WFC3/UVIS Alignment of Sparse Fields Using Headerlets

Introduction

TweakReg is generally quite efficient at aligning images. There are, however, cases where additional steps are required to refine image alignment. In this example, image alignment fits are determined using (1) *FLT* images with most cosmic rays removed, and (2) by masking areas where TweakReg produces spurious sources. The resulting updated WCS information is saved to headerlets that are applied to the original *FLT* files before they are drizzle-combined by AstroDrizzle.

In this scenario, a user is obtaining a series of observations of a tidal disruption event. Each observation is a set of dithered images that are drizzled-combined to create an image suitable for analysis. The user wishes to look for changes in the target by subtracting drizzle-combined images from each epoch. Therefore, the best possible alignment is required, for drizzle-combined images between epochs and for *FLT* images within a given epoch.

The data in this example represents an observation from one of the epochs: a 3-point dither using the WFC3/UVIS. There are few stars in the field and the images are heavily affected by cosmic rays. Features in the prominent diffraction spikes of two very bright stars are erroneously identified as sources by TweakReg. These conditions result in a high RMS for the image alignment fit.

When TweakReg is run on the original *FLT* images, typical alignment fit RMS values are much greater than 0.1 pixels, too high for the science requirements of this study. This is because there are too few stars to compensate for errors, introduced by cosmic rays and spurious sources, that are treated as sources in the alignment fit. However, using the methods in this example, the alignment fit RMS can be reduced down to ~0.03 pixels.

There are two ways to reduce this alignment fit RMS dramatically. In most cases, both techniques are required since one method alone has less effect. They are:

I. Remove cosmic rays from each *FLT* image using AstroDrizzle so that TweakReg operates only on real sources.
II. Exclude regions of the chip with spurious sources so that TweakReg operates only on real sources.

Summary of Steps

1. Description of the Data.
2. Run TweakReg on the original *FLT* data to see the alignment fit RMS values.
3. Run AstroDrizzle on the data to create *FLT* images free of cosmic rays (with suffix *crclean.fits*).
4. Run TweakReg on cosmic ray-cleaned images to create object catalogs to help identify spurious sources.
5. Use ***ds9*** to create regions files around bright stars so TweakReg can exclude those areas during the source-finding stage.
6. Run TweakReg again using the cosmic ray-cleaned images and regions files to determine more accurate image alignments.
7. If the alignment fit RMS values are acceptable, run TweakReg one last time, using the optimal settings from the previous step, to create headerlets containing new WCS information.
8. Apply the new WCS keyword values in the headerlets to the original *FLT* images so that the newly-derived WCS becomes the primary WCS for the *FLT* files.
9. Run AstroDrizzle on the WCS-updated *FLT* images.

Notes:

(1) If you are working through this exercise, expect to see small insignificant differences in values compared to those in this example. This is due to running slightly different versions of the software, or using different operating systems. This example used TweakReg 1.0.2 and AstroDrizzle 1.0.2. Software version numbers can be obtained as follows:

--> from drizzlepac import tweakreg, astrodrizzle
--> print tweakreg.__version__,tweakreg.__vdate__
1.0.2 11-Apr-2012
--> print astrodrizzle.__version__,astrodrizzle.__vdate__
1.0.2 13-July-2012

(2) DrizzlePac help files can be obtained in the following ways:

- Click the Help button at the top right of the task GUI
- At the PyRAF command line:
  --> help imagefindpars
- At the Python and PyRAF command lines:
  --> from drizzlepac import tweakreg
  --> tweakreg.help()

1. Description of Data

The target GRB-110328A was observed on April 4, 2011, as a 3-point dither using WFC3/UVIS in the F606W filter. This data is from proposal 12447, Visit 1, line 2. Additional information is shown in the table below.

Image Name	Association ID	POS TARG (arcsec)	PA_V3	Exposure Time (s)
ibof01aoq_flt.fits	IBOF01020	0.000,0.000	57.0188	420.00
ibof01aqq_flt.fits	IBOF01020	1.446,2.926	57.0187	420.00
ibof01asq_flt.fits	IBOF01020	-1.446,-2.926	57.0187	420.00

2. Preliminary Image Alignment with TweakReg

In this section, three original unaltered *FLT* images from the Archive are aligned using TweakReg, just to see the results without cosmic ray removal and the masking of spurious objects.

Notes:

(1) Throughout this example, the Python TEAL GUI parameter editor will be used by both TweakReg, ImageFindPars, and AstroDrizzle, and referred to as the “TweakReg GUI,” “ImageFindPars GUI,” and “AstroDrizzle GUI.”
(2) In this example, commands preceded by the prompt “**-->**” are executed in the PyRAF environment.

To load DrizzlePac in the PyRAF environment:

Load DrizzlePac
--> import drizzlepac

Open the TweakReg GUI
--> epar tweakreg

In the GUI:
First, click on the ***Defaults*** button on the upper right of the window to reset the parameters. Then, set the following non-default parameter values:

- Click on the ***imagefind Parameters*** button (third line in the TweakReg GUI list).
  This opens a new window that shows parameters for the ImageFindPars task; these are settings for the star-finding algorithm. In the ImageFindPars GUI, click the ***Defaults*** button at the upper right of the window to reset the parameters. Then,
  - Set ***peakmax*** to ***50000***, to avoid saturated stars.
  - Set ***threshold*** to ***15*** to avoid most faint extended objects.
  - Set ***fluxmin*** to ***3000*** to avoid very faint sources that are problematic for position measurements.
  - Click ***Save & Quit*** to return to the TweakReg GUI.
- In the ***OBJECT MATCHING PARAMETERS*** section of the TweakReg parameters (near the bottom of the window) set ***minobj*** to ***10*** because this is a sparse field.
- Click the ***Execute*** button on the upper left of the GUI.

As TweakReg executes, it will display plots showing offsets and residuals for each image-reference image alignment fit. After inspecting each set, type ***n*** in the PyRAF window to continue to the next set of plots.

Note:

> ImageFindPars parameters for this example were obtained after several test runs to maximize the selection of suitable objects for image alignment. Please refer to the TweakReg and ImageFindPars help files for additional information.

When a wildcard is specified, i.e., **flt.fits*, the images are "listed" in alphabetical and numeric order. Image *ibof01aoq_flt.fits* is the first image on the "list" and, by default, is chosen as the reference image. TweakReg provides offset values between the reference image and each of the other dithered images: *ibof01aqq_flt.fits* and *ibof01asq_flt.fits*. Alignment information is recorded in a file with the suffix *catalog_fit.match*, one for each non-reference image. Each file also contains a list of objects used to determine the alignment fit.

An excerpt of *ibof01aqq_flt_catalog_fit.match* containing the offsets and the first three lines of coordinate matches found by TweakReg is shown below.

```
# Input image: ibof01aqq_flt
# Coordinate mapping parameters:
#    X and Y rms:        0.621482        0.568678
#    X and Y shift:     -0.0488541       -0.179985
#    X and Y scale:        1.00003         1.00003
#    X and Y rotation:    1.50454e-05
#
# Input Coordinate Listing
#     Column 1: X (reference)
#     Column 2: Y (reference)
#     Column 3: X (input)
#     Column 4: Y (input)
#     Column 5: X (fit)
#     Column 6: Y (fit)
#     Column 7: X (residual)
#     Column 8: Y (residual)
```

```
#     Column 9: Original X (reference)
#     Column 10: Original Y (reference)
#     Column 11: Original X (input)
#     Column 12: Original Y (input)
#     Column 13: Ref ID
#     Column 14: Input ID
#     Column 15: Input EXTVER ID
#
    -329.740190   -2090.766749   -329.535921   -2090.993388   -329.273268   -2090.983179      0.262654      0.010209
1729.921153      2.961445    1732.437943      5.033883      0      1      1
    1500.721411   -1782.189669    1500.738541   -1782.128267    1500.764118   -1781.837943      0.025576      0.290324
3541.365766    205.647358    3543.705905    208.007432    331    265      1
    1103.891211   -1677.698401    1103.901008   -1677.634639    1103.930033   -1677.344904      0.029024      0.289735
3149.961958    330.967752    3152.293802    333.335565    567    439      1
...
```

An excerpt of *ibof01asq_flt_catalog_fit.match* showing the offsets and the first three lines of coordinate matches found by TweakReg is provided below.

```
# Input image: ibof01asq_flt
# Coordinate mapping parameters:
#    X and Y rms:       0.634409        0.547156
#    X and Y shift:      -0.177326       0.0137268
#    X and Y scale:              1               1
#    X and Y rotation:      0.00412133
#
# Input Coordinate Listing
#     Column 1: X (reference)
#     Column 2: Y (reference)
#     Column 3: X (input)
#     Column 4: Y (input)
#     Column 5: X (fit)
#     Column 6: Y (fit)
#     Column 7: X (residual)
#     Column 8: Y (residual)
#     Column 9: Original X (reference)
#     Column 10: Original Y (reference)
#     Column 11: Original X (input)
#     Column 12: Original Y (input)
#     Column 13: Ref ID
#     Column 14: Input ID
#     Column 15: Input EXTVER ID
#
   -1594.521850   -1899.555516   -1594.128803   -1899.713911   -1593.415265   -1899.992951      0.713538     -0.279040
464.976073    263.653398    469.993925    268.119804     448     363      1
   -1229.600830   -1812.419079   -1229.411259   -1811.488962   -1228.908959   -1810.653598      0.502300      0.835364
829.988232    329.549939    834.796452    335.107265     564     481      1
    1500.721411   -1782.189669    1500.806646   -1782.140145    1501.191258   -1781.989103      0.384612      0.151042
3541.365766    205.647358    3546.088759    210.296568    331    280      1

...
```

The *X,Y* RMS for the alignment fit between the reference image (*ibof01aoq_flt.fits*) and the two other images, *ibof01aoq_flt.fits* and *ibof01asq_flt.fits*, as shown in their *catalog_fit.match* files are, respectively, 0.621482, 0.568678 and 0.634409, 0.547156. Clearly these alignment fits need to be improved

Figure 1A: Top Section of the TweakReg GUI

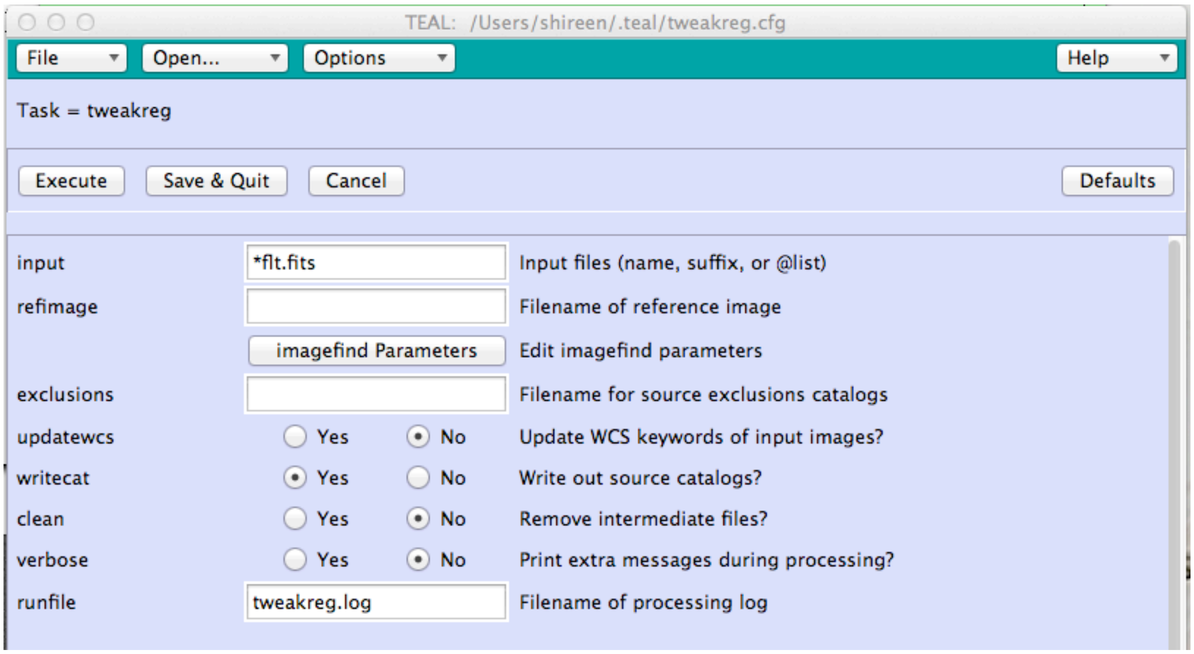
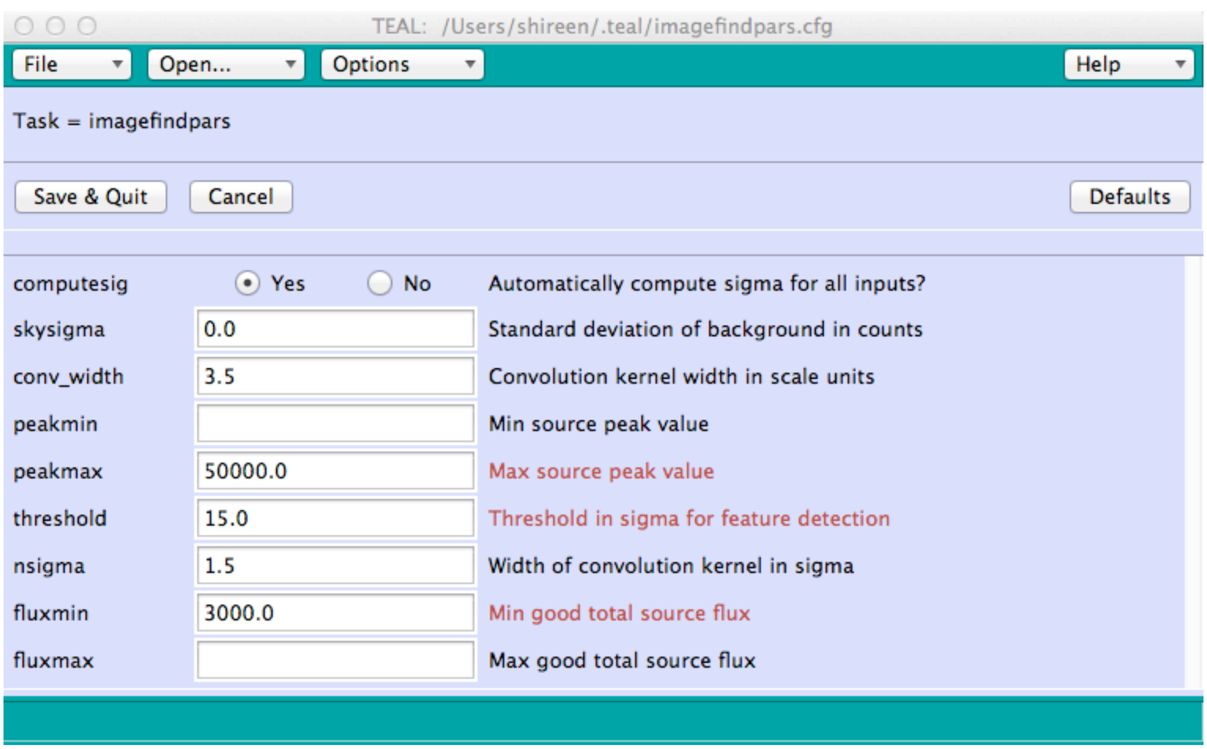


Figure 1B: ImageFindPars GUI Showing all Parameters
For non-default parameter values, the parameter descriptions are in red.



3. Remove Cosmic Rays from *FLT* Images Using AstroDrizzle

In the previous section, TweakReg produced alignment fits with large residuals because the source-finding software was picking up fake sources like cosmic rays and other artifacts. If most cosmic rays could be removed, leaving mostly real objects for use in aligning the images, the TweakReg alignment fit RMS could be smaller.

AstroDrizzle can be used to create a close approximation to cosmic ray-cleaned *FLT* images. These "clean" images are created by combining WCS-aligned images to create a median image; the median image is then compared to input images to create a cosmic ray mask for each image. Values for pixels flagged as cosmic rays in each image are replaced with corresponding pixel values in the median image.

However, these cosmic ray-cleaned images are not suitable for science analysis because the images are aligned based only on their WCS information; that level of alignment may not be good enough for the science goals. These clean images, however, have very few remnant cosmic rays so TweakReg is able to work with real objects that will produce more accurate alignments.

AstroDrizzle is invoked in the PyRAF environment using the **epar** command to open the AstroDrizzle GUI.

--> epar astrodrizzle

First, click on the ***Defaults*** button on the upper right of the GUI to reset the parameters. Then, set the following non-default parameter values to create *FLT* images free of cosmic rays, with suffix *crclean.fits*.

- Under ***CUSTOM WCS FOR SEPARATE OUTPUTS***, set ***driz_sep_wcs*** to ***Yes***, and set ***driz_sep_scale*** to ***0.03333.*** The first setting tells AstroDrizzle that non-default parameters are being used in that section, and the second setting specifies the output image scale for each drizzled *FLT* image to be 0.03333 arcseconds/pixels. This scale was chosen because it samples the PSF fairly well for UVIS data, and is convenient because three pixels equal 0.1 arcseconds. The exact value of the output scale, however, is not critical, but using an output pixel size that well samples the PSF is recommended for cosmic ray rejection.
- Under ***REMOVE COSMIC RAYS WITH DERIV, DRIZ_CR***, set ***driz_cr_corr*** to ***yes***. This tells AstroDrizzle to create versions of the input *FLT* images that are cleaned of cosmic rays (these images will have suffix *crclean.fits*).
- Under ***DRIZZLE FINAL COMBINED IMAGE***, set ***driz_combine*** to ***No*** because a combined product is not needed at this stage.
- Click the ***Execute*** button to run AstroDrizzle.

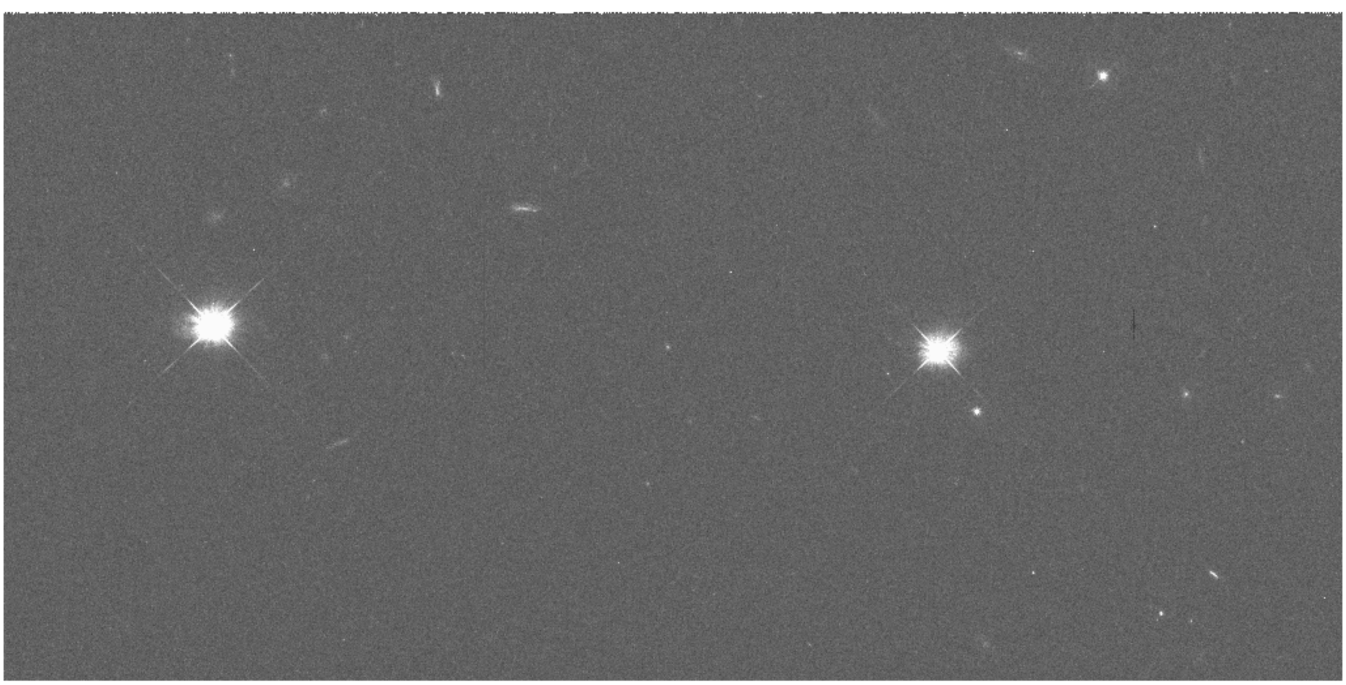
Note:

> Another way to apply parameter settings in a task GUI is to use a configuration file. This is a text file containing all task parameters values that can be loaded into the task GUI to set its parameter values.
>
> In this case, instead of individually setting parameter values in the AstroDrizzle GUI, an AstroDrizzle configuration file containing all the previously-used parameter settings, called *ad_crclean.cfg,* has been prepared for this example; it can be download from this example's page at the DrizzlePac website, and placed in the working directory.
>
> To use a configuration file, open the AstroDrizzle GUI as before.
>
> **--> epar astrodrizzle**
>
> Click on the ***Open*** pull-down menu (top-left of the window) and select ***Other*** to locate the *ad_crclean.cfg* configuration file. Select the configuration file and click ***OK*** to load the configuration file into AstroDrizzle. Then, click the ***Execute*** button in the AstroDrizzle GUI to run AstroDrizzle.
>
> To create a configuration file for future editing: under the GUI's ***File*** menu item, select ***Defaults***, then click ***Save As*** to save the parameter settings to a uniquely-named text file with suffix *cfg*.

When AstroDrizzle has completed, display and blink the original *FLT* images with the corresponding *crclean.fits* images to verify there are no anomalies before proceeding to the next step.

Figure 2a: Image *ibof01aoq_flt.fits[1]* Shows Heavy Cosmic Ray Artifacts

Figure 2b: Image *ibof01aoq_crclean.fits[1]* Shows a “Clean” Image With Most Cosmic Rays Removed



4. Run TweakReg on Cosmic Ray-Cleaned Images to Create New Catalogs

Run TweakReg again, using identical parameters as in step 1, except set ***input*** to ****crclean.fits***. The residuals from running cosmic ray-cleaned files in TweakReg may not be much better, perhaps even worse. But the purpose of this step is to generate a catalog of objects used for the alignment fit--this is needed for the next step.

The alignment fits RMSs for this TweakReg run are shown below, they're still large.

Excerpt of *ibof01aqq_crclean_catalog_fit.match*:

```
# Input image: ibof01aqq_crclean
# Coordinate mapping parameters:
#    X and Y rms:        0.437755        0.436666
#    X and Y shift:      0.0119763       0.00583629
#    X and Y scale:       0.999978        0.999978
#    X and Y rotation:    0.000679112
```

Excerpt of *ibof01asq_crclean_catalog_fit.match*:

```
# Input image: ibof01asq_crclean
# Coordinate mapping parameters:
#    X and Y rms:        0.329246        0.268464
#    X and Y shift:     0.00260892        0.112075
#    X and Y scale:        1.00005         1.00005
#    X and Y rotation:     0.00130125
```

5. Create *ds9* Regions Files to Mask Spurious Sources

A catalog of sources, used for the alignment fit calculations, was created for each image in the previous step. In this step, those catalogs will be used to identify and record areas containing spurious sources so that those regions can be avoided in the next TweakReg run in step 6. These "exclusions regions" will be created using ***ds9***.

Load the first image, *ibof01aoq_crclean.fits[1]*, using the ***ds9*** interface:

- In ***ds9***, click ***File*** in the main menu
- Select ***Open***, find and enter the image name and group, *ibof01aoq_crclean.fits[1].*
- Click ***OK***.

Note:

> Do not load the image using the PyRAF **display** command because all WCS information will be lost in ***ds9***. The image must be directly loaded from the ***ds9*** window.

Hint:

> To scale the image in ***ds9*** for better viewing, click ***Scale*** in the main menu and select ***Scale Parameters***. A pixel value histogram window will pop open. For this set of images, a ***Low*** value of ***-10*** and ***High*** of ***700*** worked well. Then, adjust the stretch in ***ds9*** to improve the image's appearance.

The catalog files, created in step 4, for *ibof01aoq_crclean.fits* are *ibof01aoq_crclean_sci1_xy_catalog.coo* and *ibof01aoq_crclean_sci2_xy_catalog.coo*, one for each image group.

For image *ibof01aoq_crclean.fits[1]*, overlay a plot of the objects in its corresponding source catalog, *ibof01aoq_crclean_sci1_xy_catalog.coo*. It contains the x,y coordinates of sources in the image.

- In the ***ds9*** menu, select ***Region*** and click on ***Load Regions***. This opens a pop-up window for finding the catalog file.
- At the bottom of the pop-up window, click on ***All*** to show all files in the working directory. Find *ibof01aoq_crclean_sci1_xy_catalog.coo* and click on it.
- A pop-up window labeled ***Load Regions*** will appear. In it, set ***Format*** to ***xy***, ***Coordinate System*** to ***image***, select ***Load into Current Frame***, then click ***OK*** to plot the objects on the image.

Figure 3A: Image *ibof01aoq_crclean.fits[1]* With an Overlay of Catalog Objects
The image below shows objects found by TweakReg in step 4, as recorded in catalog file *ibof01aoq_crclean_sci1_xy_catalog.coo*. Many spurious objects appear in the bright stars.

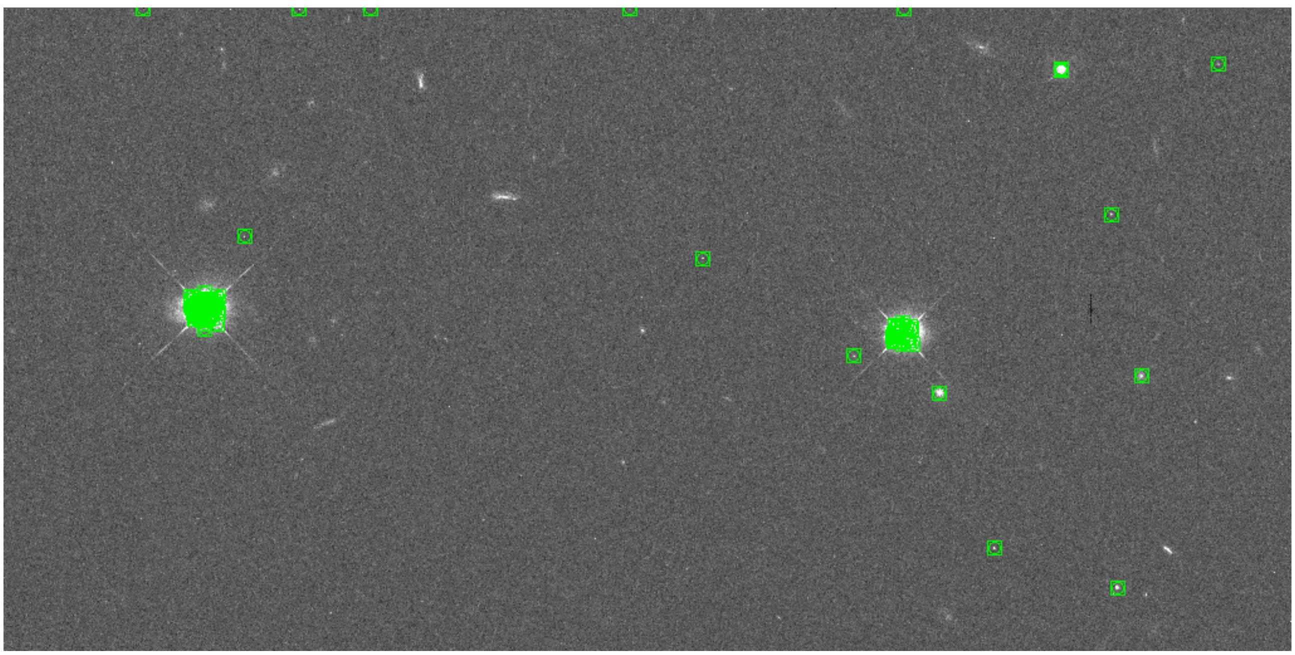
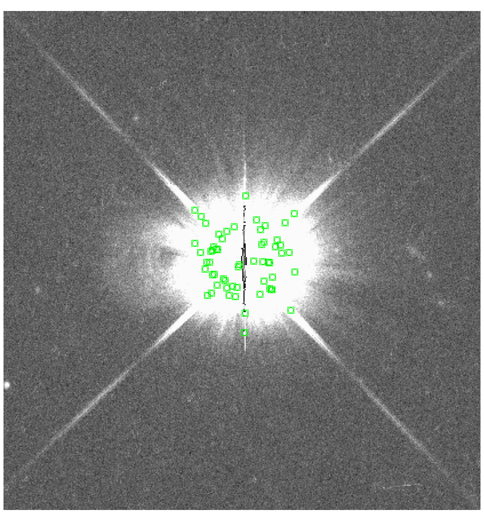


Figure 3B: Close-up of Left-most Bright Star Showing Spurious Sources Detected by TweakReg



It is evident that spurious sources detected in the bright stars in group 1 caused the large TweakReg alignment fit RMS values in step 4.

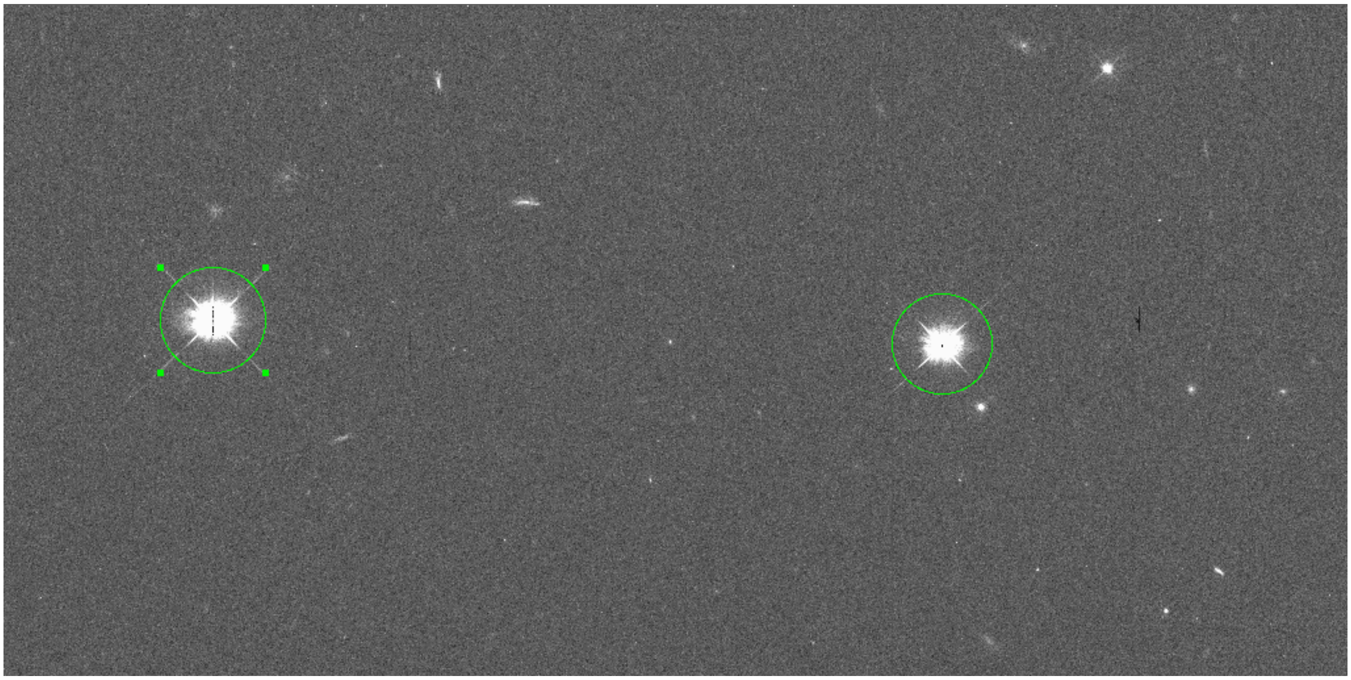
The alignment fit can, therefore, be improved by excluding spurious sources in those two bright stars. This is done by creating a ***ds9*** **“**regions file” to record the location around each of those bright stars, using the cursor to mark a circle that encompasses the star and most of its wings. (At this time, TweakReg only accepts circular regions.) That information is saved to a regions file in the form of a position and radius, one line for each star. This regions file will be used to mask the regions around the bright stars when TweakReg is run in step 6.

To create a regions file around the two bright stars:

- In the ***Region*** pull-down menu in ***ds9***, select ***Delete All Regions*** to remove the catalog plots.
- Under ***regions***, select ***shape***, then select ***circle***.
- Use the cursor to draw a circle around each of the two bright start in *ibof01aoq_crclean.fits[1]*, large enough to encompass sections of the diffraction spikes that may generate spurious sources.
- When both stars have been marked with circles, save the position and radius of the circles to a file: under the ***Region*** menu, select ***Save regions***. For this example, name it ***exclude.reg***. Click ***OK***.
- A pop-up menu appears; in it, select ***ds9*** for the format, ***WCS*** for the coordinate system, and click ***OK***. Since the regions file was saved in the WCS coordinate system (RA and Dec of a point, and a radius) this file can be used for all images in the next TweakReg run.

Figure 4: Exclusion Regions Around Bright Stars in Group 1 of the Images

Exclusion regions, in the format of a ***ds9*** regions file, allows the user to define areas that should be excluded by the TweakReg source-finding algorithm in step 6.



Shown below is the content of the regions file, *exclude.reg*, used for group 1 of all images:

```
# Region file format: DS9 version 4.1
# Filename: /Users/shireen/ASTRODRIZZLE/000HEADERLET/test_25jan13_iraf/ibof01aoq_crclean.fits[SCI]
global color=green dashlist=8 3 width=1 font="helvetica 10 normal" select=1 highlite=1 dash=0 fixed=0 edit=1 move=1
delete=1 include=1 source=1
fk5
circle(251.20165,57.56813,6.3482374")
circle(251.21233,57.591945,6.0029121")
```

Next, *check ibof01aoq_crclean.fits[4]*, the second science image group by opening it in ***ds9*** and loading its catalog, *ibof01aoq_crclean_sci2_xy_catalog*. Since there are no spurious objects in this field, a regions file is not needed.

6. Run TweakReg On "Clean" Images Using the *ds9* Exclusion File

TweakReg will use the regions file generated in the previous step as exclusion zones for its source detection software. It accepts a file that lists each image and its corresponding exclusion file. For this set of images, the exclusions file list for this example, *excl_list*, is shown below.

```
ibof01aoq_crclean.fits exclude.reg
ibof01aqq_crclean.fits exclude.reg
ibof01asq_crclean.fits exclude.reg
```

Notes:

The format of an exclusions list is one image per line. The first column contains the image name, the second column has the regions file for group 1 (sci,1), and the third column has the regions file for group 4 (sci,2).

- If an image only has a regions file for group 1, the regions file is named in the second column, and the third column is left blank in the exclusions list:
  ```
  image1_flt.fits  region1.reg
  ```

- If an image only has a regions file for group 4 (sci,2), it would appear in the exclusions list as
  ```
  image2_flt.fits  none  region2.reg
  ```

- If an image does not have a regions file, but other images in the list have them, the image without the regions file (image3_flt.fits below) should still appear in the exclusions list.
  ```
  image1_flt.fits  region1.reg
  image2_flt.fits  none  region2.reg
  image3_flt.fits
  ```

Additional information can be found in Section 4.4.2 of the DrizzlePac Handbook.

Run TweakReg on the *crclean.fits* files again, with the same settings as step 4, but this time, set the TweakReg field ***exclusions*** to ***excl_list***.

After TweakReg has completed processing, an inspection of the *catalog_fit.match* files, excerpted below, shows that the alignment fit RMS has been significantly reduced.

Excerpt of *ibof01aqq_crclean_catalog_fit.match*

```
# Input image: ibof01aqq_crclean
# Coordinate mapping parameters:
#    X and Y rms:       0.0789606        0.0577324
#    X and Y shift:     -0.0105367        0.0569829
#    X and Y scale:              1               1
#    X and Y rotation:     0.000153791
```

Excerpt of *ibof01asq_crclean_catalog_fit.match*

```
# Input image: ibof01asq_crclean
# Coordinate mapping parameters:
#    X and Y rms:       0.0474837        0.0395003
#    X and Y shift:      0.0740501        0.0314989
#    X and Y scale:              1               1
#    X and Y rotation:           360
```

7. Use TweakReg to Create Headerlets

Now that a good alignment fit has been obtained using cosmic ray-cleaned images and regions files, the new WCS keyword values as a result of these alignment can be recorded in FITS files called "headerlets." These headerlets will later be used to transfer the new WCS keyword values to the original input *FLT* images.

The headerlets are created by running TweakReg on the *crclean.fits* files one last time, using identical parameters as step 6, except for these:

- In the ***UPDATE HEADER*** section, set ***updatehder*** to ***yes***.
- In the ***HEADERLET CREATION*** section, set ***headerlet*** to ***yes***.
- Since the plots were inspected in the previous run, they don't need to be viewed again.
  - Under ***OBJECT MATCHING PARAMETERS***, set ***see2dplot*** to ***No***
  - Under ***CATALOG FITTING PARAMETERS***, set ***residplot*** to ***No Plot***.

Upon completion, three headerlets are created by TweakReg, one for each image. The first headerlet belongs to the reference image. The other two contain updated SIP keywords for each image group.

ibof01aoq_crclean_hlet.fits
ibof01aqq_crclean_hlet.fits
ibof01asq_crclean_hlet.fits

The PyRAF **imhead** task can be used to show the SIP header keyword and their values. For example, to view them in group 4 (sci, 2) of image *ibof01aqq_flt.fits*:

--> imhead ibof01aqq_crclean_hlet.fits[2] l+

A Python command can also be used to view the same information.

--> import pyfits
--> pyfits.getheader('ibof01aqq_crclean_hlet.fits',ext=2)

8. Apply New WCS Information to Original *FLT* Images Using Headerlets

Headerlets contain the new WCS keyword values required to bring the two non-reference *FLT* images, *ibof01aqq_flt.fits* and *ibof01asq_flt.fits*, in alignment with the reference image *ibof01aoq_flt.fits*.

For each image, the task **apply_headerlet** replaces the original *FLT* primary WCS keyword values with those in the headerlet. This step must be done for each image, including the reference image, *ibof01asq_flt.fits*, to ensure that all three images have the same value for the header keyword "WCSNAME." **apply_headerlet** can be run using the **epar** command, once for each image-headerlet pair.

However, using the "epar" command to edit each image, especially if there are many of them, can become quite tiresome. A few lines of Python can accomplish this much faster. In the example below, note that there must be four spaces before the command on the 4th line. The "..." symbols indicate that Python is expecting additional input from the user.

```
--> import glob
--> from stwcs import wcsutil
--> for fname,hdrlet in zip(glob.glob('*flt.fits'),glob.glob('*hlet.fits')):
...     wcsutil.headerlet.apply_headerlet_as_primary(fname,hdrlet)
...     (type return)
```

The addition of the headerlet to the image can be seen using PyFITS.

```
--> import pyfits
--> pyfits.info('ibof01aqq_flt.fits')
```

```
Filename: ibof01aqq_flt.fits
No. Name       Type          Cards  Dimensions    Format
0   PRIMARY    PrimaryHDU    250    ()            int16
1   SCI        ImageHDU      209    (4096, 2051)  float32
2   ERR        ImageHDU      78     (4096, 2051)  float32
3   DQ         ImageHDU      70     (4096, 2051)  int16
4   SCI        ImageHDU      209    (4096, 2051)  float32
5   ERR        ImageHDU      78     (4096, 2051)  float32
6   DQ         ImageHDU      70     (4096, 2051)  int16
7   WCSCORR    BinTableHDU   59     14R x 24C     [40A, I, 1A, 24A, 24A, 24A, 24A, D, D, D, D, D, D, D, D, 24A,
24A, D, D, D, D, J, 40A, 128A]
8   HDRLET     HeaderletHDU  17     ()
9   HDRLET     HeaderletHDU  18     ()
```

Extension 7 is a FITS table that keeps track of the headerlets and WCS changes that have been applied to the image. It can be examined using the PyRAF command **tprint** or any other program that can interpret FITS tables. Here, only the image science group, and RMS for RA and Dec are displayed

--> tprint ibof01aqq_flt.fits[7] columns=WCS_ID,EXTVER,RMS_RA,RMS_Dec

```
#  Table ibof01aqq_flt.fits[7]  Thu 14:57:14 06-Dec-2012
# row WCS_ID          EXTVER  RMS_RA                  RMS_Dec
#
    1 OPUS                 1  0.                      0.
    2 OPUS                 2  0.                      0.
    3 IDC_w2r1956ri        1  0.                      0.
    4 IDC_w2r1956ri        2  0.                      0.
    5 TWEAK                1  1.496291389460000E-6    7.179168974860000E-7
    6 TWEAK                2  1.496291389460000E-6    7.179168974860000E-7
    7                      0  0.                      0.
    8                      0  0.                      0.
    9                      0  0.                      0.
   10                      0  0.                      0.
   11                      0  0.                      0.
   12                      0  0.                      0.
   13                      0  0.                      0.
   14                      0  0.                      0.
```

In *ibof01aqq_flt.fits[7]*, shown above,

- Rows 1 & 2: "OPUS" is the initial WCS ID from the pipeline.
- Rows 3 & 4: "IDC_w2r1956ri" is the WCS ID created by the task UpdateWCS when AstroDrizzle was first run in the pipeline.
- Rows 5 & 6: The WCS ID, TWEAK, was the default name used in this exercise. (The WCS_ID name can be customized in the TweakReg parameter ***wcsname***.)
- The RMS values shown in the columns, RMS_RA and RMS_Dec, are in units of degrees.
- Empty rows are space holders for future WCS updates to the FITS header

The Python command **pyfits.getheader** can be used to look at the headerlet information

--> pyfits.getheader('ibof01aqq_flt.fits', ext=9)

```
XTENSION  = 'HDRLET  '        / FITS extension
BITPIX    =          8        / array data type
NAXIS     =          1        / number of array dimensions
NAXIS1    =      14400        / Axis length
PCOUNT    =          0        / number of parameters
GCOUNT    =          1        / number of groups
XIND1     =       2880        / byte offset of extension 1
XIND2     =       8640        / byte offset of extension 2
COMPRESS  =          F        / Uses gzip compression
HDRNAME   = 'TWEAK   '        / Headerlet name
DATE      = '2013-01-25T17:04:33' / Date FITS file was generated
SIPNAME   = 'ibof01aqq_w2r1956ri' / SIP distortion model name
WCSNAME   = 'TWEAK   '        / WCS name
DISTNAME  = 'ibof01aqq_w2r1956ri-NOMODEL-NOMODEL' / Distortion model name
NPOLFILE  = 'NOMODEL '        / origin of non-polynmial distortion model
D2IMFILE  = 'NOMODEL '        / origin of detector to image correction
EXTNAME   = 'HDRLET  '        / Extension name
EXTVER    =          2
```

8. Run AstroDrizzle on the *FLT* Images With Updated WCS Information

Open the AstroDrizzle GUI in PyRAF using the **epar** command.

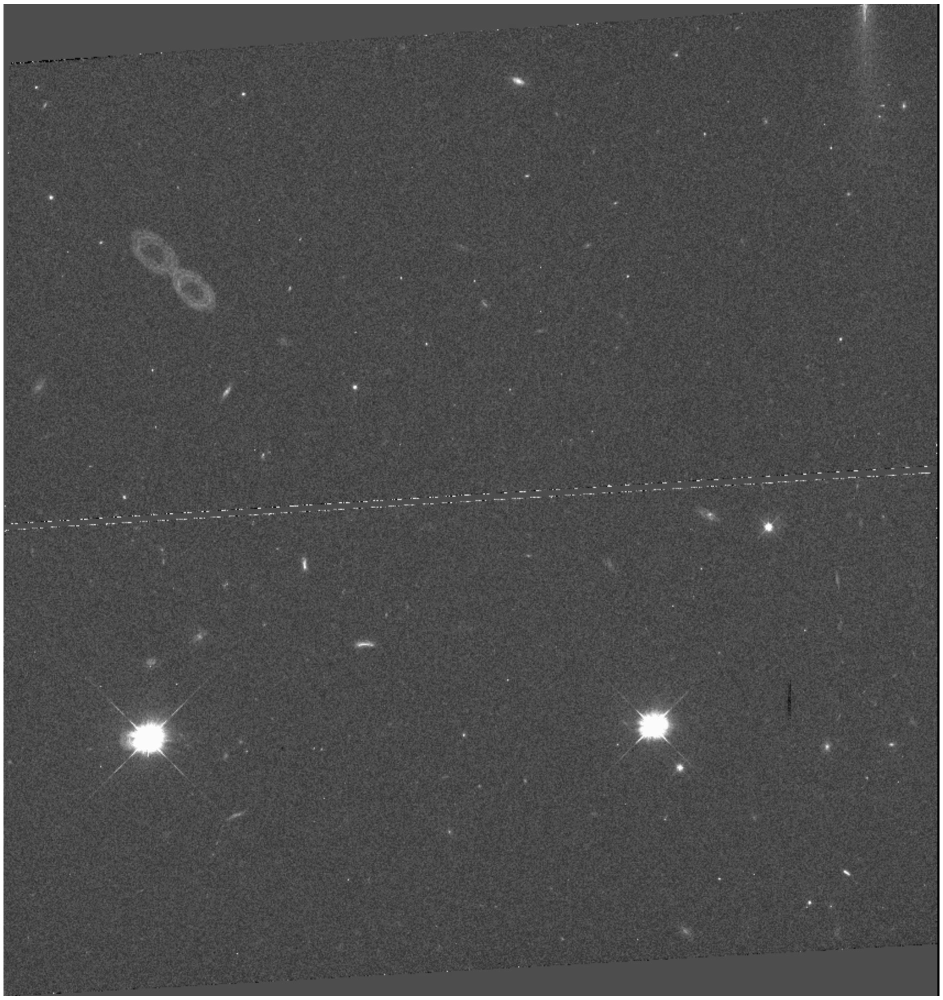
--> epar astrodrizzle

First, click on the ***Defaults*** button on the upper right of the GUI to reset the parameters.

- By default, under ***STATE OF INPUT FILES***, ***clean*** should be set to ***No***, to remove intermediate files.
- Under ***CUSTOM WCS FOR SEPARATE OUTPUTS***, set ***driz_sep_wcs*** to ***Yes***, and set ***driz_sep_scale*** to ***0.03333.*** The first parameter tells AstroDrizzle that non-default parameters are being used in that section, and the second parameter sets the output image scale for each drizzled *FLT* image to 0.03333 arcseconds/pixels. This scale was chosen because it samples the PSF fairly well for UVIS data, and is convenient because three pixels equal 0.1 arcseconds. The exact value of the output scale, however, is not critical, but using an output pixel size that well-samples the PSF is recommended for cosmic ray rejection.
- By default, under ***REMOVE COSMIC RAYS WITH DERIV, DRIZ_CR***, ***driz_cr_corr*** should be set to ***no***. This tells AstroDrizzle that cosmic ray-cleaned *FLT* files (*crclean.fits*) images are not needed.
- By default, under ***DRIZZLE FINAL COMBINED IMAGE***, ***driz_combine*** should be set to ***Yes*** to create a final combined image.
- ***driz_sep_bits*** and ***final_bits*** are set to "***64,32***" so that AstroDrizzle treats input pixels flagged in the *FLT* data quality extensions with the values 64 (CTE tails) and 32 (warm pixels) as "good" pixels.
- Under ***CUSTOM WCS FOR FINAL OUTPUT***, set ***final_wcs*** to ***Yes***, and set ***final_scale*** to ***0.03333*** (for the same reasons explained above for ***driz_sep_scale***).
- Click the ***Execute*** button to run AstroDrizzle.

By default, AstroDrizzle will create three separate final products, *final_drz_sci.fits* , *final_drz_wht.fits*, and *final_drz_ctx.fits*, the science image, weight image, and context image, respectively. A unique drizzle-combined image name can be given in the AstroDrizzle parameter ***output***. A single FITS file with three extensions can be created by setting the parameter ***build*** to ***Yes***.

Figure 5: the final product *final_drz_sci.fits*



Conclusion

This example demonstrates a technique for obtaining good alignments using AstroDrizzle and TweakReg, even when there are only a few good sources and the images have cosmic rays and heavily saturated stars. In this case, the images were already well aligned. But the same process can be applied to images taken in multiple visits as long as there are a reasonable number of good stars in overlapping regions.

For the original science goals described at the start of this example, this entire process needs to be repeated for each visit (epoch) to create images aligned within each visit.

Images for each visit (epoch) would then be combined to create a drizzled product that could then be used as input to TweakReg to align all the visits to each other. The updates to the drizzled image headers written by TweakReg would then be used as input to the TweakBack task to update each of the *FLT* images with the final alignment solution for all visits (see Example 7 in the DrizzlePac Handbook). Note that the process described in this example completely eliminates the need for the deprecated, unsupported shift files (used in MultiDrizzle).