 43.2e-, Red: 12.7%

Noise: 35.7e-, Red: 18.0%



with flat
with post-pedsky

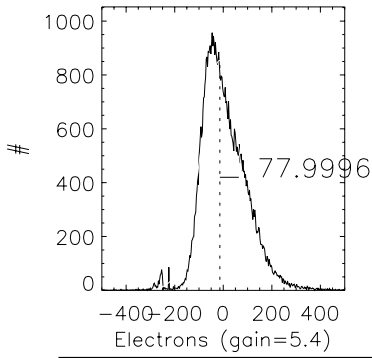
Noise: 42.2e-, Red: 46.7%

Noise: 35.4e-, Red: 35.5%

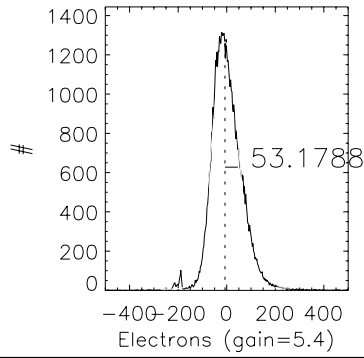
Noise: 42.8e-, Red: 13.4%

Noise: 35.4e-, Red: 18.7%

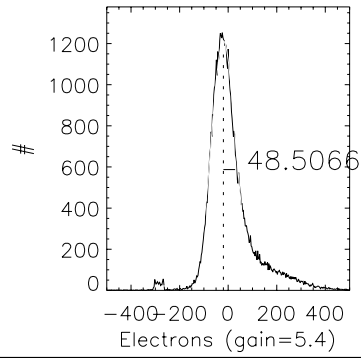
UNCLEAN, WITH PRE-PEDSKY



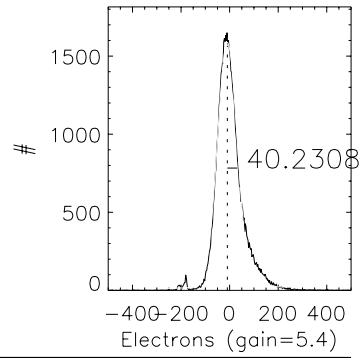
UNCLEAN, WITH PRE-PEDSKY



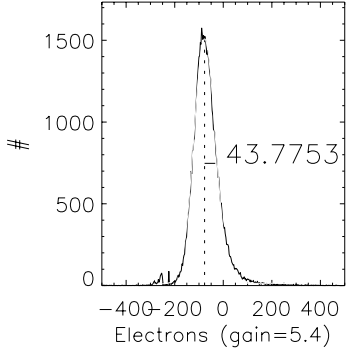
UNCLEAN, WITH PRE-PEDSKY



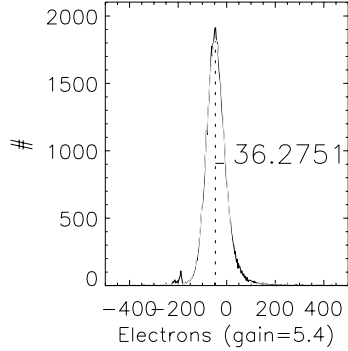
UNCLEAN, WITH PRE-PEDSKY



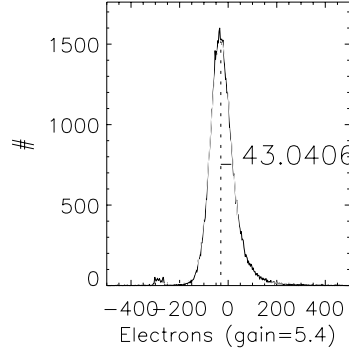
no flat, no post-pedsky



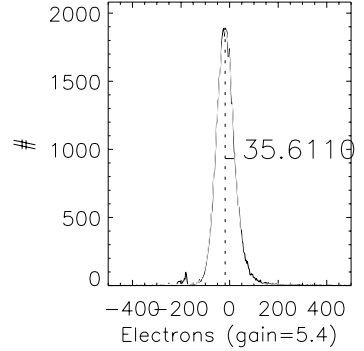
no flat, no post-pedsky



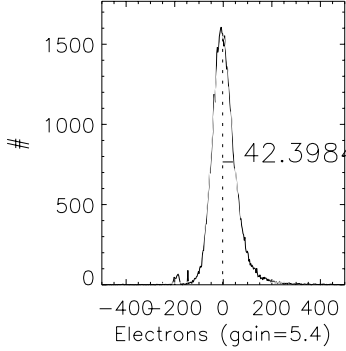
no flat, no post-pedsky



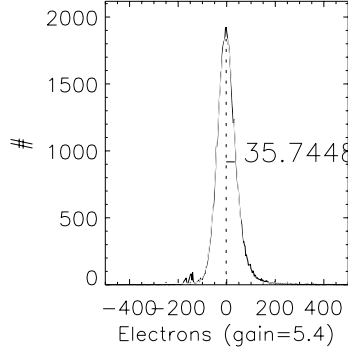
no flat, no post-pedsky



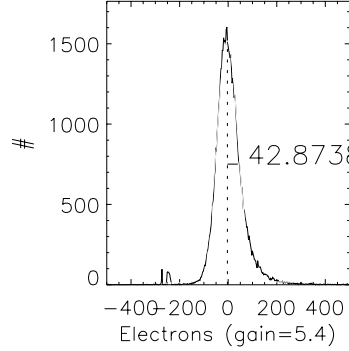
no flat, with post-pedsky



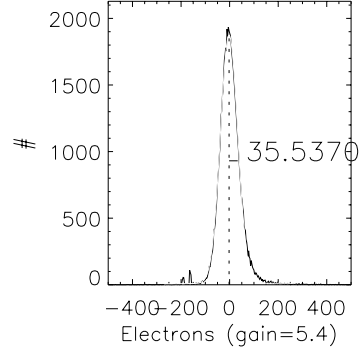
no flat, with post-pedsky



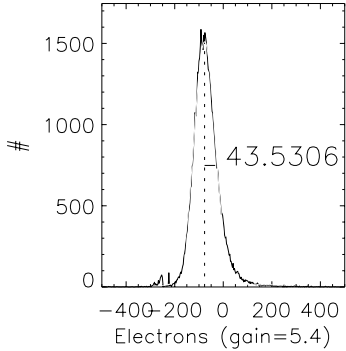
no flat, with post-pedsky



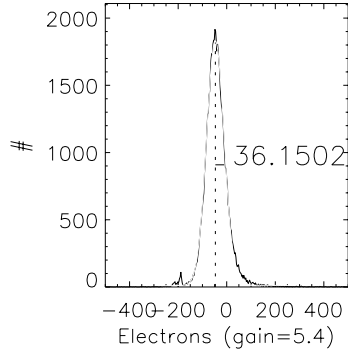
no flat, with post-pedsky



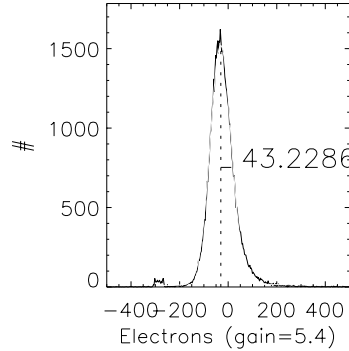
with flat, no post-pedsky



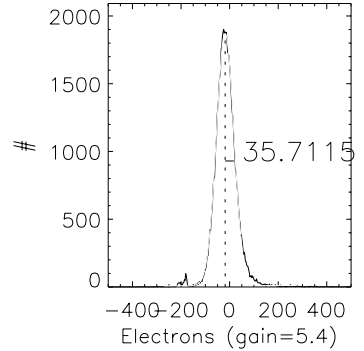
with flat, no post-pedsky



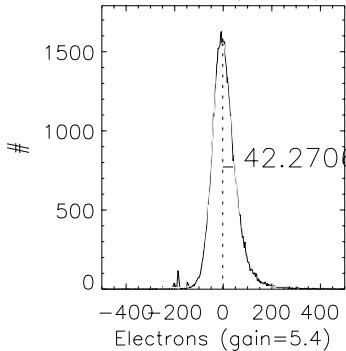
with flat, no post-pedsky



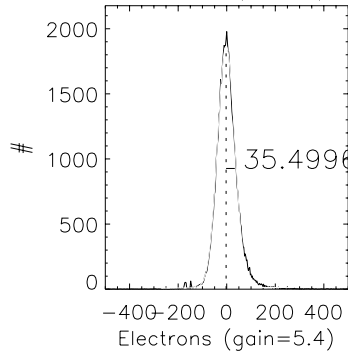
with flat, no post-pedsky



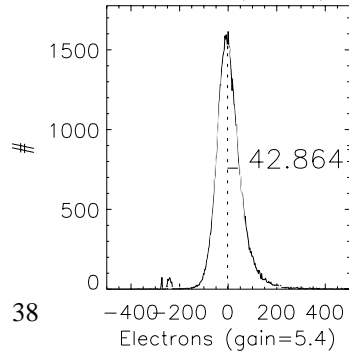
with flat, with post-pedsky



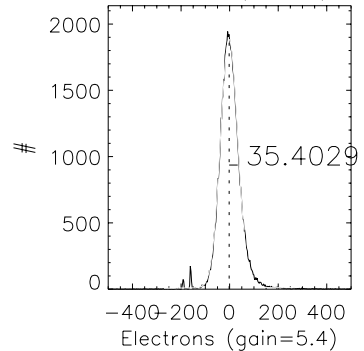
with flat, with post-pedsky

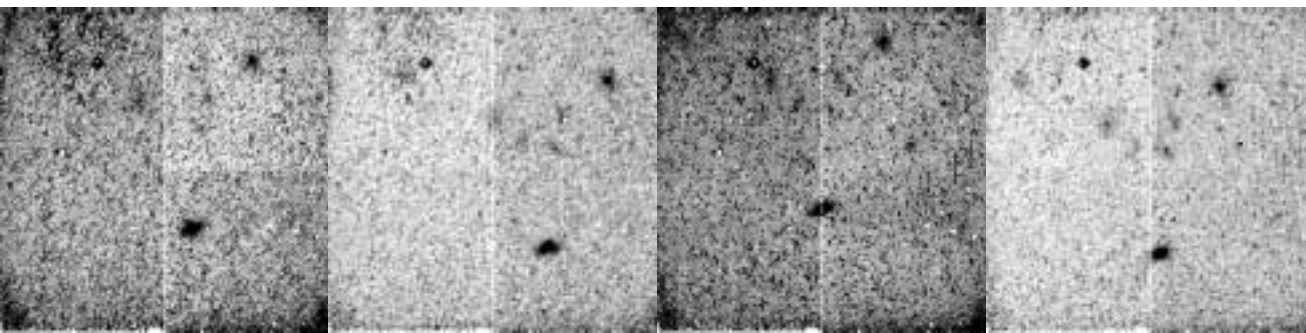


with flat, with post-pedsky



with flat, with post-pedsky





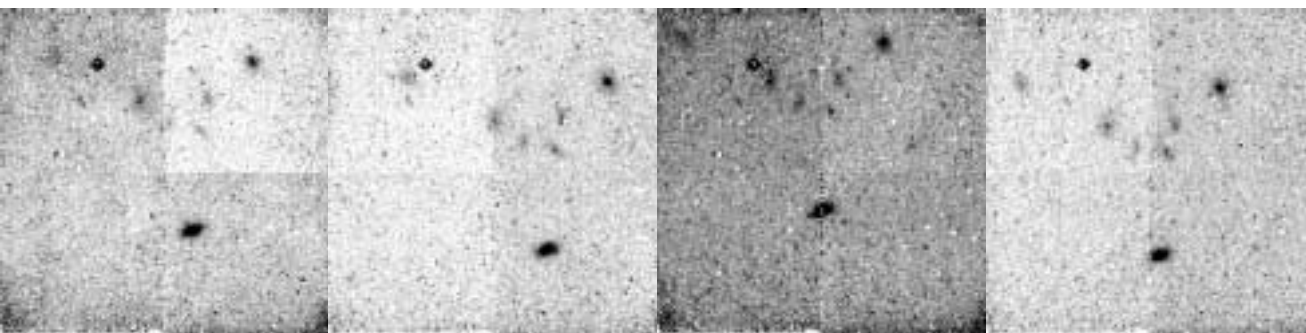
UNCLEAN
NO PRE-PEDSKY

Noise: 77.9e-

Noise: 53.1e-

Noise: 48.5e-

Noise: 40.2e-



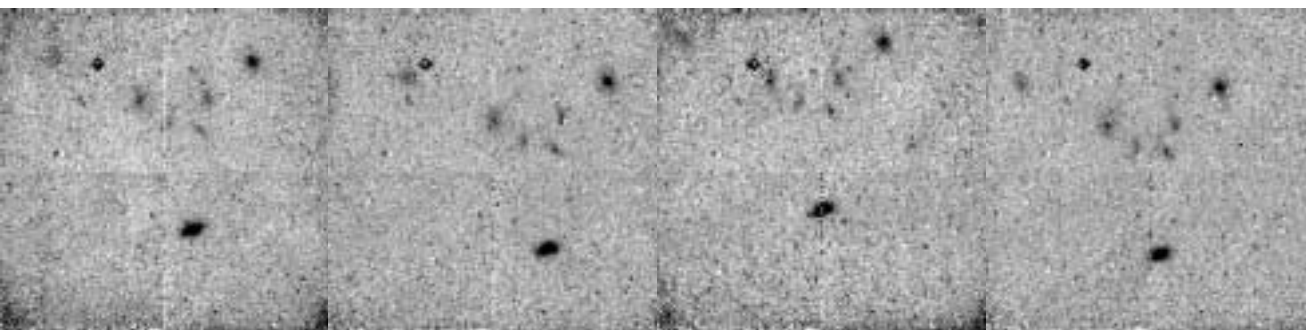
no flat
no post-pedsky

Noise: 48.4e-, Red: 37.9%

Noise: 39.3e-, Red: 25.9%

Noise: 44.5e-, Red: 8.14%

Noise: 38.9e-, Red: 3.21%



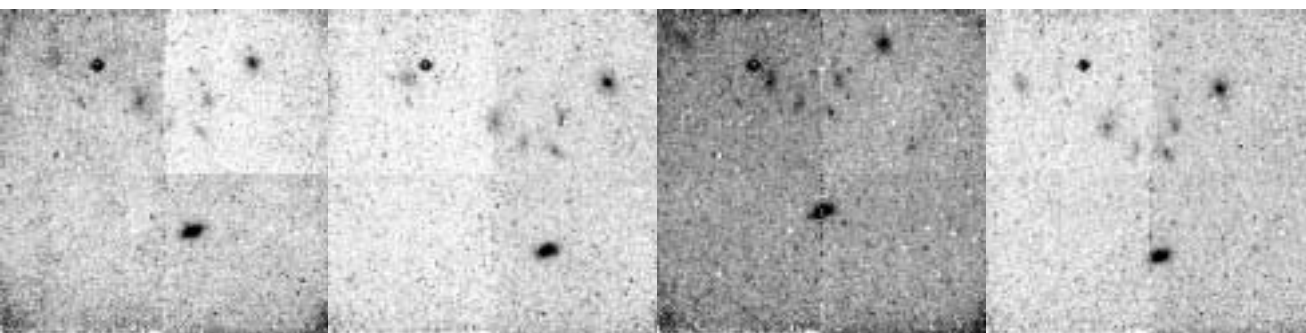
no flat
with post-pedsky

Noise: 42.6e-, Red: 45.3%

Noise: 36.1e-, Red: 32.0%

Noise: 43.2e-, Red: 10.8%

Noise: 35.8e-, Red: 10.9%



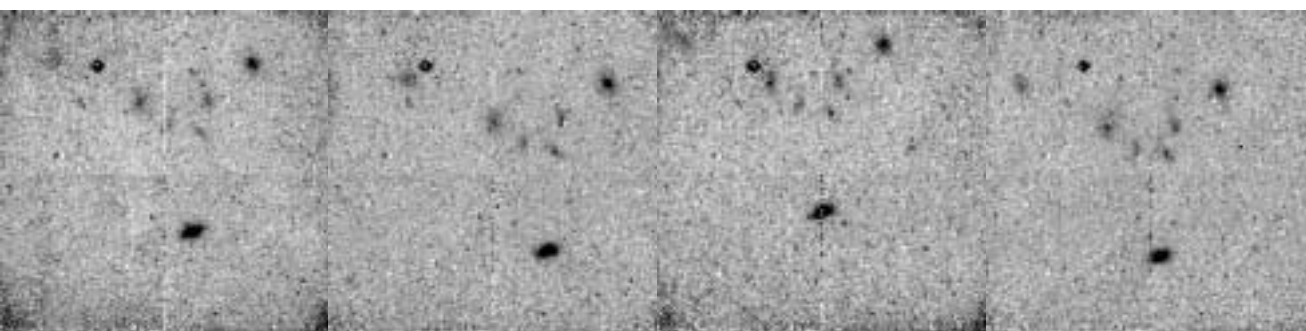
with flat
no post-pedsky

Noise: 48.0e-, Red: 38.4%

Noise: 38.9e-, Red: 26.7%

Noise: 44.5e-, Red: 8.13%

Noise: 38.7e-, Red: 3.58%



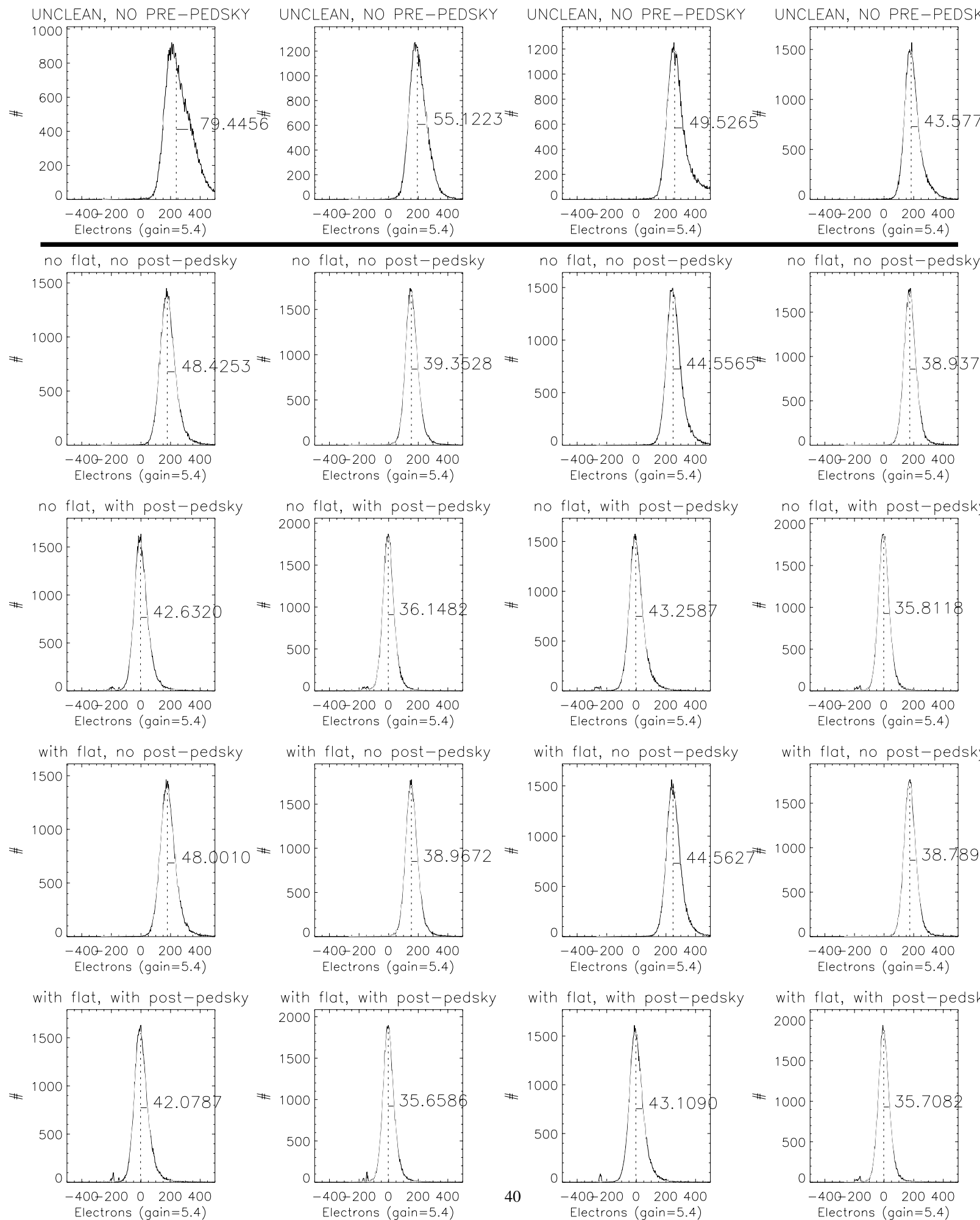
with flat
with post-pedsky

Noise: 42.0e-, Red: 46.0%

Noise: 35.6e-, Red: 32.9%

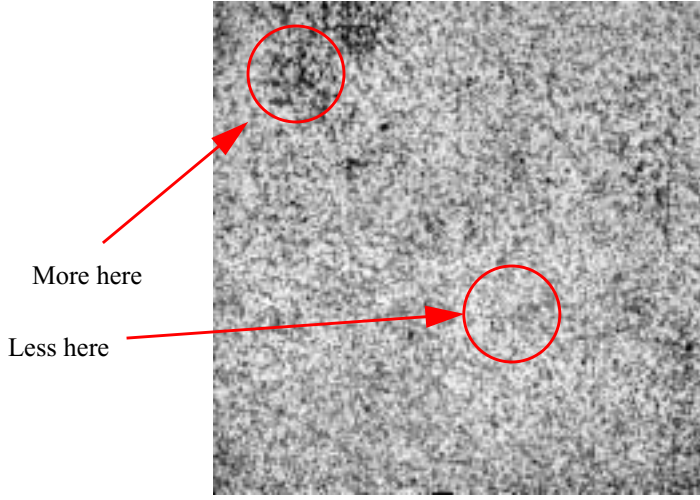
Noise: 43.1e-, Red: 11.1%

Noise: 35.7e-, Red: 11.2%



Appendix C

While analyzing a large number of post-SAA darks taken in May 2002 to make figure 9, we noticed that in the NIC2 images the persistent signal showed higher persistent countrates on average in the upper left part of the detector (figure 23). This spatial dependence likely represents a



real trap density distribution change from place to place on the device. In other words, more traps in the upper left mean more trapped charge seen where CRs hit during the SAA passage. If you assume a uniform (and random) distribution of particle energies in a given SAA passage, then you can produce a map of the trap density by taking the average - or better still, the max - of a large number of SAACRMAP images. 232 post-SAA darks were analyzed. These covered a wide range of SAA passage depths. Figure 24 shows a plot of the mean persistence countrate for each camera in this dataset, and the histograms of each.

Figure 23. SAACRMP showing uneven spatial distribution of CR persistence.

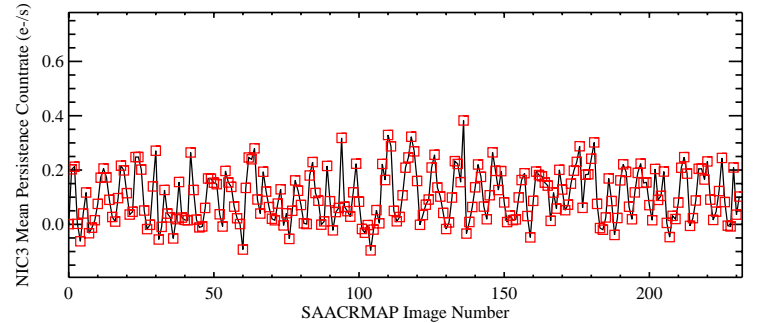
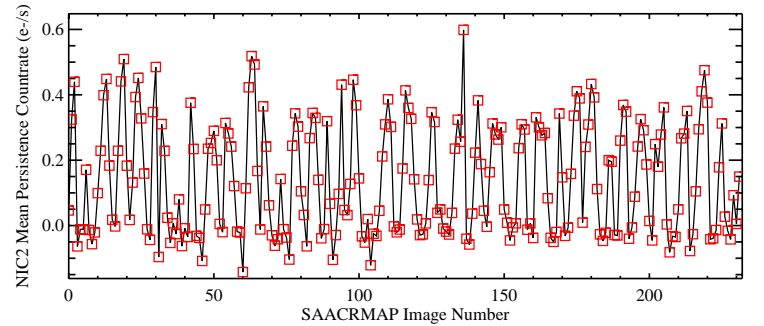
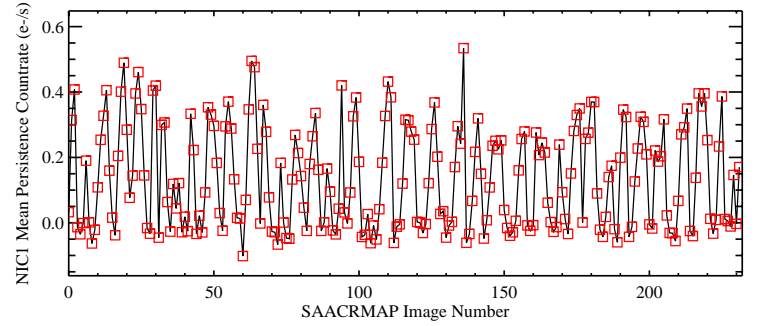
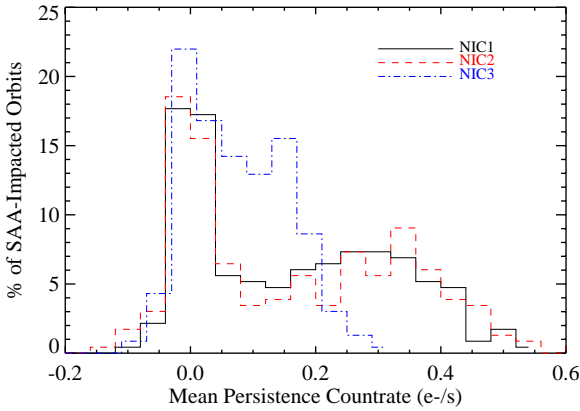


Figure 24. (Right) Array mean SAA persistence countrate measured in a 256s exposure starting 174s after exit from the SAA for 232 datasets taken in May 2002. Assumed gain was 5.4, 5.4 and 6.5 e-/DN for NIC1, 2 and 3 respectively. (Above) Histograms of the countrate data at right.

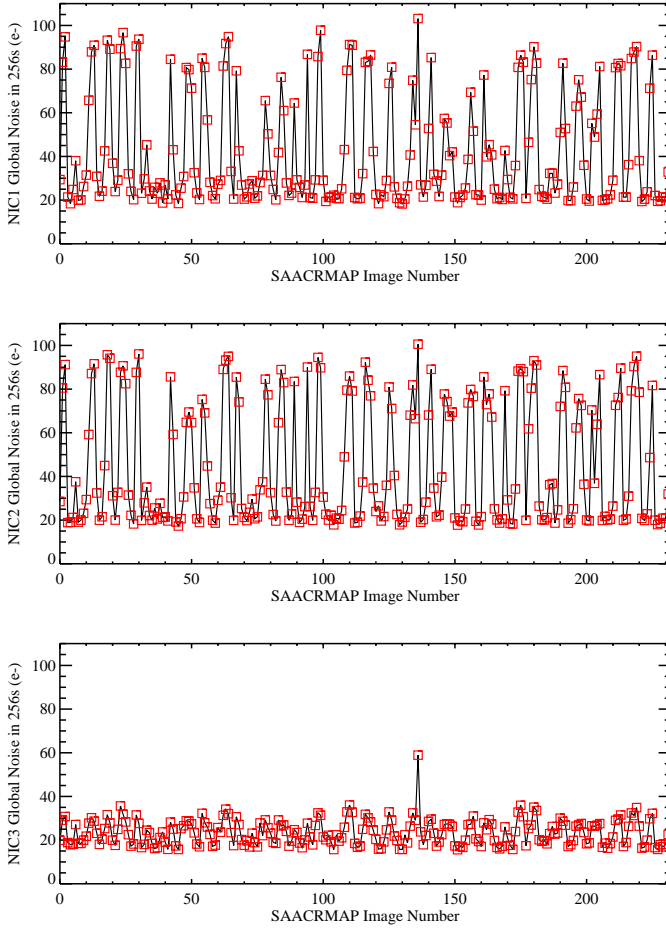
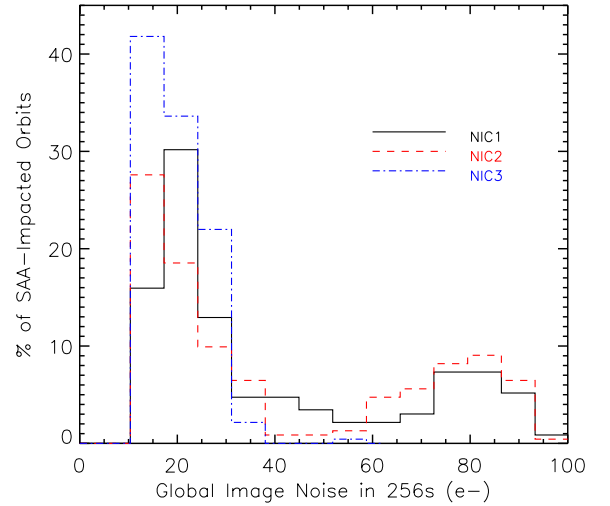


Figure 25. (Above) Global image noise in a 256s exposure starting 174s after exit from the SAA for 232 datasets taken in May 2002. Assumed gain was 5.4, 5.4 and 6.5 e-/DN for NIC1, 2 and 3 respectively. (Right) Histograms of the above sigma data.

Figure 25 shows the Global noise measured in the same images as figure 24. Note the min noise is ~ 20 e-, which is about what is expected for non-impacted NREAD=25 ACCUM mode darks. Once again, this is the reason we use the NREAD=25 ACCUMs for the post-saa darks (see section 3 in the text). A typical MULTI-ACCUM sequence of the same duration would have ~ 32 e- of noise.

All the images taken here were made in parallel, so each point in figures 24 and 25 was observed simultaneously with the other two cameras. Note that the NIC3 on average has a lower



mean persistence countrate than NIC1 and NIC2, which are very similar to each other. The same is true of the induced noise. An interesting pattern appears when the mean persistent signal is plotted against the measured sigma (figure 26). This is likely due to a combination of the HST ground tracks through the varying SAA particle density and the exponential decay. The farther the SAA exit is from the highest particle density portion of the SAA (see figure 1) then the more time has elapsed since the majority of the pixels were hit. After some time, the rapid part of the persistence decay curve (figure 2) has passed. Thus the highest signal hits which cause the largest sigma values have decayed and are closer to the array mean.

The max value per pixel for all the persistence images was taken to make a map of the relative trap density. This was done for all three cameras (figure 27). As can be seen from these images and from the plots in figures 24 and 25, NIC3 has an apparent lower trap density than the other two cameras. There are two possible explanations for this. 1) that device simply has an inherently lower trap density, or 2). perhaps the NIC3 is shielded somewhat better from the charged particles in the SAA than the NIC1 and NIC2 devices (which are physically very close to each other in the cold-well, while the NIC3 is closer to the top and the filter wheels). There is a single point in the data - image #136 - which has an unusually high mean countrate and sigma. In figure 26, this point is the single high NIC3 (blue) point rising up the “jump” at ~ 0.38 e-/s mean

countrate. The fact that NIC3 can exhibit even higher persistent signal when the “illumination” is higher suggests that the NIC3 device is physically similar to the other two, but for some reason gets fewer particle hits or a prevalence of lower energy ones. Looking at some bright-star persistence images using all three cameras may help resolve the issue. In any case the *relative* spatial trap density shown in figure 27 within each device is valid even if the absolute comparison from device to device cannot be made here because of the uncertainty in illuminating flux from the SAA.

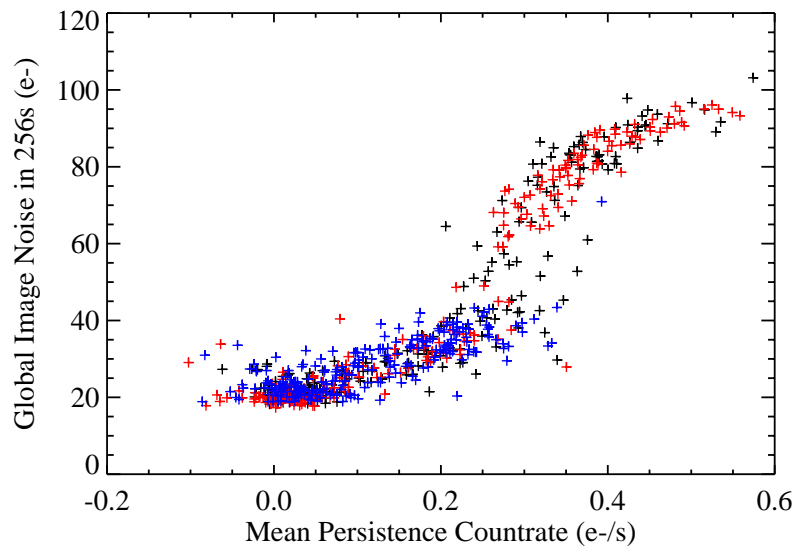


Figure 26. Plot of 232 post-saa darks (per camera), mean vs. sigma. Black is NIC1, red is NIC2 and blue is NIC3



Figure 27. Relative trap density maps for the three NICMOS detectors as determined from post-SAA cosmic ray persistence. From left to right, NIC1, NIC2 and NIC3. Darker is higher density.

Appendix D

The degree to which an SAA impact affects data quality in subsequent science exposures is a function of at least three variables: the depth of the SAA crossing, the time spent in the SAA, and the length between the exit from the SAA and the science observation. We have examined the dependence of the amplitude of SAA persistence on the first two of these variables, eliminating the third by using the post-SAA darks which are taken at a nearly uniform delay time after SAA-exit.

A total of 1273 pairs of post-SAA ACCUM darks for NICMOS camera 2 were retrieved from the HST Archive and processed through a modified version of the post-SAA IDL tool. A clean 256 second ACCUM dark, generated by the post-SAA tool, was subtracted from each post-SAA dark to remove the nominally constant bias and dark current structure from the images. The pairs of images were then combined with cosmic-ray rejection, as described in Section 3), using the default scaling parameters. These images therefore map the pattern of post-SAA persistence after each of 1273 different SAA passages.

We measured various statistics on these images, excluding pixels which are known to be bad (i.e., which are flagged in the data quality masks), or which were consistently deviant from zero in all images (as measured from a median stack of all of the data). Several statistics were tried. The overall amplitude of the persistence signal was estimated by the mean level of the images, computed after iterative 10-sigma clipping to reject extreme outliers. The amplitude of the spatial structure of the persistence signal was quantified by the RMS of the images, again after 10-sigma iterative clipping. The third statistic looked at the positive, non-Gaussian tail of the pixel value distribution that is caused by the persistence signal. We analyzed a large number of relatively “clean” post-SAA dark pair combinations, taken during shallow SAA passages (see below), and found that their RMS in absence of persistence is, on average, approximately 0.02 ADU/second. We therefore adopted 0.1 ADU/sec (or 5 times the “clean” RMS) as a threshold indicating bad persistence signal, and counted the number of pixels in each post-SAA-dark combined image that exceeded this threshold (after excluding the statically masked bad pixels). This third statistic turns out to be the most sensitive to the SAA signal, as tails of distributions often tend to be; the mean and RMS measures are both “diluted” by large numbers of relatively modestly affected pixels and are less sensitive to the differences between SAA passages.

A database of information about the 1273 SAA passages was provided by Merle Reinhart. This gave the entry and exit times for HST crossings through the various SAA contours. The contours represent concentric levels of SAA intensity, and their numbering scheme is purely historical. In order of increasing depth, the contours are numbered 23, 26, 24, and 2. (Contour 5 is equivalent to 23, and 25 is the same as 24.) The SAA event log, together with the Modified Julian Date of the first post-SAA dark exposure, was used to evaluate (1) the deepest contour crossed during the passage, and (2) the duration of the passage (for each contour crossed) from entry to exit.

Figure 29 shows histograms of the distributions of these three statistics for the 1273 crossings analyzed here, divided into two subsets: orbits which crossed SAA contour 2, and those which did not. The amount of persistence, by any of these statistics, is much greater for contour 2 crossings than for orbits which do not cross contour 2. Inspecting similar histograms for various maximum-depth contour crossing thresholds show modest (but noticeable) increases in the typical level of SAA persistence as one moves deeper from contour 23 to 26 to 24. However, in nearly all cases the persistence is mild compared to that seen for contour 2 orbits.

Even for contour 2 crossings, however, some frames are only mildly affected by persistence. Figure 30 shows the values of these three statistics measured for individual post-SAA dark pairs in orbits with contour 2 crossings, plotted as a function of the length of time spent in contour 2. The amplitude of the persistence is a clear function of the time spent within contour 2, increasing gradually from crossing durations of 0 to 700 seconds, then rising more sharply thereafter to the maximum duration of approximately 1300 seconds.

Many NICMOS science observations, particular of faint objects, are adversely impacted by post-SAA persistence. The development of the SAA correction tool described here can ameliorate this to some degree, but deep imaging programs of faint targets are best scheduled in SAA-free orbits. Based on the analysis presented here, we have recommended that

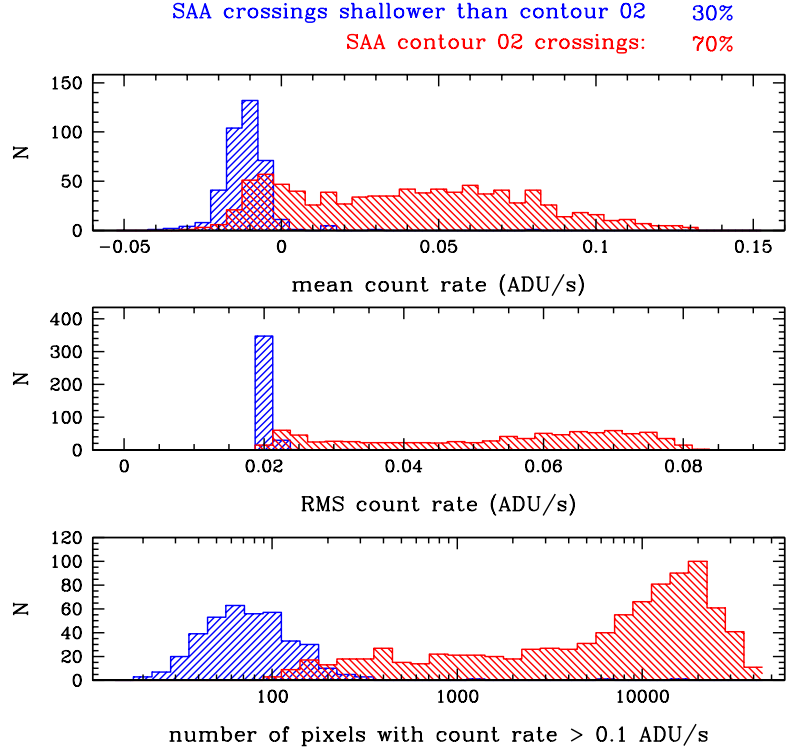
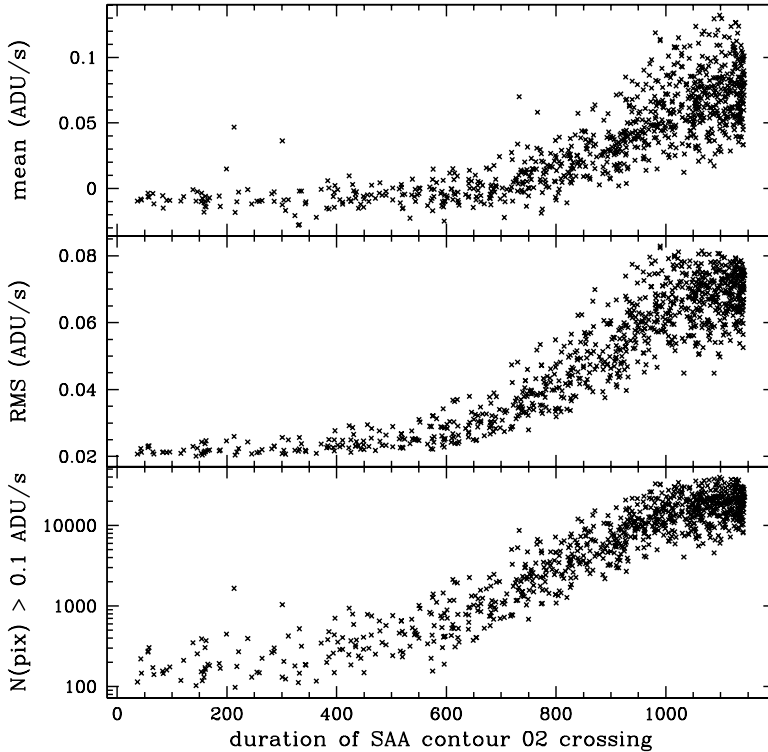


Figure 29. The distributions of three SAA-induced noise statistics as a function of the two defined operational SAA contours for NICMOS. Compare the top two panels with Figures 24 and 25 of Appendix C, which show similar stats broken down as a function of camera.



“SAA-free” orbits (from the NICMOS standpoint) be defined as those orbits which do not cross contour 2. Out of the 1273 passages (i.e., contour 23 crossings) analyzed here, 70% crossed SAA contour 2. Redefining SAA-free NICMOS orbits to be crossings outside contour 2 therefore has a small, but significant, impact on the number of orbits available for deep NICMOS imaging. The post-SAA darks for correction of SAA persistence should continue to be taken following crossings of both contour 02 *and* contour 05.

Figure 30. The same three statistics as in Figure 29 plotted as a function of the duration of the preceding SAA contour 02 Crossing. The contour 2 and contour 5 outlines are shown on the map in Figure 1.