**Instrument Science Report NICMOS 2010-001**

# Finesky – removing higher order sky residuals

Tomas Dahlen and David Grumm
July 02, 2010

**ABSTRACT**

*We report on a new IRAF task called* finesky *that removes higher order sky residuals in NICMOS images by creating a masked median image of the observed sky. This median sky image is thereafter subtracted from the science images. A residual signal after image processing using the calibration software* calnica *may be present due to reference files that do not sufficiently match the conditions of the observations. This includes a slight mismatch in the dark current or the flat-field corrections. The task described here can also used to create sky flat-field images.*

## Introduction

There are cases where fully reduced NICMOS images show signs of a residual signal even after the images have been processed through the full set of available reduction software. These residuals may be due to a slight mismatch between the true dark signal and the dark reference file. In particular, uncorrected signal due to amplifier glow or shading may be present. Residuals can also be due to non-optimal flat-fielding, reflecting that the color of the observed sky differs from the color of the internal lamps used to produce the reference flat-fields. Complicating an accurate correction by reference files is the observed change in A-to-D conversion with time in the NICMOS detectors. In the era prior to the installation of the NICMOS Cooling System (NCS), this was due to a change in detector temperature as the solid nitrogen evaporated. For the NCS era, 2002-2008, the A-to-D change was due to a slight drift in the bias voltage. Even though the *calnica*

software has been updated to use reference files that corrects for these changes, there are still cases where a non-optimal image reduction leads to residuals in the final calibrated images.
In this report, we describe a new task called *finesky* that creates a residual sky image from a stack of fully calibrated NICMOS images. The residual sky image is thereafter subtracted from each individual image in order to eliminate as much as possible of the higher order residual signal. Even though the task is developed for NICMOS, it should be straightforward to use it also for other *HST* instruments. In the absence of residuals due to non-optimal reference calibration files, the task can be used to create sky flat-field images.

The tasks needed for running *finesky* are available in the STSCI_PYTHON package and can be downloaded from the STScI webpage (http://www.stsci.edu/resources/software_hardware/pyraf/stsci_python). In house at STScI, the script is available on unix machines (linux) and can be run either on the command line or within the Python environment, as shown in examples below.

## Removing higher order sky signal

Two prerequisites for creating an accurate sky residual image are that there exists a sufficient number of individual images to combine and that the area of the field not covered by objects is sufficiently large. Images should also preferentially be dithered so that all pixels of the detector sample the sky in at least one or a few exposures. Therefore, typical programs where this task can be used are surveys of sparse fields.

There are a number of steps involved in producing and subtracting the residual sky signal. In Figure 1, we show a flow chart outlining the different steps. Note that the procedure is interactive and will at some points rely on the subjective judgments of the user, e.g., how many iterations are needed in order to achieve an optimal subtraction and which sigma threshold to be used for masking objects.

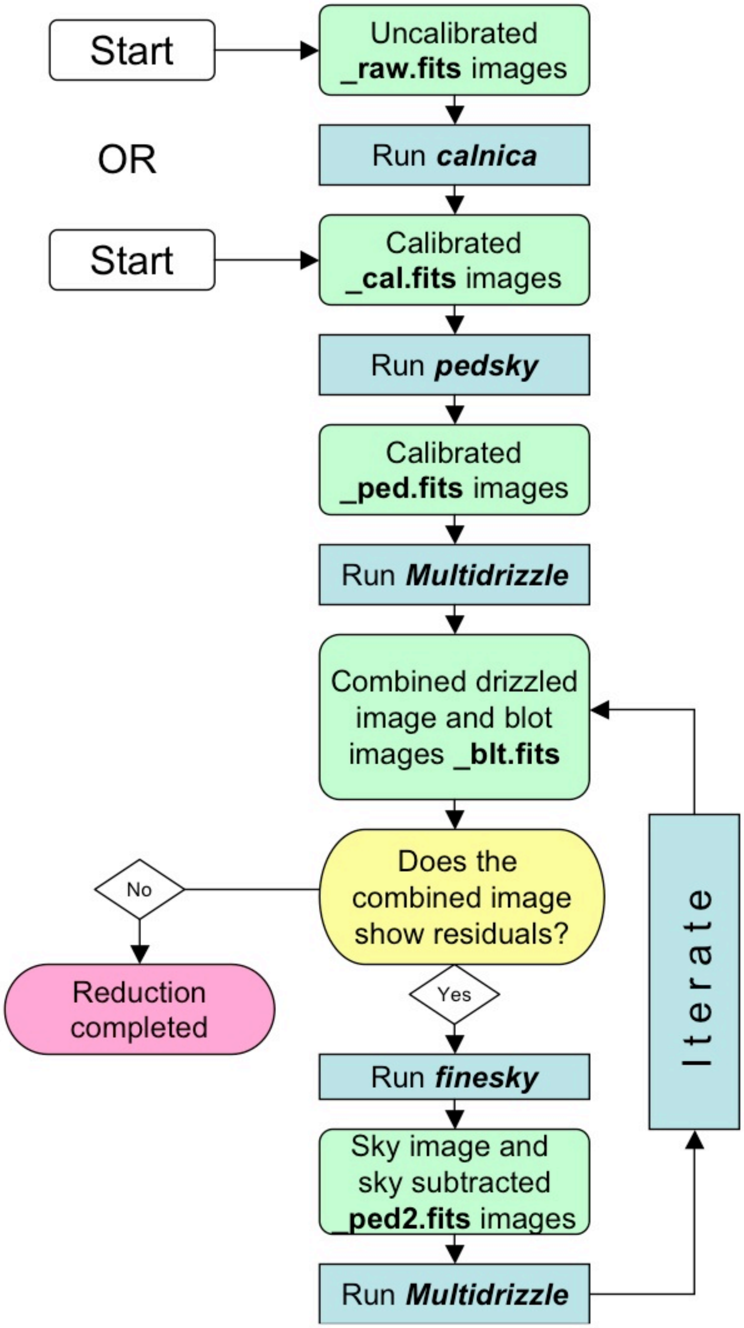

Start
Uncalibrated _raw.fits images
Run calnica
OR
Start
Calibrated _cal.fits images
Run pedsky
Calibrated _ped.fits images
Run Multidrizzle
Combined drizzled image and blot images _blt.fits
Does the combined image show residuals?
No
Reduction completed
Yes
Run finesky
Sky image and sky subtracted _ped2.fits images
Run Multidrizzle
Iterate

Figure 1. Flow chart describing the different steps for creating and subtracting a residual sky image.

### *Initial steps*

The first step in this procedure is to run the un-calibrated `_raw.fits` files through the *calnica* software to produce calibrated `_cal.fits` files. Thereafter, the `_cal.fits` files should be run through the *pedsky* software, which calculates and removes a sky signal and a quadrant-dependent residual bias (or "pedestal") signal in NICMOS images. Instead of starting with the `_raw.fits` files, already calibrated `_cal.fits` files retrieved from the On-the-Fly Reprocessing (OTFR) pipeline can be used. However, the *pedsky* task is not run by the OTFR and will therefore have to be applied. In the following, we use the extension `_ped.fits` for images that have been calibrated through both *calnica* and *pedsky*.

At this stage, a visual inspection of the calibrated images should reveal if there is any higher order residual sky signal left and that the calibration therefore should benefit from applying the higher order sky subtraction. Figure 2 shows an example of a calibrated image before and after *pedsky* removal. In this image, the sky and pedestal features are particularly evident (left panel), but even after running *pedsky* there are some residual features in the background (right panel). There is a vertical band of elevated sky in the right part of the image and there is also a faint horizontal band above the midpoint of the image. This example is taken from the NICMOS imaging of HDF-N (Program 7235, PI: Thompson).

If there is a residual signal even after running *pedsky*, such as in Figure 2, the reduction should likely benefit if a higher order sky image is subtracted. However, even if there is not an obvious residual, you may still want to proceed to produce a sky image, just to verify that this signal is insignificant.

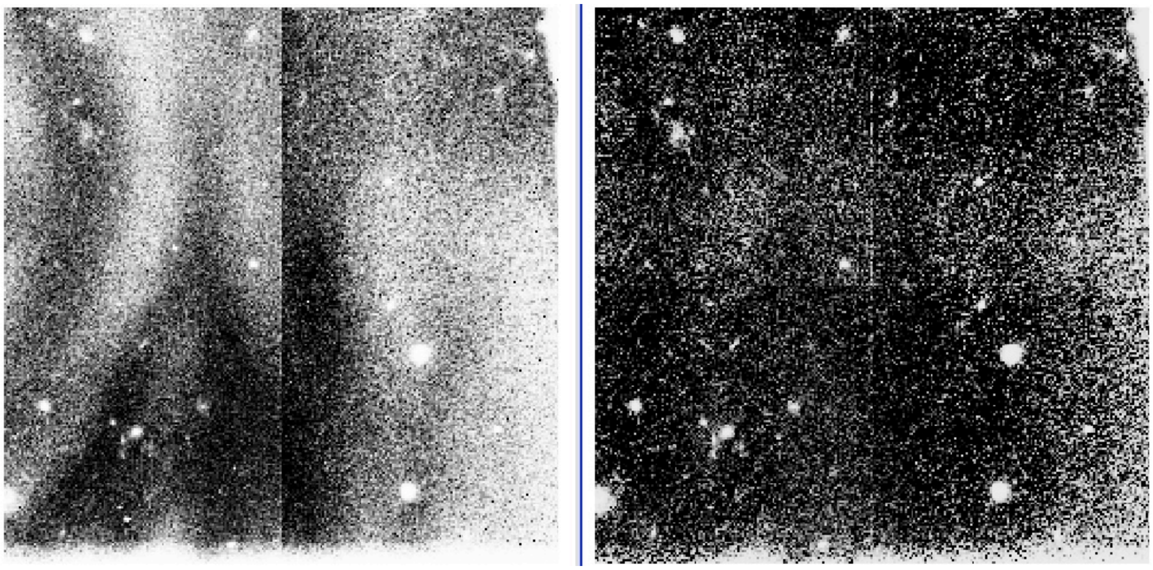


Figure 2. Left panel shows a calibrated file retrieved from the OTFR. There are clear signals from sky and quadrant dependent bias offsets. Even after running *pedsky* (right panel), there remain some higher order sky residuals.

### *First* Multidrizzle *run*

Prior to running the actual *finesky* task, the dithered `_ped.fits` files should be run through *Multidrizzle* to produce a combined image from the stack of available individual images. For information on *Multidrizzle* see **http://stsdas.stsci.edu/multidrizzle/**. The next step in the procedure is to use the blot images created by *Multidrizzle*, therefore set "blot=yes" when running the code. The blot images (hereafter referred to as *_blt.fits* files) have the same size and covers the same area as each of the individual images, but have the depth of the full combined image. Inspecting the output drizzled image should further reveal the extent of the residuals.

### *Running* Finesky

The *finesky* task is now ready to create a residual sky image. First the *_blt.fits* files are used to create an individual mask image for each of the *_ped.fits* files. This is done by masking pixels that are above the mean rms of the image by some predefined sigma value. The sigma value is set to 0.5 by default, but can be changed by the user to be more or less restrictive. The masked objects are thereafter grown in size to also mask wings of the objects profiles that fall below the sigma threshold. The benefit of using the blot images instead of the *_ped.fits* files to create the individual masks is that the blot images have the depth of the full combined image. Therefore, objects that may be too faint to be detected in individual exposures can be effectively masked out when using the blot images. All *_ped.fits* images are thereafter stacked together after masking out objects and for each pixel, a median is calculated from all files having a valid value for that pixel. For a pixel having no valid values, the mean of the valid values among the 8 immediate neighbors (a radius of 1) is used. If there are no valid values among the 8 immediate neighbors, the mean of the valid values among the neighbors within a radius of 2 is used. If there are still no valid values, the mean of the valid values among the neighbors within a radius of 3 is used. If there still are no valid values, the pixel is assigned a zero value, i.e., no sky will be subtracted at that pixel. When running the task, *finesky* outputs the number of pixels that have no valid values among all images. Also, a warning is issued if the number of pixels with no valid values exceeds 1000.

The *finesky* script also creates copies of the input *_ped.fits* files where the sky image has been subtracted. These subtracted images are assigned an extension *_ped2.fits*. As mentioned above, the *finesky* task can be used to create sky flat-field images. To do this, omit the flat-field correction when running *calnica* (or the corresponding task for other HST instruments), by editing the header in the *_raw.fits* files to "FLATCORR=OMIT". The *pedsky* run will also have to be skipped in this case since that task otherwise will interpret the flat-fields signal as an unwanted sky signal and remove it. The output sky image from the *finesky* task will in this case be a sky flat-field.

### *Second* Multidrizzle *run*

The next step is to run *Multidrizzle* using the files that have had the sky image subtracted. Inspection of the final drizzled image should reveal how well the residuals have been subtracted. In most cases, this should produce an image where most of the unwanted signal is removed. However, if there still are residuals, then a second iteration of *finesky* can be applied. In this case, use the subtracted *`_ped2.fits`* files and the *`_blt.fits`* files from the second *Multidrizzle* run as input to *finesky* and rerun the script.

## Example

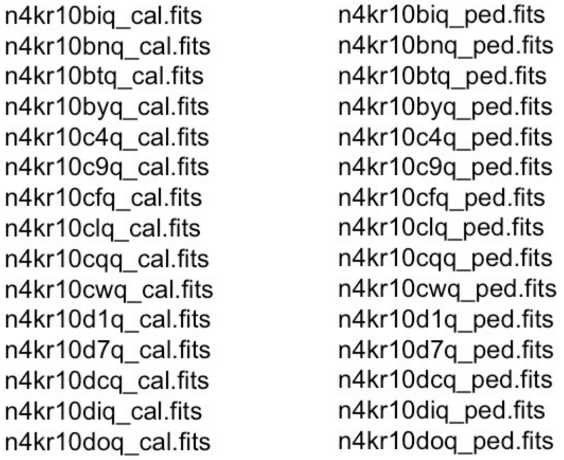
To illustrate how *finesky* works, we will go through the different steps using a set of images from the NICMOS observations of HDF-N, program 7235 (PI Thompson). We start with a set of 15 calibrated *`_cal.fits`* images (which is only a small subset of images available from that program). These could either be files directly retrieved from the OTFR, or *`_raw.fits`* files run through the *calnica* software by the user to produce the *`_cal.fits`* files.

1) Run *pedsky*.

Create two text files, one that lists the input *`_cal.fits`* files and one with names to call to output files from *pedsky*. Here we chose to assign the extension *`_ped.fits`* to the files that have been run through *pedsky*. The content of the input text file “list_cal.txt” is shown in the left column of Figure 3, while the output files that will be created by *pedsky* are listed in the text file “list_ped.txt”, shown in the right column.

The *pedsky* task is found in the *hst_calib.nicmos* package of *stsdas*. The syntax of running the script on the images listed in the text file is:

```
> pedsky @list_cal.txt @list_ped.txt
```



```
n4kr10biq_cal.fits      n4kr10biq_ped.fits
n4kr10bnq_cal.fits      n4kr10bnq_ped.fits
n4kr10btq_cal.fits      n4kr10btq_ped.fits
n4kr10byq_cal.fits      n4kr10byq_ped.fits
n4kr10c4q_cal.fits      n4kr10c4q_ped.fits
n4kr10c9q_cal.fits      n4kr10c9q_ped.fits
n4kr10cfq_cal.fits      n4kr10cfq_ped.fits
n4kr10clq_cal.fits      n4kr10clq_ped.fits
n4kr10cqq_cal.fits      n4kr10cqq_ped.fits
n4kr10cwq_cal.fits      n4kr10cwq_ped.fits
n4kr10d1q_cal.fits      n4kr10d1q_ped.fits
n4kr10d7q_cal.fits      n4kr10d7q_ped.fits
n4kr10dcq_cal.fits      n4kr10dcq_ped.fits
n4kr10diq_cal.fits      n4kr10diq_ped.fits
n4kr10doq_cal.fits      n4kr10doq_ped.fits
```

Figure 3. Left column lists the input images in the text file “list_cal.txt”, while the right

column lists the output files in the text file “list_ped.txt”.

2) Run *Multidrizzle*.

The *Multidrizzle* task can be found in the *dither* package of *stsdas*. The input files to *Multidrizzle* are the same as the images in the output file “list_ped.txt” from the run of the *pedsky* task. In this example, we run *Multidrizzle* with default settings, except that we specify that blot files should be created. We call the combined drizzled image “testimg”. The call to *Multidrizzle* will then be:

```
 > multidrizzle @list_ped.txt output='testimg' blot='yes'
```

In Figure 4 we show the resulting combined output image from *Multidrizzle* (testimg.fits). It is evident from the figure that there are uncorrected residuals affecting the image quality. Note that the blot files created by *Multidrizzle* automatically get extension `_sci1_blt.fits`. For example, the blot file corresponding to the first input file n4kr10biq_ped.fits is named n4kr10biq_ped_sci1_blt.fits.

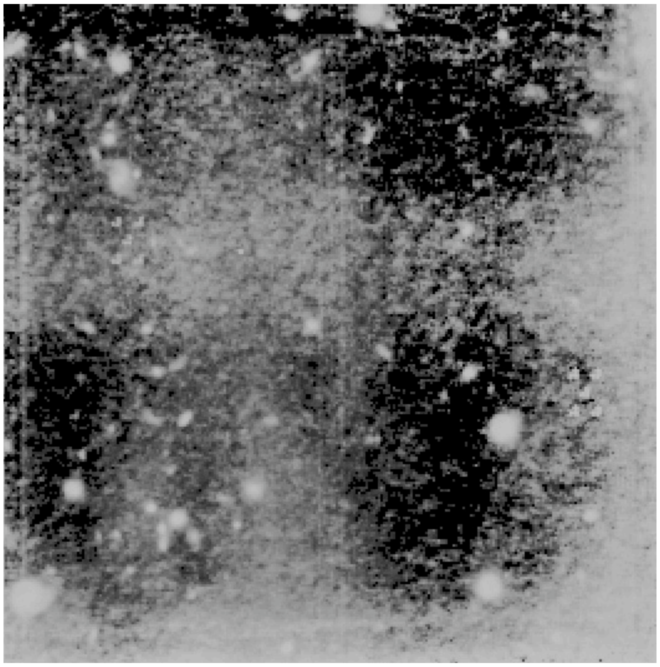


Figure 4. Result from combining 15 individual images. Uncorrected residuals are evident.

3) We are now ready to run the *finesky* task. Input to the task is the same list of file names as used for input to *Multidrizzle*, here “list_ped.txt”. The *finesky* task requires that *Multidrizzle* has been run and that blot files corresponding to each files in the input list have been created. Furthermore, the output name of the median sky image has to be specified. Here we choose to name the output sky image “medsky.fits”. The

sigma threshold for masking objects is set to 0.5 by default. In this example we choose to try a slightly higher threshold of 0.7. Finally, the task is run in "quiet" mode by default, but we choose to change this to "verbose". Running the script with the settings specified above from the (linux) command line looks like:

```
> finesky.py -t 0.7 -m "medsky.fits" -c "list_ped.txt" –v
```

The resulting sky image is plotted in Figure 5. This shows signs of both residual sky and uncorrected flat-field features. The output subtracted individual images created by *finesky* have extensions `_ped2.fits`. The same example as above run within pyraf looks like:

```
> import finesky

> m_mask = finesky.Makemedmask( medfile='medsky.fits', thresh
= 0.7, verbosity = 1,  callist='list_ped.txt')

> m_mask.makemask()
```

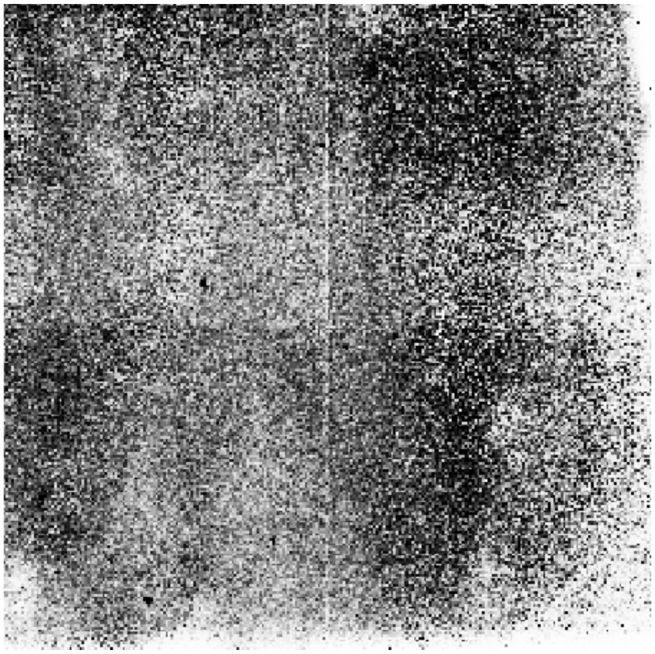


Figure 5. Sky image produced from a median of 15 masked input images.

4) As a final step we use *Multidrizzle* to produce a combined image using as input the sky-subtracted images produced by *finesky*. The list of input file names will therefore be the same as the first *Mutidrizzle* run, except that "ped" is replaced by "ped2" for each image. We call this file "list_ped2.txt" and name the output

drizzled image “test2img”. Also, we choose to create blot files since these can be used if we choose to make a second iteration through *finesky*.

```
> multidrizzle @list_ped2.txt output='test2img' blot='yes'
```

In Figure 6 we show the combined image produced from the files that have had the higher order residuals removed by running the *finesky* task. Most of the residual signal visible in Figure 4 has been successfully removed. Closer inspection shows that there could still be some remaining signal and that a second iteration through *finesky* may be beneficial. In that case, restart at point 3) above. Input to the *finsky* task will in this case be the `_ped2.fits` files produced in the first *finesky* run and the `_ped2_sci1_blt.fits` files produced in the second *Multidrizzle* run. Note that *finesky* will append a “2” to the input filenames when creating the new subtracted individual images. Therefore, when we start with files named `_ped2.fits`, the output from *finesky* will be called `_ped22.fits`. It is also recommended to redo the calculation trying different values for the sigma threshold to evaluate how this affects the removal of the residuals.

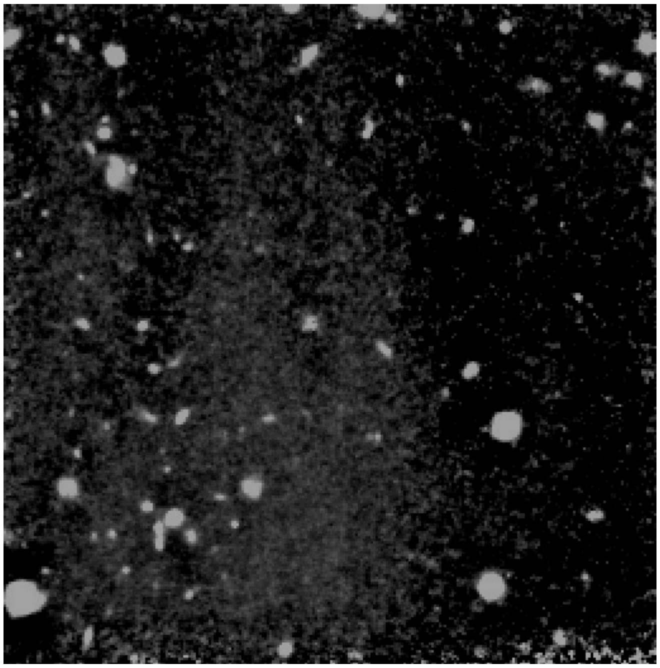


Figure 6. Combined image produced from individual images that have had the higher order sky signal subtracted. Compare with Figure 4.

***Making a sky flat-field image***

To make a sky flat-field image, update the headers of the `_raw.fits` files by setting FLATCORR=”OMIT” using e.g., iraf task *hedit*. Thereafter run *calnica*, followed by

*Multidrizzle* and *finesky* as described above. However, note that the *pedsky* task should not be run since this will interpret the flat-field signal as a sky background and subtract it. The output sky image from *finesky* will be the flat-field. The resulting flat-field will have to be properly normalized (e.g., to unity) before they can be used. Note also that the subtracted individual images produced by *finesky* do not have any scientific value since the flat-field has been subtracted from the images (instead of the images being divided by the flat-field to produce a calibrated image.) The left panel of Figure 7 shows a sky flat-field produced from the 15 science images from program 7235, while the right panel shows the actual flat-field produced by internal flat-field lamps and used by the OTFR pipe-line. The two images show the same structure (although displayed with slightly different stretch).

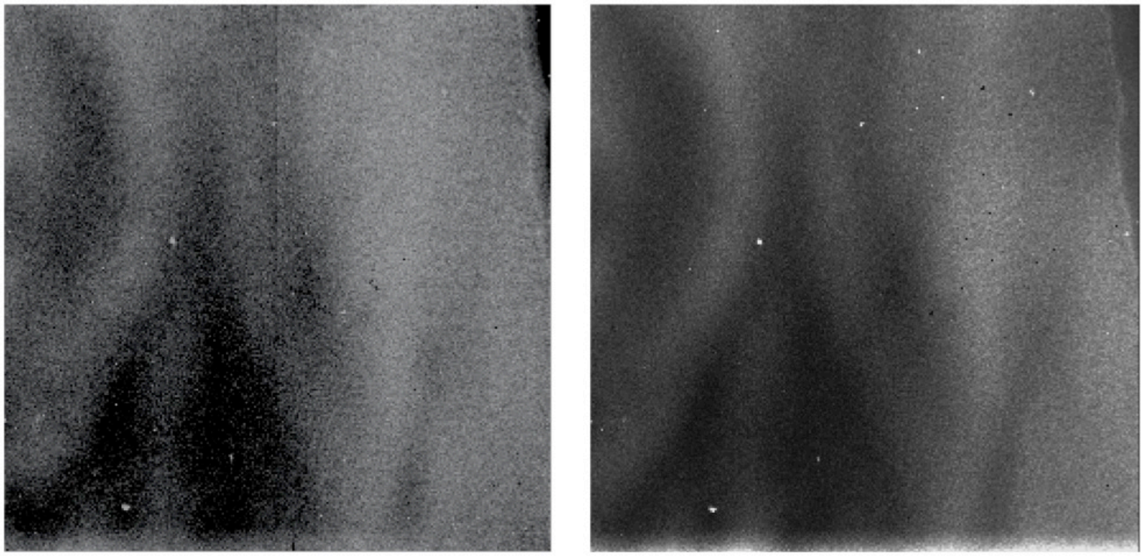


Figure 7. Left panel shows a sky flat-field image produced by finesky, while the right panel shows the flat-field produced by internal lamps and used by the OTFR.

## Summary

This report describes the new task *finesky* that removes higher order sky residuals that may be present in NICMOS images after standard reduction using e.g., the *calnica* and *pedsky* software. The *finesky* task can also be used to produce sky flat-field images.