

ACS calibration pipeline testing: cosmic ray rejection

Max Mutchler, Warren Hack, Robert Jedrzejewski, Doug Van Orsow
Space Telescope Science Institute
30 November 1999

ABSTRACT

We describe the testing that was done to ensure that CALACS, the calibration software “pipeline” for the Advanced Camera for Surveys (ACS), properly rejects cosmic ray contamination when combining multiple images. The suite of simulated raw images and reference files described here are reproduceable and therefore can serve as a regression test as future versions of CALACS are created.

1. Introduction

The ACS calibration software pipeline (CALACS) will be automatically applied to raw ACS images, to remove various instrument artifacts, and populate the image headers with relevant information. For a detailed description of CALACS and the testing of basic image reduction, see ACS Instrument Science Reports 99-03 and 99-04. This report describes testing of only the ACSREJ task, which rejects cosmic ray contamination while combining multiple images.

Most ACS observations will be dithered for improved resolution, and/or split into multiple exposures to facilitate cosmic ray rejection (CR-SPLIT in proposal syntax). Therefore, most observations will be a part of an association of several images which must be processed simultaneously by CALACS.

We created several simulated raw input images for each CCD detector: the Wide Field Channel (WFC), and the High Resolution Channel (HRC). The Solar Blind Channel (SBC) is a MAMA detector which is not sensitive to cosmic rays, so CALACS does not perform cosmic ray rejection for SBC data. We will use the HRC as an example throughout this report, although the same testing was done with simulated WFC images.

2. Background on cosmic rays

Most of the signal in a typical Hubble Space Telescope (HST) image comes from the light intentionally collected by the optical system. But HST images also get contaminated by cosmic rays and protons from the Earth's radiation belts. They are very energetic and pass directly through the spacecraft and anything inside it, including the CCD detectors. Therefore, the HST optical system is irrelevant for determining the cosmic ray rate: only the local cosmic ray flux at the time of the observation, and the physical cross-section of the detector are relevant. The cosmic ray flux varies depending on the geomagnetic latitude of HST, and its proximity to the South Atlantic Anomaly (SAA), where the flux is much higher. For our testing purposes, however, we will ignore this variability, and assume average non-SAA cosmic ray flux values. The following equation can be used to estimate the percentage of pixels affected by cosmic rays in an image:

$$C = \frac{A \times R \times N \times T}{P}$$

'C' is the percentage of pixels contaminated by cosmic rays. The rejection of cosmic rays becomes increasingly difficult as a larger percentage of the pixels are contaminated. Exposure time should be split (CR-SPLIT) enough to ensure successful cosmic ray rejection. If long exposure times (i.e. much longer than 600s) are used, then the observer must be careful to obtain an adequate number of exposures for complete cosmic ray rejection.

'A' is the physical area (cross-section) of the detector. The WFC has 4096 x 2048 pixels which are each 15 microns across, so the total area is 18.87 cm². The HRC has 1024 x 1024 pixels which are each 21 microns across, so the total area is 4.62 cm².

'R' is the rate of cosmic ray events. Analyses of this rate have been conducted for other HST instruments -- see the WFPC2 Instrument Handbook, and STIS Instrument Science Report 98-22 by Shaw. They report a range from 0.5 to 1.9 events/sec/cm². We will use the average value of 1.2 events/sec/cm² for our testing purposes.

'N' is the number of pixels affected by each cosmic ray event. The STIS report estimates that 6-10 pixels are typically affected by each cosmic ray event. Our analysis of the cosmic rays present in archival STIS dark exposures affirms these values for R and N. The STIS CCD is identical to the HRC, and so we use the average value of 8 pixels per event for the HRC. Since WFC pixels have about half the surface area as HRC pixels, we can expect that 12-20 WFC pixels would be affected by each cosmic ray event. We use the average value of 16 pixels per event for the WFC.

'T' is the exposure time. For WFPC2 and STIS, exposures longer than 10 minutes are automatically split to keep the value of 'C' reasonably low, unless the user turns off this default by specifying CR-SPLIT=NO in their proposal. ACS will likely have a similar

splitting threshold. Certainly, some proposers will design observations with longer exposure times, but they will then need to make sure they have an adequate number of exposures for successful cosmic ray rejection. For our test, we created some 600 second exposures (default CR-SPLIT), and some 1800 second exposures.

‘P’ is the total number of pixels in the detector. Each WFC CCD has $4096 \times 2048 = 8,388,608$ pixels. The HRC CCD has $1024 \times 1024 = 1,048,576$ pixels.

Table 1 displays the results of applying the equation and estimates above. Note that the values for the cosmic ray rate (R) and the number of pixels affected by each event (N) are “best guess” averages. Our goal is to create simulated images which contain a typical amount of cosmic ray contamination for various exposure times, and this equation provides only a crude estimate. The previous studies mentioned above state that actual contamination levels may vary by as much as 60% from the average.

Table 1: Estimated amounts of cosmic ray contamination for the ACS CCDs

CCD detector	exposure time	number of cosmic ray events	number of contaminated pixels	contamination rate	percentage of contaminated pixels
HRC	600 sec	3330	26600	44 pixels/sec	2.5%
HRC	1800 sec	9990	79900	44 pixels/sec	7.6%
WFC	600 sec	13590	217400	362 pixels/sec	2.6%
WFC	1800 sec	40770	652300	362 pixels/sec	7.8%

3. Simulated raw images

We created raw images using the `noao.artdata` package in IRAF. See the `make-eraw.cl` script in the Appendix for details. Simulated images are ideal for testing the degree of success of cosmic ray rejection, since we have the luxury of being able to compare the CALACS output image, with the “true” image (i.e. our raw image before we added cosmic rays).

Each raw image contains stars, galaxies, a sky background, and cosmic rays (see Figure 1). We also added a 5,000 DN bias level with corresponding overscan regions. The cores of bright stars and galaxies pose a good test for cosmic ray rejection, since they can have energies similar to cosmic ray events. The stars have peak pixel values ranging from about 100 DN (just above the sky background) up to values near the full well capacity of 65,000 DN (which would be saturated in a real image). For our simulated stars, we used synthetic point spread functions (PSFs) generated by TinyTIM which are typical of the detector’s response at 800 nanometers (see Figure 2).

For both the WFC and HRC, we created two sets of raw images. The first set is three 600 second exposures, typical of an observation with the default CR-SPLIT. The other set

is eight 1800 second exposures, typical of an observation where the default splitting has been turned off (CR-SPLIT=NO). Figure 3 shows a comparison of the true image with both a 600 second and an 1800 second raw image. Figure 5 shows the full-format pipeline processed output for the HRC.

The cosmic rays that we added roughly conform to the quantities and energies appropriate for our chosen exposure times, as listed in Table 1. Figure 4 shows a cumulative histogram of our cosmic ray masks, which confirms that we contaminated an appropriate amount of pixels in each simulated raw image. As a check, we made a similar histogram for the cosmic ray masks of several STIS dark exposures, to verify that real exposures have a comparable amount of contamination.

4. Pipeline processed images

Association tables are used as CALACS input when multiple images are to be processed together, as in the case of images split for cosmic ray rejection, and/or dithered images. For both the short and long exposures, we created association tables that included all the raw images, and other tables which only included various subsets of the raw images. For example, the following is the association table for two 600 second HRC exposures:

```
# Table hr12_asn.fits[1] Tue 16:22:53 22-Jun-99
# MEMNAME          MEMTYPE          MEMPRSNT
  hr1              EXP-CR1             yes
  hr2              EXP-CR1             yes
  hr12             PROD-CR1            yes
```

When only two images are used as CALACS input, some residual cosmic ray contamination is evident in the output images (see Figure 6). This is an example of what happens to observations with an inadequate amount of splitting: it becomes impossible for CALACS to completely rid the combined output image of cosmic ray contamination.

When a third raw image is included in the association, the output for the 600 second exposures is cleaner. However, there is still a significant contamination residual in the output for three 1800 second exposures, which is only removed after a fourth exposure is included.

To test CALACS ability to handle a large association of images, we used another association table that included all eight 1800 second raw images, which CALACS was able to process without any problems. In the case of WFC, this is a considerable amount of raw input data: 550 megabytes! Furthermore, the following equation can be used to estimate

the amount of disk space required to run CALACS on an association of (n) exposures (from ACS ISR 99-03 by Hack):

$$\text{disk space} = (1+n)(\text{calibrated image size}) + (n)(\text{raw image size})$$

Raw HRC images are 4.4 MB, and calibrated HRC images are 10.5 MB. Raw WFC images are 68.7 MB, and calibrated WFC images are 168 MB. So, for example, running CALACS for an association of eight WFC images requires 2062 MB of disk space!

While running CALACS on our simulated images, we used mostly “identity” reference files which should have no effect on the output image. Our identity BIASFILE, DARKFILE, and SHADFILE, have all pixels set to 0 DN. Our PFLTFILE and DFLTFILE have all pixel values set to 1 DN. For details about these test reference files, see ACS Instrument Science Report 99-04. Most relevant to this test is the cosmic ray rejection reference file (*_crr.fits):

```
# Table hr_crr_id.fits[1] Mon 15:15:16 18-Oct-1999
# CRSPLIT INITGUES SKYSUB CRSIGMAS CRRADIUS CRTHRESH CCDCHIP
  2 minimum mode 6,5,4 2. 0.1 1
  3 minimum mode 6,5,4 2. 0.1 1
  4 minimum mode 6,5,4 2. 0.1 1
  8 minimum mode 6,5,4 2. 0.1 1
```

The CRSIGMAS control the number of iterations and the stringency of the cosmic ray identification, e.g. how many sigmas above the sky background a pixel must be to be identified as cosmic ray contaminated. See ACS ISR 99-08 by Hack for details about this and other reference files.

The success of the cosmic ray rejection can be determined by comparing the CALACS output image to the true image -- ideally, they should look identical. We subtracted the CALACS output from an appropriately scaled and trimmed version of the true image, and found minimal cosmic ray residuals (see Figure 7). The small amount of residual contamination visible is generally in pixels adjacent to pixels which experienced more significant contamination in one of the raw images, and is to be expected on this level.

5. Using STIS images to check CALACS

As a check, we also tested CALACS cosmic ray rejection using real STIS CCD data, which should be very similar to real HRC data. We obtained raw STIS data from the archive (rootname o46p1u010), along with the reference files used, and the corresponding CALSTIS output files.

Before we could run the STIS data through the CALACS pipeline, some image format differences had to be resolved. The CALACS code uses line by line I/O unlike the CAL-

STIS code which reads in whole images at once. The HRC FITS files contain only one imset, unlike the STIS FITS files which can contain more than one. The ACSREJ task requires more than the one STIS FITS file, so the imsets were copied into preexisting HRC fits files, making sure not to overwrite the HRC header information.

To compare the images (i.e. subtract one from the other) , they must have the same dimensions and align with each other. The overscan regions in the raw STIS images have slightly different dimensions, and the virtual overscan region is located at the the bottom of the chip instead of the top, so the overscan reference table was modified accordingly.

After running the raw STIS data through the ACSCCD and ACSREJ tasks, the LTV2 keyword must be adjusted in order to process the remaining steps in the ACS2D task. LTV2 is set from 0 to -20 due to the virtual overscan difference described above, but needs to be set back to 0 in order to process the DARK and FLAT reference files.

We ran the raw STIS images through the CALACS pipeline using the same reference files used by CALSTIS, and then compared the output from the two pipelines by subtracting one from the other.

Most of the differences between the CALSTIS and CALACS output images are marked in the CALSTIS data quality file as "lost data replaced by fill values". By turning off DQICORR and rerunning the data through both pipelines again, we were able to eliminate these discrepancies.

A satellite trail is seen in one of the two raw images. It can also be seen in both the CALSTIS and CALACS cosmic ray rejected images and the subtracted image. There appears to be some type of ramp across the subtracted image which can also be seen in the CALACS cr-rejected image.

6. Summary

The testing described in this report demonstrates that the ACSREJ task in CALACS is functioning properly. The scripts that produced the raw images and our diagnostic products are provided in the Appendix. This test can be reproduced with the information provided in this report, and serve as a standard regression test for future versions of CALACS.

The parameters used in our test CRREJTAB suited our testing needs, but will require futher testing with real data to determine optimum settings for various datasets. Future reports and the ACS Instrument Handbook will provide guidance to proposers trying to determine optimum exposure splitting and calibration strategies.

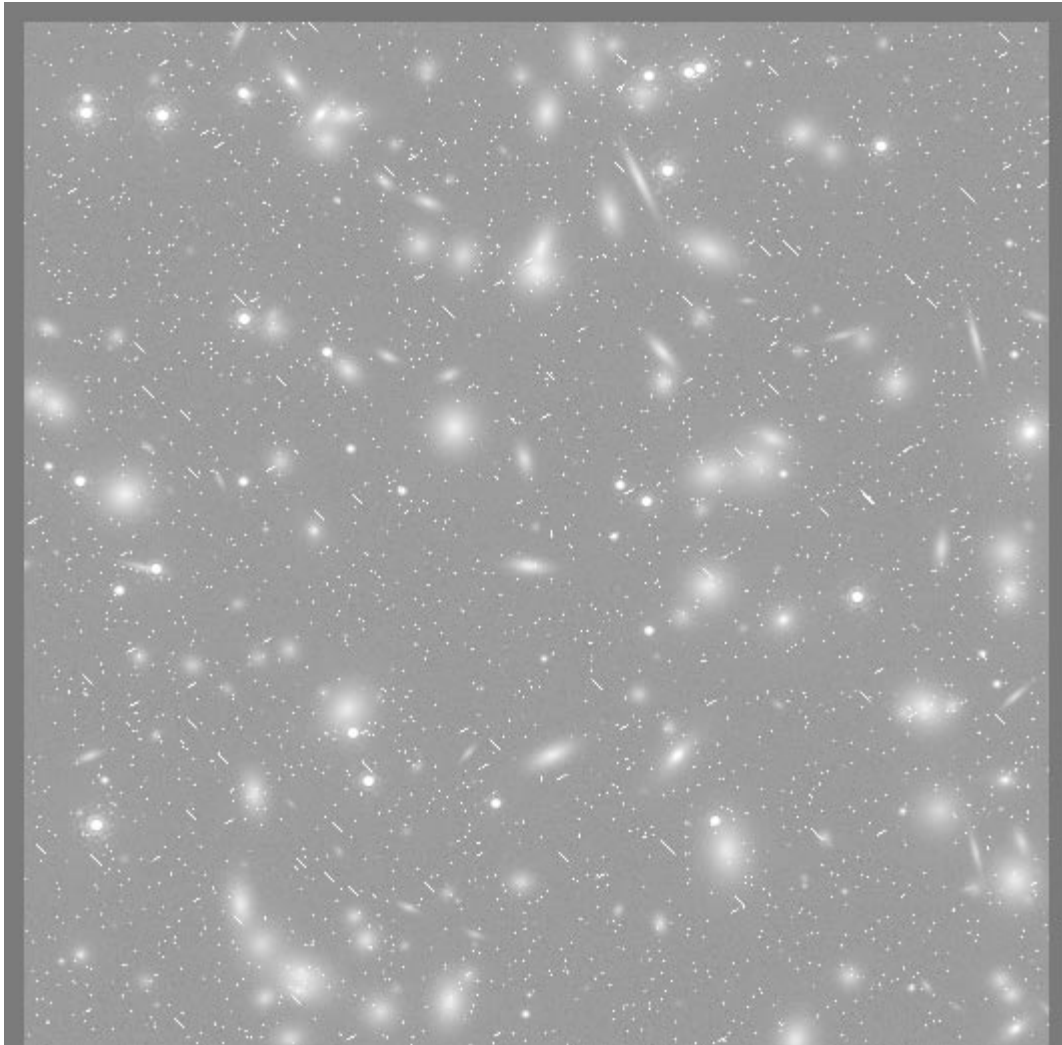


Figure 1. One of the three 600 second raw HRC images we created with the IRAF `noao.artdata` package. The image is displayed with a stretch that reveals the overscan regions. The physical overscan regions are the first and last 19 pixels in each row, and the virtual overscan region is the last 20 rows on top. The dimensions of a raw HRC image are 1062x1044 pixels. Two more 600 second images like this were created with different cosmic rays.

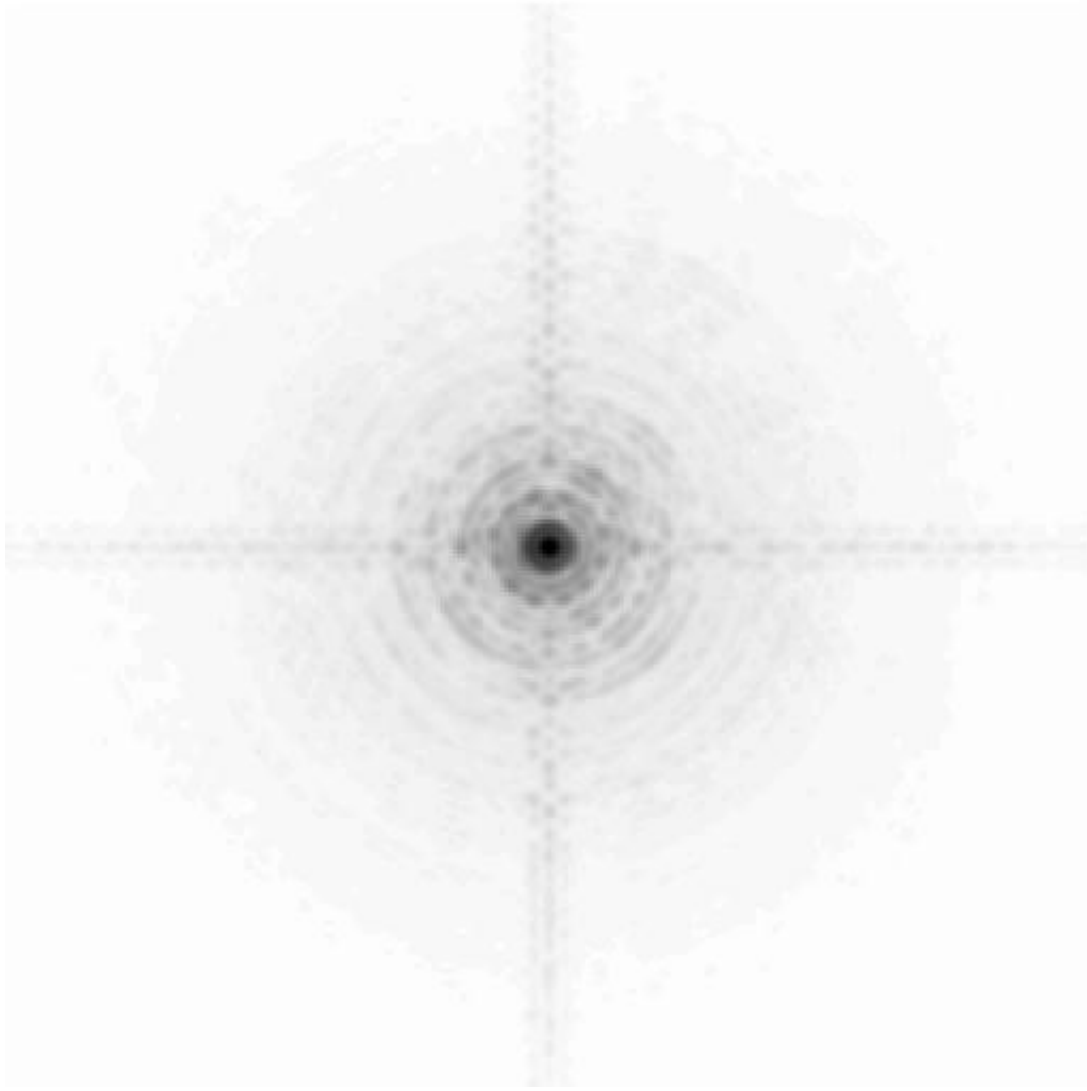


Figure 2. The point spread function (PSF) used to create stars in our simulated HRC images. This is a synthetic 800 nm HRC PSF created by George Hartig using TinyTIM. A similar PSF was used to make stars in the WFC images.

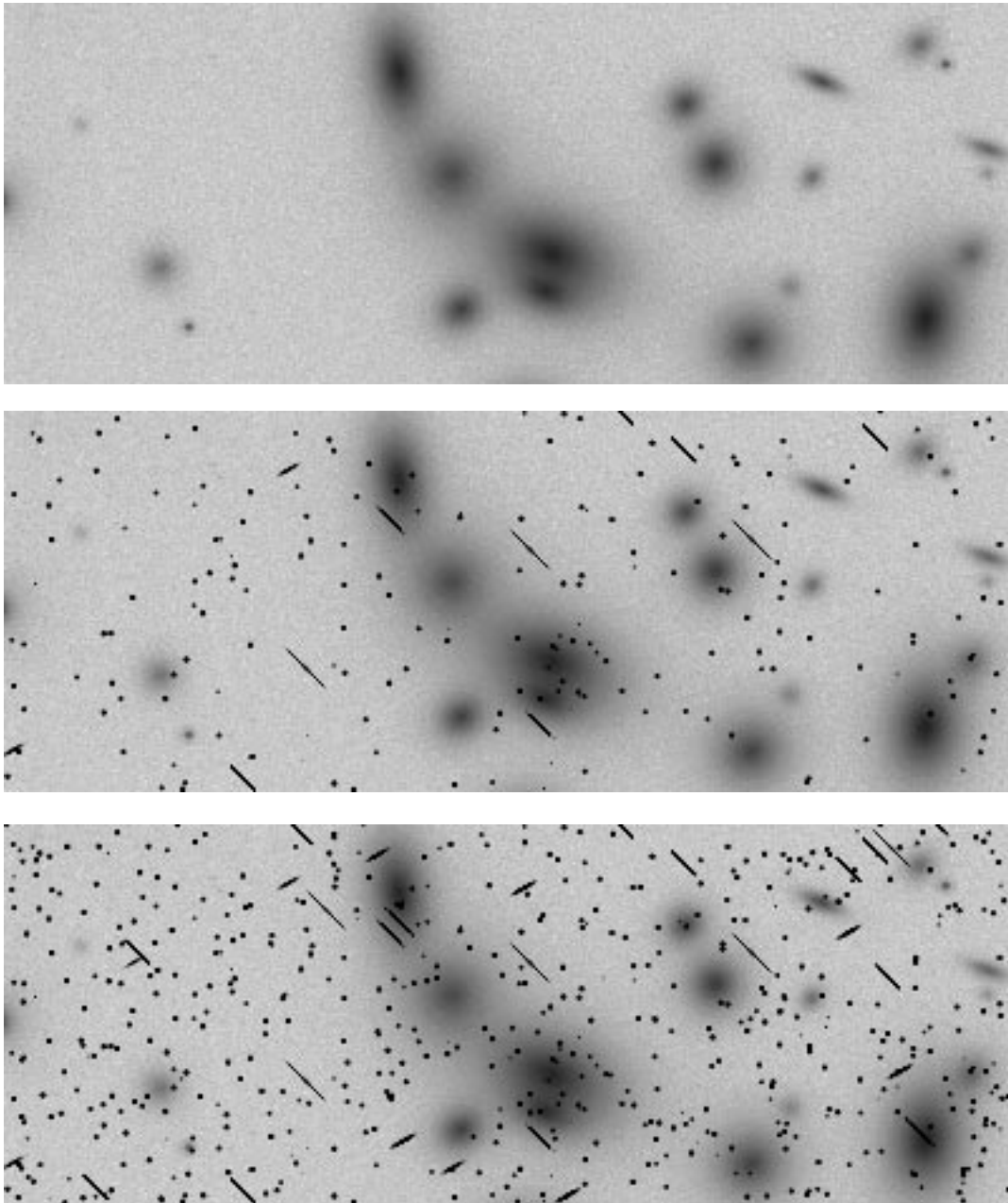


Figure 3. A comparison of a section of our simulated “true” HRC image (top) with no cosmic rays, one of our 600 second raw HRC images (middle), and one of our 1800 second raw HRC images (bottom). Note the relative degree of cosmic ray contamination in the raw images, and the mixture of energy levels and incidence angles.

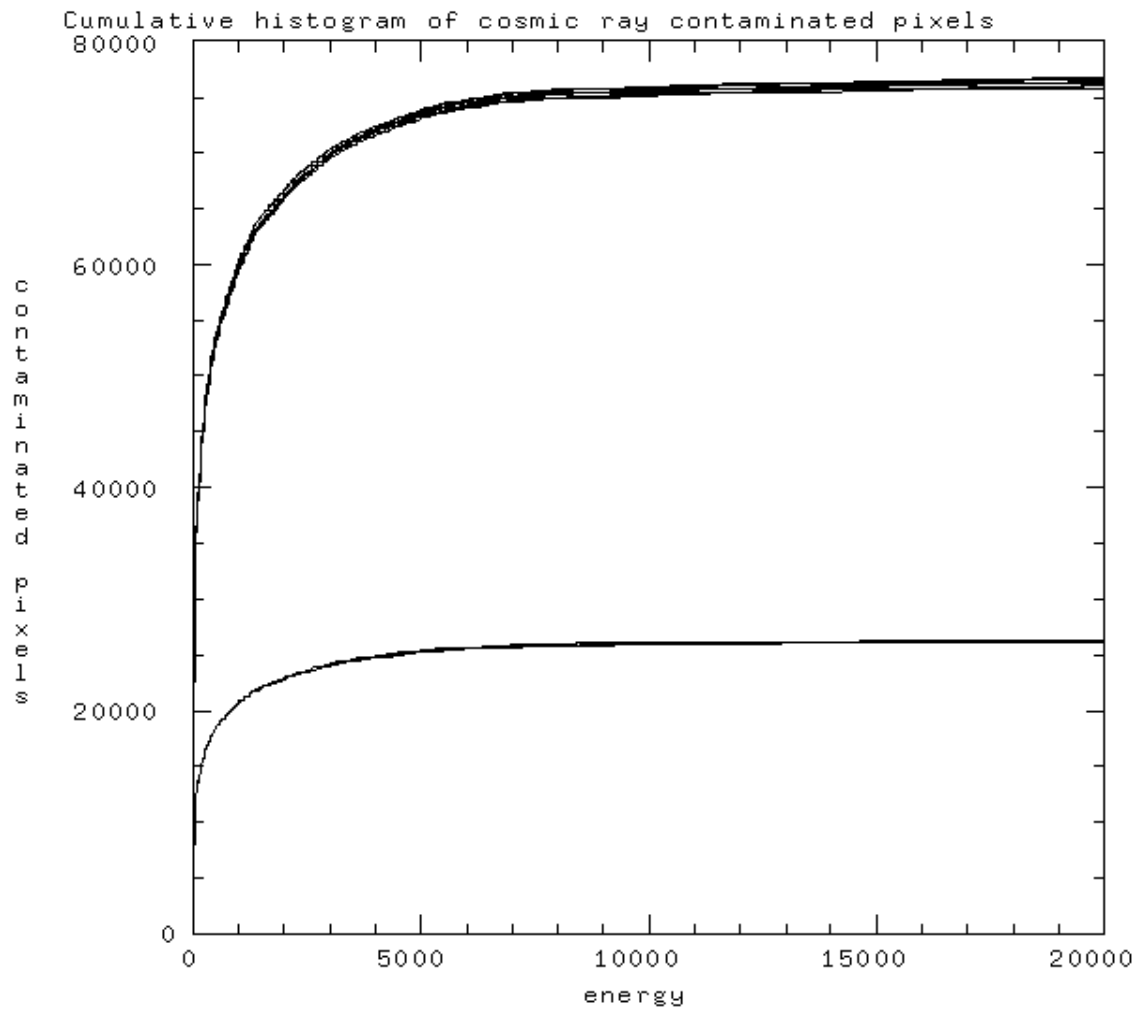


Figure 4. The cumulative histogram of contaminated pixels for all of our simulated raw HRC images. The lower curves are created by our three 600 second exposures, which verifies that they contain about 26,000 contaminated pixels (2.5%). The upper curves are created by our eight 1800 second exposures, which verifies that they contain about 79,000 contaminated pixels (7.6%).

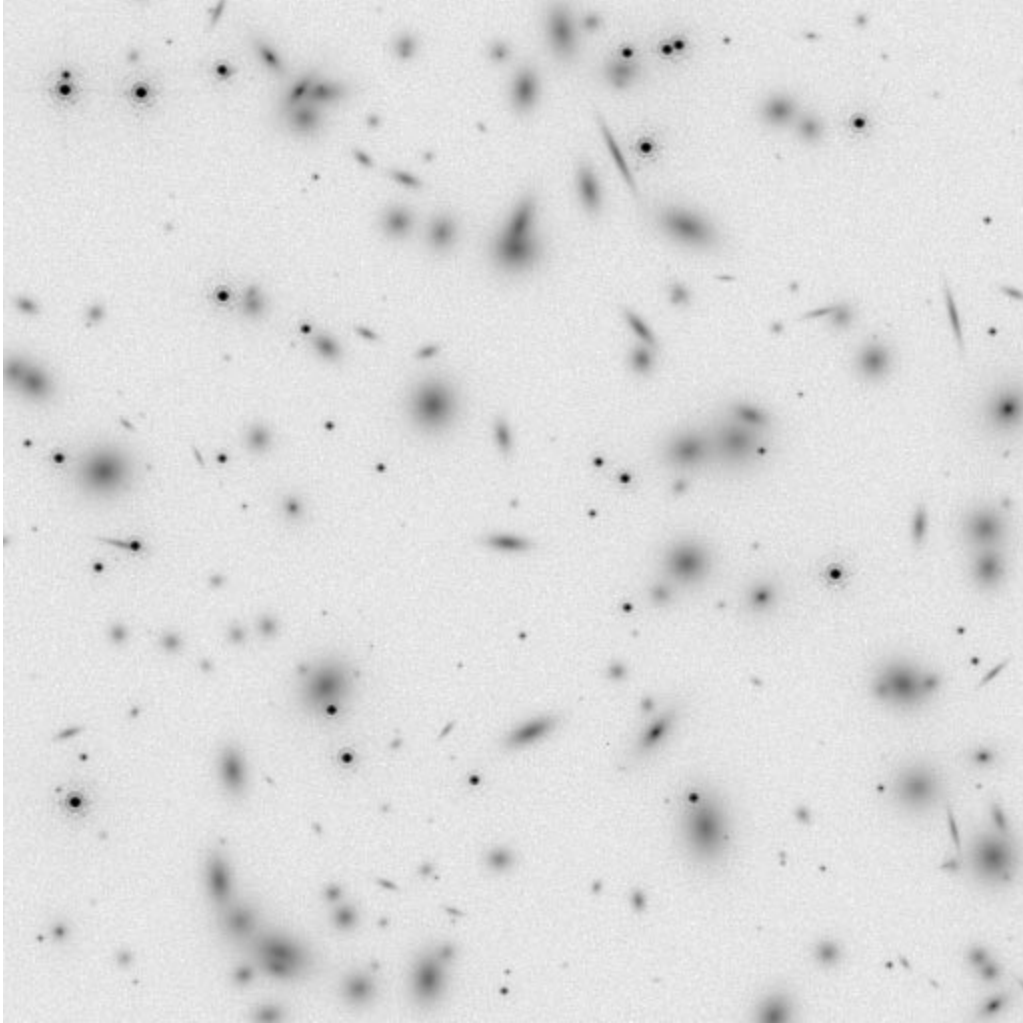


Figure 5. The CALACS output when all three 600 second raw HRC test images are input. Cosmic rays are removed, and the bias level (5,000 DN) has been measured from the overscan regions, and then subtracted from the entire image. Then the overscan regions were trimmed off, so the output image dimensions are 1024x1024 pixels. For the WFC, there are 24 physical overscans at the beginning and end of each row, so each inset is trimmed from 4144x2068 to 4096x2048 pixels. There are no overscans for MAMA (SBC) data.

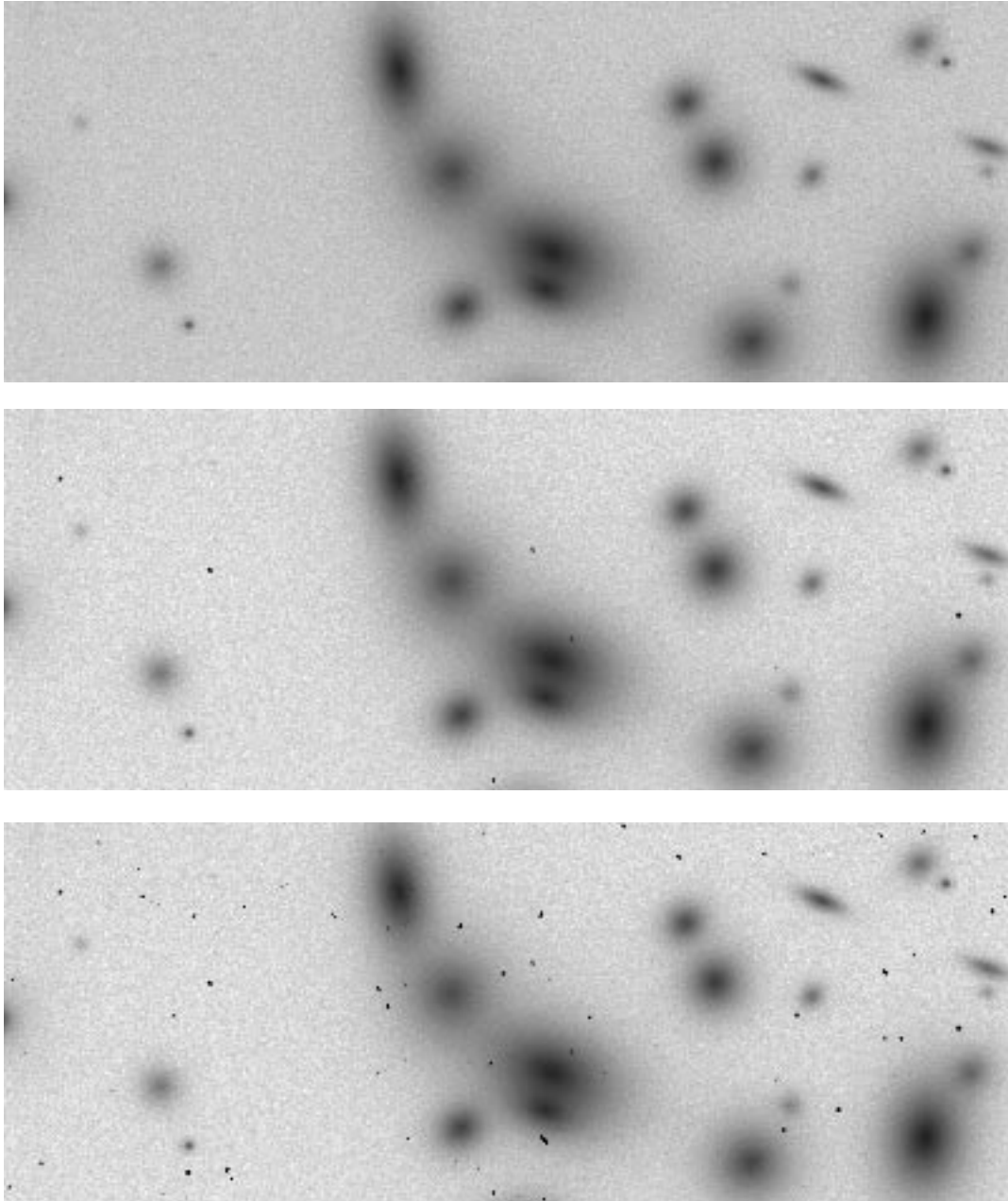


Figure 6. The results of CALACS cosmic ray rejection when only two raw images are used as input. Compare a section of the true image (top), with the corresponding section of the CALACS output for the 600 second exposures (middle) and 1800 second exposures (bottom). Note the varying degree of residual cosmic ray contamination in the output images. To completely reject all the cosmic ray events, it took three of the 600 second exposures, and four of the 1800 second exposures.

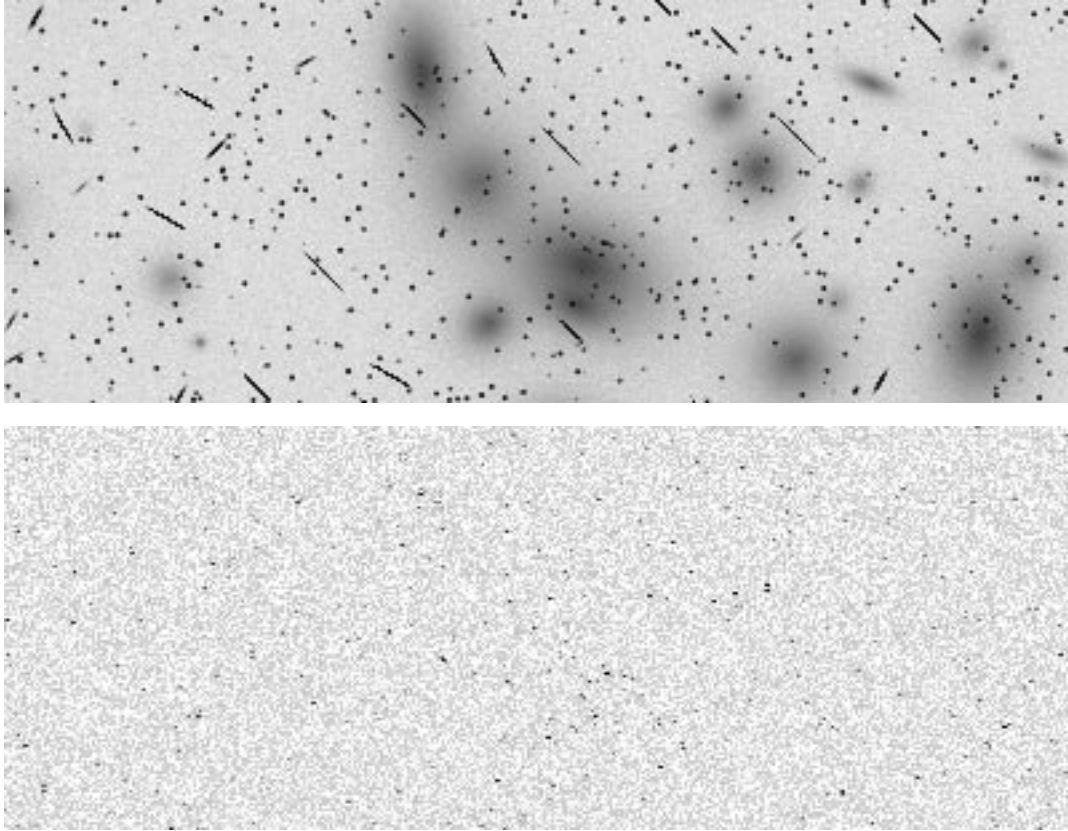


Figure 7. Residual cosmic ray contamination. The image on top is a section of a composite image made by stacking all three 600 second images, with an upper scale of 18000 DN, to show the location of all the cosmic rays involved in the processing. On the bottom is the corresponding section of the residual image created by subtracting a properly trimmed and scaled version of the true image from the CALACS output image. The residual image is scaled to 100 DN so any features are very faint compared to features in the stacked image -- the maximum residual pixel value is only 647 DN.

7. Appendix: IRAF scripts for cosmic ray rejection testing

The IRAF scripts created for testing CALACS cosmic ray rejection are included here in the order of their use. We include only the HRC scripts, but similar scripts were created for the WFC testing.

The `makeraw.cl` script generates the “true” image with stars, galaxies, and sky background. Then it generates a set of “raw” images by making three copies of the true image and adding 600 seconds worth of cosmic rays, a bias level, and overscan regions. Another set of three long-exposure raw images is similarly generated by adding 1800 seconds worth of cosmic rays to each. Finally, the exposure time (EXPTIME) and gain (CCDGAIN) keywords in each image header are set.

Next, the `runcal.cl` script sets the calibration switches and identifies the reference files in the raw image headers. Then it runs CALACS for both sets of raw images: first for only two images, then for all three images. See ACS ISR 99-04 for the contents of this script.

The `compare.cl` script generates several files useful for analyzing the results of the cosmic ray rejection. It creates cosmic ray masks for all three raw input images. It scales and registers the true image with the final output images, so that one can be subtracted from the other, to identify any residuals of the cosmic ray rejection.

makeraw.cl

```
print ("Removing old files...")
del objects.dat
del hr_obj.fits
del hr_obj_nobias.fits
del hr*_art*.fits
del hr*_raw.fits
del hr_psf*.fits
print ("Making stars...")
starlist.sfile = ""
starlist.lfile = ""
starlist.interactive = no
starlist.spatial = "uniform"
starlist.xmin = 1.
starlist.xmax = 1062.
starlist.ymin = 1.
starlist.ymax = 1044.
starlist.xcenter = INDEF
starlist.ycenter = INDEF
starlist.core_radius = 30.
starlist.base = 0.
starlist.sseed = 100
starlist.luminosity = "uniform"
starlist.minmag = -7.5
```

```
starlist.maxmag = 17.  
starlist.mzero = -4.  
starlist.power = 0.6  
starlist.alpha = 0.74  
starlist.beta = 0.04  
starlist.delta = 0.294  
starlist.mstar = 1.28  
starlist.lseed = 100  
starlist.nssample = 100  
starlist.sorder = 10  
starlist.nlsample = 100  
starlist.lorder = 10  
starlist.rbinsize = 10.  
starlist.mbinsize = 0.5  
starlist.graphics = "stdgraph"  
starlist.cursor = ""  
starlist objects.dat 200  
print ("Making galaxies...")  
gallist.interactive = no  
gallist.spatial = "uniform"  
gallist.xmin = 1.  
gallist.xmax = 1062.  
gallist.ymin = 1.  
gallist.ymax = 1044.  
gallist.xcenter = INDEF  
gallist.ycenter = INDEF  
gallist.core_radius = 100.  
gallist.base = 0.  
gallist.sseed = 200  
gallist.luminosity = "uniform"  
gallist.minmag = -7.  
gallist.maxmag = 0.  
gallist.mzero = 15.  
gallist.power = 0.6  
gallist.alpha = -1.24  
gallist.mstar = -21.41  
gallist.lseed = 200  
gallist.egalmix = 0.1  
gallist.ar = 0.3  
gallist.eradius = 20.  
gallist.sradius = 1.  
gallist.absorption = 1.2  
gallist.z = 0.05  
gallist.sfile = ""  
gallist.nssample = 100  
gallist.sorder = 10  
gallist.lfile = ""  
gallist.nlsample = 100  
gallist.lorder = 10  
gallist.rbinsize = 10.
```

```

gallist.mbinsize = 0.5
gallist.dbinsize = 0.5
gallist.ebinsize = 0.1
gallist.pbinsize = 20.
gallist.graphics = "stdgraph"
gallist.cursor = ""
gallist.objects.dat 200
print ("Getting a simulated PSF...")
!cp /data/milan6/calacs/psf/HRC_typ_800_halo.fits hr_psf.fits
print ("Making true image with real objects only, no CRs yet...")
mkobjects.title = "HRC test image by Max Mutchler"
mkobjects.ncols = 1062
mkobjects.nlines = 1044
mkobjects.header = "hrctext.hdr1"
mkobjects.background = 30.
mkobjects.objects = "objects.dat"
mkobjects.xoffset = 0.
mkobjects.yoffset = 0.
mkobjects.beta = 2.5
mkobjects.ar = 1.
mkobjects.pa = 0.
mkobjects.distance = 1.
mkobjects.exptime = 1.
mkobjects.magzero = 7.
mkobjects.gain = 1.
mkobjects.rdnoise = 4.
mkobjects.poisson = no
mkobjects.seed = 1
mkobjects.comments = yes
#mkobjects hr_obj_nobias.fits star=moffat radius=1
mkobjects hr_obj_nobias.fits star=hr_psf.fits[0] radius=100
print ("Adding bias...")
imcalc hr_obj_nobias.fits[0] hr_obj.fits[0] "im1+5000"
print ("Making multiple raw images with different cosmic rays...")
mknoise.title = "HRC test image by Max Mutchler"
mknoise.ncols = 1062
mknoise.nlines = 1044
mknoise.header = "hrctext.hdr1"
mknoise.background = 0.
mknoise.gain = 1.
mknoise.rdnoise = 0.
mknoise.poisson = no
mknoise.cosrays = ""
mknoise.comments = yes
# This creates three 600 second raw images, each with
# 3329 cosmic ray events affecting about 26,650 pixels:
#These cosmic rays affect about 1 pixel each
mknoise.ncosrays = 133
mknoise.energy = 1000.
mknoise.radius = 0.1

```

```

mknoise.ar = 1.
mknoise.pa = 0.
mknoise_hr_obj.fits output=hr1_art1.fits seed=11
mknoise_hr_obj.fits output=hr2_art1.fits seed=21
mknoise_hr_obj.fits output=hr3_art1.fits seed=31
#These cosmic rays affect about 8 pixels each
mknoise.ncosrays = 3063
mknoise.energy = 10000.
mknoise.radius = 0.36
mknoise.ar = 1.
mknoise.pa = 0
mknoise_hr1_art1.fits output=hr1_art2.fits seed=12
mknoise_hr2_art1.fits output=hr2_art2.fits seed=22
mknoise_hr3_art1.fits output=hr3_art2.fits seed=32
#These cosmic rays affect about 26 pixels each
mknoise.ncosrays = 67
mknoise.energy = 100000.
mknoise.radius = 2.0
mknoise.ar = 0.1
mknoise_hr1_art2.fits output=hr1_art3.fits seed=13 pa=30
mknoise_hr2_art2.fits output=hr2_art3.fits seed=23 pa=45
mknoise_hr3_art2.fits output=hr3_art3.fits seed=33 pa=60
#These cosmic rays affect about 23 pixels each
mknoise.ncosrays = 67
mknoise.energy = 1000000.
mknoise.radius = 3.0
mknoise.ar = 0.01
mknoise_hr1_art3.fits output=hr1_art.fits seed=14 pa=-45
mknoise_hr2_art3.fits output=hr2_art.fits seed=24 pa=-60
mknoise_hr3_art3.fits output=hr3_art.fits seed=34 pa=-30
print ("Copying simulated images into raw images...")
!cp hr_raw.fits hr1_raw.fits
!cp hr_raw.fits hr2_raw.fits
!cp hr_raw.fits hr3_raw.fits
imcopy hr1_art.fits[0] hr1_raw.fits[sci,1,overwrite]
imcopy hr2_art.fits[0] hr2_raw.fits[sci,1,overwrite]
imcopy hr3_art.fits[0] hr3_raw.fits[sci,1,overwrite]
print ("Removing temporary files...")
del hr*_art*.fits
print ("Setting overscan level...")
imreplace.imaginary = 0.
imreplace.radius = 0.
imrep hr1_raw.fits[sci,1][1:1062,1025:1044] 5000.
imrep hr2_raw.fits[sci,1][1:1062,1025:1044] 5000.
imrep hr3_raw.fits[sci,1][1:1062,1025:1044] 5000.
imrep hr1_raw.fits[sci,1][1:19,1:1044] 5000.
imrep hr2_raw.fits[sci,1][1:19,1:1044] 5000.
imrep hr3_raw.fits[sci,1][1:19,1:1044] 5000.
imrep hr1_raw.fits[sci,1][1044:1062,1:1044] 5000.
imrep hr2_raw.fits[sci,1][1044:1062,1:1044] 5000.

```

```

imrep hr3_raw.fits[sci,1][1044:1062,1:1044] 5000.
print ("Setting amp, gain, and exposure time in raw images...")
hedit hr?_raw.fits[0] CCDAMP D
hedit hr?_raw.fits[0] CCDGAIN 2
hedit hr1_raw.fits[0] EXPTIME 600
hedit hr2_raw.fits[0] EXPTIME 600
hedit hr3_raw.fits[0] EXPTIME 600

```

compare.cl

```

print ("Stacking raw images and trimming overscans to register with output images...")
del hr*stack*.fits
imcalc hr1_raw.fits[sci,1],hr2_raw.fits[sci,1],hr3_raw.fits[sci,1] hr123_stack.fits "im1+im2+im3"
imcopy hr123_stack.fits[0][20:1043,1:1024] hr123_stackT.fits
print ("Making CR masks for each of the raw images...")
del hr*mask*.fits
imcalc hr1_raw.fits[sci,1],hr_obj.fits[0] hr1_mask.fits "im1-im2"
imcalc hr2_raw.fits[sci,1],hr_obj.fits[0] hr2_mask.fits "im1-im2"
imcalc hr3_raw.fits[sci,1],hr_obj.fits[0] hr3_mask.fits "im1-im2"
print ("Making a composite CR mask and trimming overscans...")
imcalc hr1_mask.fits[0],hr2_mask.fits[0],hr3_mask.fits[0] hr123_mask.fits "im1+im2+im3"
imcopy hr123_mask.fits[0][20:1043,1:1024] hr123_maskT.fits
print ("Scaling and trimming the true image to match the output images...")
del hr*objS.fits
imcalc hr_obj_nobias.fits[0] hr12_objS.fits "im1*2"
imcalc hr_obj_nobias.fits[0] hr123_objS.fits "im1*3"
del hr*objT.fits
imcopy hr12_objS.fits[0][20:1043,1:1024] hr12_objT.fits
imcopy hr123_objS.fits[0][20:1043,1:1024] hr123_objT.fits
print ("Subtracting the true image from the output images...")
del hr*residual.fits
imcalc hr12_crj.fits[sci,1],hr12_objT.fits[0] hr12_residual.fits "im1-im2"
imcalc hr123_crj.fits[sci,1],hr123_objT.fits[0] hr123_residual.fits "im1-im2"

```