7.5 WFC3/UVIS: Aligning Images by Defining an Output Reference Frame

Introduction

The following example describes the alignment of images in two filters obtained with the WFC3/UVIS channel. While **tweakreg** aligns the image World Coordinate System (WCS) header information, this example demonstrates how to define the output frame reference image for aligning the two filters in pixel space. This example can be generalized for aligning images obtained in different cameras, for example WFC3/UVIS to ACS/WFC.

Step-by-step instructions are provided below, in a cookbook fashion, with minimal discussion of parameters. Commands are given using Python syntax executed in PyRAF. Since the tasks have been restored to default values prior to execution, mostly non-default parameters are specified in the command-line examples.

Summary of Steps

1. Description of the data.
2. Use **tweakreg** to align images in a given filter.
3. For each filter, combine the images using **astrodrizzle** to produce high signal-to-noise drizzled products, free from cosmic rays and detector artifacts.
4. Improve the alignment between the two *drizzled* images, each representing a filter, using **tweakreg**.
5. Using **tweakback**, propagate the new "tweaked" solutions obtained from aligning the two drizzled images back to their respective original `flt.fits` images.
6. Drizzle-combine the updated `flt.fits` images for each filter, using the improved WCS offsets, to create WCS-aligned drizzled images for each filter.
7. Demonstrate how defining a reference frame in **astrodrizzle** can create drizzled images for each filter that are aligned, in pixel space, to a common "output grid."

7.5.1 Description of the Data

In August 2009, the first epoch of M83 observations were obtained as part of the WFC3 Early Release Science program (Proposal ID 11360). Three images with small dithers (`POS TARG`s) were obtained, in each filter, to fill the gap between the two UVIS chips and to allow for rejection of cosmic rays and detector artifacts. This

example will focus on aligning two sets of images, one in the F814W filter and the other in F438W, each obtained in different visits.

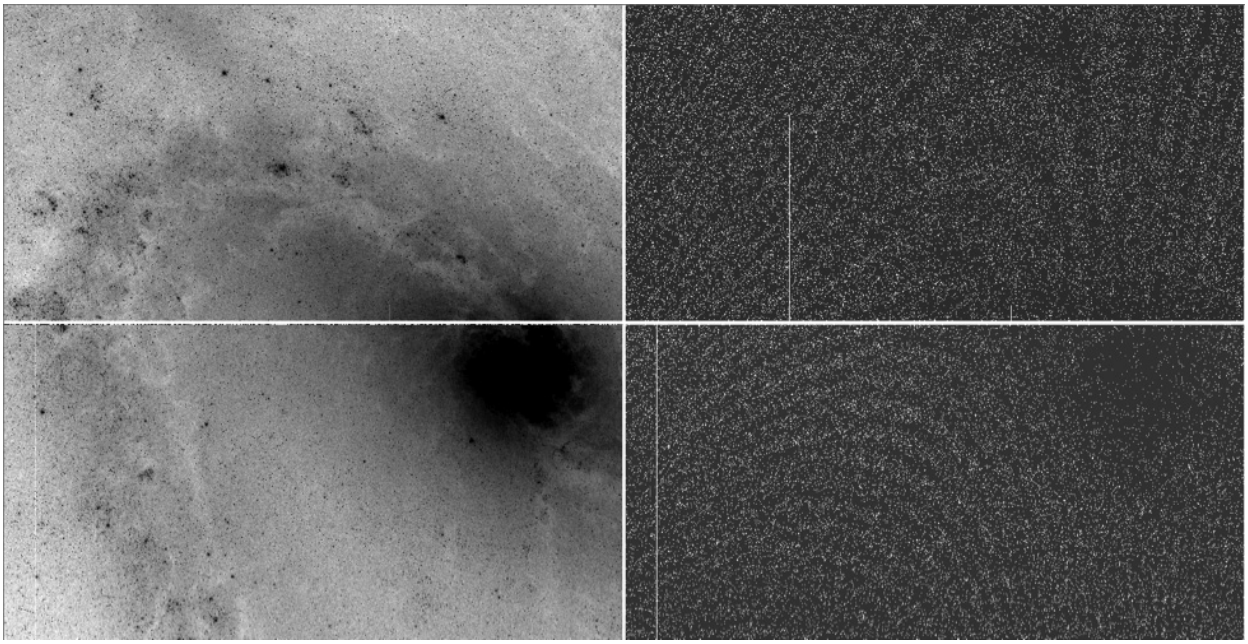
The datasets used in this example are listed in Table 7.7. Note that the images have been given different names for clarity. While the `flt.fits` suffix has been dropped in the new name, users should keep in mind that these are calibrated `flt.fits` products from the pipeline.

Table 7.7: Summary of Images for this Example

Unique & Orig. Image Name	Association ID	Proposal ID	Visit & Line Number	POS TARG (x,y in arcsec.)	PA_V3 Orientation (degrees)	Observation Date	Exposure Time (sec.)
f814w_pos1_01.fits ib6w62tvq_flt.fits	IB6W62060	11360	62.022	0.0000,0.0000	305.0004	2009-08-26	401.00
f814w_pos1_02.fits ib6w62txq_flt.fits	IB6W62060	11360	62.023	1.4460,2.9260	305.0001	2009-08-26	401.00
f814w_pos1_03.fits ib6w62tzq_flt.fits	IB6W62060	11360	62.024	-1.4460,-2.9260	305.0007	2009-08-26	401.00
f438w_pos1_01.fits ib6w61uaq_flt.fits	IB6W61050	11360	61.001	0.0000,0.0000	305.0004	2009-08-26	600.00
f438w_pos1_02.fits ib6w61umq_flt.fit	IB6W61050	11360	61.008	1.4460,2.9260	305.0001	2009-08-26	640.00
f438w_pos1_03.fits ib6w61uzq_flt.fits	IB6W61050	11360	61.015	-1.4460,-2.9260	305.0007	2009-08-26	640.00

Science and data quality extensions for one of the calibrated pipeline products, the first F814W image, are shown in Figure 7.29. Note that the pipeline was overly aggressive at flagging cosmic rays, as reflected in the large number of white pixels in the data quality (DQ) array; this is likely due to a poor estimate of the sky background. These cosmic ray flags, with the value of 4096 in the `flt.fits` DQ array, will be, by default, automatically reset to be treated as "good" pixels by **astrodrizzle** (using ***resetbits=4096***). Reprocessing the dataset with different parameter values will then allow the user to manually optimize cosmic ray rejection.

Figure 7.29: Science and Data Quality Extensions for the First F814W Calibrated (`flt.fits`) Image



The science extensions, [sci,2] and [sci,1], on the left, with their corresponding data quality extensions [dq,2] and [dq,1], on the right. Note that the pipeline did a poor job in flagging cosmic rays, resulting in a pattern which reflects the noise in the sky background. Manual reprocessing with astrodrizzle will allow these flags to be reset and optimized by the user.

7.5.2 Align Images for Each Filter Set Using tweakreg

Before running **tweakreg**, the **drizzlepac** package has to be loaded and the required tasks for this example have to be imported and reset to default values.

```
--> import drizzlepac
--> from drizzlepac import tweakreg
--> from drizzlepac import astrodrizzle
--> from drizzlepac import tweakback

--> unlearn tweakreg force=yes
--> unlearn imagefindpars force=yes
--> unlearn astrodrizzle force=yes
--> unlearn tweakback force=yes
```

This target has few actual point sources, so the alignment is primarily based on the positions of H II regions. The commands below reflect parameters that have been optimized after several interactive trials.

The parameter value for ***conv_width*** is recommended to be approximately twice the FWHM of the PSF. By default, **imagefindpars** will automatically compute the sigma value (standard deviation) of the sky background, so it is necessary to adjust the ***threshold*** parameter in **imagefindpars** to ensure an adequate number of sources are found to compute the fit. (For images containing a large number of saturated objects, the **imagefindpars** parameter ***peakmax*** can be set to exclude those sources from the fit.)

Using the **tweakreg** parameters shown below (that also includes parameters from the **imagefindpar** task that is used by **tweakreg** for object detection settings), several thousand sources are initially detected with several hundred matches, providing solutions accurate to much better than 0.1 pixels r.ms. Users are strongly encouraged to inspect the four-panel residual plot for signs of remaining systematics in the final residuals, like, for example, a slope which indicates that a residual rotation or scale still remains in the data. Instructions for examining the source catalogs are given in Section 7.5.3.

```
--> tweakreg.TweakReg('f814w_pos1_??.fits,conv_width=3.5,\
threshold=200,shiftfile=True,outshifts='shift_814_pos1.txt',updatehdr=False)

-->tweakreg.TweakReg('f438w_pos1_??.fits,conv_width=3.5,\
threshold=200,shiftfile=True,outshifts='shift_438_pos1.txt',updatehdr=False)
```

In these **tweakreg** line-commands in PyRaF (that can also be run in the Python interface), the following settings were used:

- ***conv_width=3.5***, the convolution kernel which is twice the PSF FWHM of sources. It identifies potential sources which look closest to a gaussian with the FWHM equal to this parameter's value.
- ***threshold=200*** sigma above the local background for object detection.
- ***updatehdr=False*** so that the offsets are not used to modify the WCS information in each image.
- ***shiftfile=True*** and ***outshifts='shift_438_pos1.txt'*** so that a shift file containing the computed offsets are written to a named file. The contents of the two output shift files are shown below. Shift files are only created for record-keeping purposes. Unlike **multidrizzle, astrodrizzle** does not accept a shift file as input because offsets are directly applied to the WCS in the image headers.

F814W and F435W shift files:

```
Image               dX         dY         drot        scale     xfit_rms  yfit_rms
f814w_pos1_01.fits  0.000000   0.000000   0.000000    1.000000  0.000000  0.000000
f814w_pos1_02.fits  -0.055635  -0.079298  0.000071    1.000000  0.045810  0.052170
f814w_pos1_03.fits  -0.045426  -0.022232  359.999817  0.999998  0.044504  0.054453

Image               dX         dY         drot        scale     xfit_rms  yfit_rms
f438w_pos1_01.fits  0.000000   0.000000   0.000000    1.000000  0.000000  0.000000
f438w_pos1_02.fits  -0.098221  0.042909   0.000385    1.000000  0.069631  0.066566
f438w_pos1_03.fits  0.046986   0.160370   359.999612  1.000001  0.067033  0.061197
```

When **tweakreg** is run with its default setting ***writecat=True***, it creates source catalogs with extension "`.coo`" for each science image extension.

tweakreg also automatically generates an ASCII table listing the names of the catalogs generated for each image. The name of the table can be specified by the parameter ***catfile***, but if no name is provided, the default name is <*reference_file*>`_xy_catfile.list`. For this set of data, the default name of the ***catfile*** is `f814w_pos1_01_xy_catfile.list`, and its contents are shown below:

```
f814w_pos1_01.fits f814w_pos1_01_sci1_xy_catalog.coo f814w_pos1_01_sci2_xy_catalog.coo
f814w_pos1_02.fits f814w_pos1_02_sci1_xy_catalog.coo f814w_pos1_02_sci2_xy_catalog.coo
f814w_pos1_03.fits f814w_pos1_03_sci1_xy_catalog.coo f814w_pos1_03_sci2_xy_catalog.coo
```

When the user is satisfied with the results, **tweakreg** should be run one last time with ***updatehdr=True***, to update the WCS of each `flt.fits` image with the final solution. At this point, it is no longer necessary to run the task in interactive mode. To update the WCS information in the image headers as a batch, plotting parameters can be turned off (as shown below) and the task will run without prompting the user to verify the solutions.

```
--> tweakreg.TweakReg('f814w_pos1_??.fits,writecat=False,\
catfile='f814w_pos1_01_xy_catfile.list',residplot='NoPlot',\
see2dplot=no,updatehdr=True,wcsname='TWEAK_814')

--> tweakreg.TweakReg('f438w_pos1_??.fits,writecat=False,\
catfile='f438w_pos1_01_xy_catfile.list',residplot='NoPlot',\
see2dplot=no,updatehdr=True,wcsname='TWEAK_438')
```

This final **tweakreg** run uses the same settings as the previous run, except that the WCS information in the image headers are updated (***updatehdr=True***). In addition, ***writecat*** is set to ***False*** and the ***catfile*** parameter points to a previously-made listing of catalogs so that tweakreg makes use of existing catalogs to save processing time.

When ***updatehdr*** is set to ***True***, the updated WCS solutions can be tracked via a unique name, given by the **tweakreg** parameter ***wcsname***. (The default value is the string `TWEAK`). It is useful to choose a meaningful name to keep track of it when querying the `flt.fits` headers.

The example below shows the query results for an image header with several different WCSs, each identified by a unique name that reflects the history of the dataset.

- `WCSNAMEO=OPUS`: the original WCS name from OPUS pipeline processing
- `WCSNAMEA=IDC_v5r1512gi:` the WCS corresponding to the distortion reference file (IDCTAB) from running the task **updatewcs**
- `WCSNAMEB=TWEAK_438` is a user-generated solution from this example
- `WCSNAME =TWEAK_438` is also the most current WCS

To query the `WCSNAME` keywords in the image header, the following IRAF command may be used:

```
--> hedit f???w_pos?_??.fits[1] wcsnam* .
f814w_pos1_01.fits[1],WCSNAMEO = OPUS
f814w_pos1_01.fits[1],WCSNAMEA = IDC_v5r1512gi
f814w_pos1_01.fits[1],WCSNAMEB = TWEAK_814
f814w_pos1_01.fits[1],WCSNAME = TWEAK_814
f438w_pos1_01.fits[1],WCSNAMEO = OPUS
f438w_pos1_01.fits[1],WCSNAMEA = IDC_v5r1512gi
f438w_pos1_01.fits[1],WCSNAMEB = TWEAK_438
f438w_pos1_01.fits[1],WCSNAME = TWEAK_438
```

7.5.3 Overplot Matched Sources Onto the Original flt.fits Image

While this step is optional, it is highly recommended.

The commands below use PyRAF tasks to display a chip in a `flt.fits` image, overplots the initial source catalog "`image_sci?_xy_catalog.coo`"[8] (in red) on the image, then overplots the matched catalog "`image_catalog_fit.match`" (in green). The matched catalog shows which sources were used to compute the **tweakreg** solution, allowing the user to verify that actual sources (not cosmic rays or hot pixels) were used to determine this solution (See Figure 7.30).

Matched catalog files (with suffix "`catalog_fit.match`") are created for all images except the first in the list, which serves, by default, as the reference. Columns 11 and 12 give the original position in the `flt.fits` images, and column 15 (labeled

8. "?" is a single-character wildcard for the extension number, either 1 or 2 for UVIS. The names in quotes are files suffixes for the catalogs, as illustrated in the command examples. In this example, chip 1 of the first F814W image is displayed, which corresponds to science extension 2.

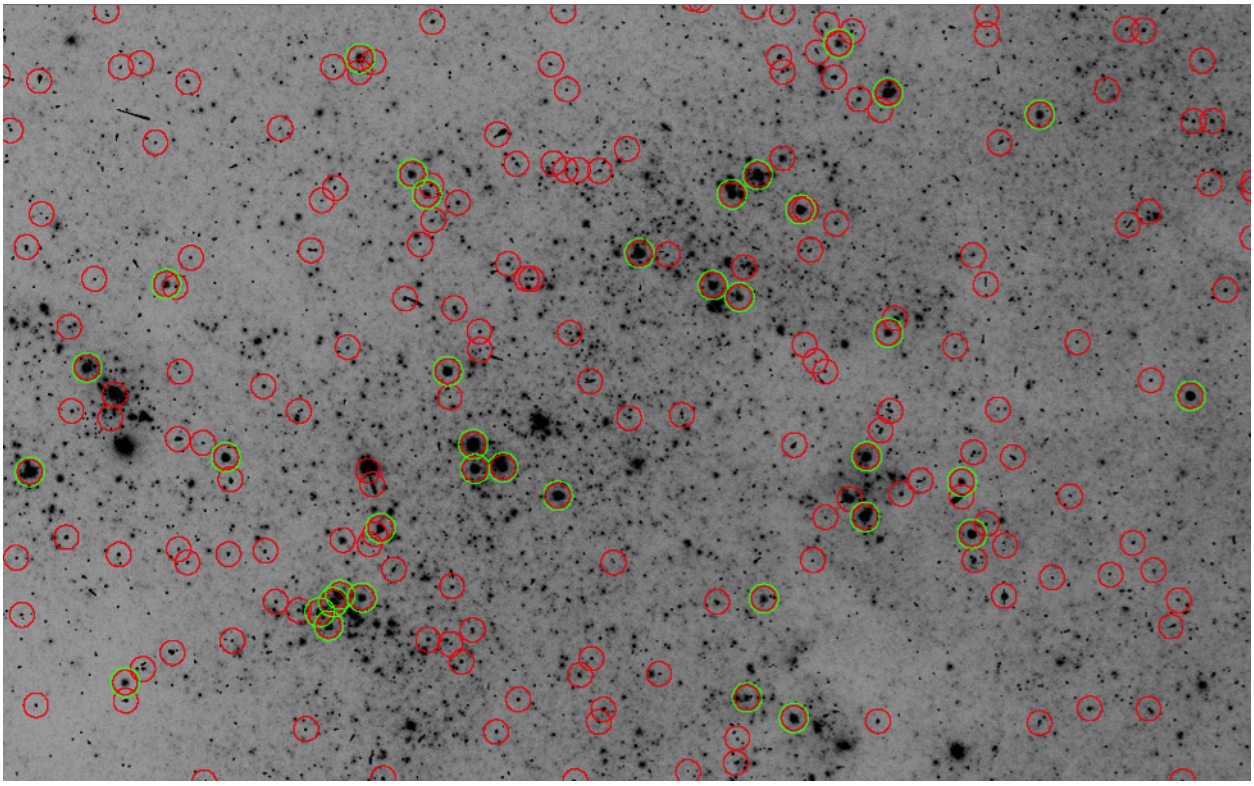
"EXTVER ID" in *catalog_fit.match files) specifies the chip that corresponds to the coordinates.

```
-> tselect f814w_pos1_02_catalog_fit.match \
f814w_pos1_02_catalog_fit_sci1.match "c15 .eq. 1"
--> tselect f814w_pos1_02_catalog_fit.match \
f814w_pos1_02_catalog_fit_sci2.match "c15 .eq. 2"

--> tproject f814w_pos1_02_catalog_fit_sci1.match \
f814w_pos1_02_catalog_fit_sci1.xyflt "c11,c12"
--> tproject f814w_pos1_02_catalog_fit_sci2.match \
f814w_pos1_02_catalog_fit_sci2.xyflt "c11,c12"

--> display f814w_pos1_02.fits[sci,2] 1
--> tvmark 1 f814w_pos1_02_sci2_xy_catalog.coo mark=circle radii=5 color=204
--> tvmark 1 f814w_pos1_02_catalog_fit_sci2.xyflt mark=circle \
radii=7 color=205
```

Figure 7.30: Source Catalogs Over-Plotted on a flt.fits Image



This is a useful way to inspect objects used for computing the offset fit. Matched sources are shown in green.

Create a Drizzle-combined Image for Each Filter

Note this common "gotcha:"
for WFC3/UVIS and ACS/WFC images,
EXTVER ID = 2 = [sci,2] = chip1 = flt.fits[4]
EXTVER ID = 1 = [sci,1] = chip2 = flt.fits[1]

For each filter, images are drizzle-combined to create a high signal-to-noise image, free from cosmic rays or detector artifacts. These two drizzled image will, themselves, be aligned in a later step.

```
--> astrodrizzle.AstroDrizzle('f814w_pos1_??.fits',\
output='f814w_pos1',driz_sep_bits=64,32',driz_cr_corr=yes,\
final_bits='64,32',final_wcs=yes,final_scale=0.0396,final_rot=0.)

--> astrodrizzle.AstroDrizzle('f438w_pos1_??.fits,\
output='f438w_pos1',driz_sep_bits='64,32',driz_cr_corr=yes,\
final_bits='64,32',final_wcs=yes,final_scale=0.0396,final_rot=0.)
```

In the **astrodrizzle** commands shown above,

- ***driz_sep_bits*** and ***final_bits*** set to "***64,32***" tells **astrodrizzle** to treat input pixels flagged in the `flt.fits` data quality extensions with the value 64 (CTE tails) and 32 (warm pixels) as "good" pixels.
- ***driz_cr_corr =yes*** is a switch to tell the software to create a cosmic ray mask for each input image.
- f***inal_wcs= yes*** because there are two "custom" output image parameter settings in that category:
  - ***final_scale***, the final output scale, is set to ***0.0396*** arcseconds/pixel
  - ***final_rot=0***[9] orients the output image such that its *y*-axis is relative to the north (at the top of the image).

Note:

Information about the primary WCS solution for each input image that was used to create the drizzled image is stored in a new drizzled file extension of type `HDRTAB`. The `WCSNAME` in the drizzled image header itself is set to the value `DRZWCZ`.

```
--> hedit *drz_sci.fits wcsnam* .
f814w_pos1_drz_sci.fits,WCSNAME = DRZWCS
f438w_pos1_drz_sci.fits,WCSNAME = DRZWCS
```

9. Setting ***final_rot*** to the default value, ***none*** or ***INDEF***, will orient the output image to the same orientation as the reference image.

Next, the data products are inspected to look for any offsets. In **ds9**, the images may be matched by the WCS or in physical (or pixel) units, and then blinked. In this case, the positions of stars in each science array appear to be offset by approximately (+1,+6) pixels.

```
--> displ f438w_pos1_drz_sci.fits 1 zs-zr-z1=-0.2 z2=0.5 \
ztran=log fill+
--> displ f814w_pos1_drz_sci.fits 2 zs-zr-z1=-0.2 z2=0.5 \
ztran=log fill+
```

The weight images should also be inspected. In this case, the final weight image parameter was set to ***EXP***, so the `WHT` images for each filter resemble an exposure time map of the combined dataset. For UVIS data, it is important to verify that an imprint of sources is not apparent in the weight image, which could imply that one or more frames were being excluded from the final product.

```
--> displ f438w_pos1_drz_wht.fits 3 fill+
--> displ f814w_pos1_drz_wht.fits 4 fill+
```

Figure 7.31: Science Extension of the F814W Drizzled Product Oriented North-Up

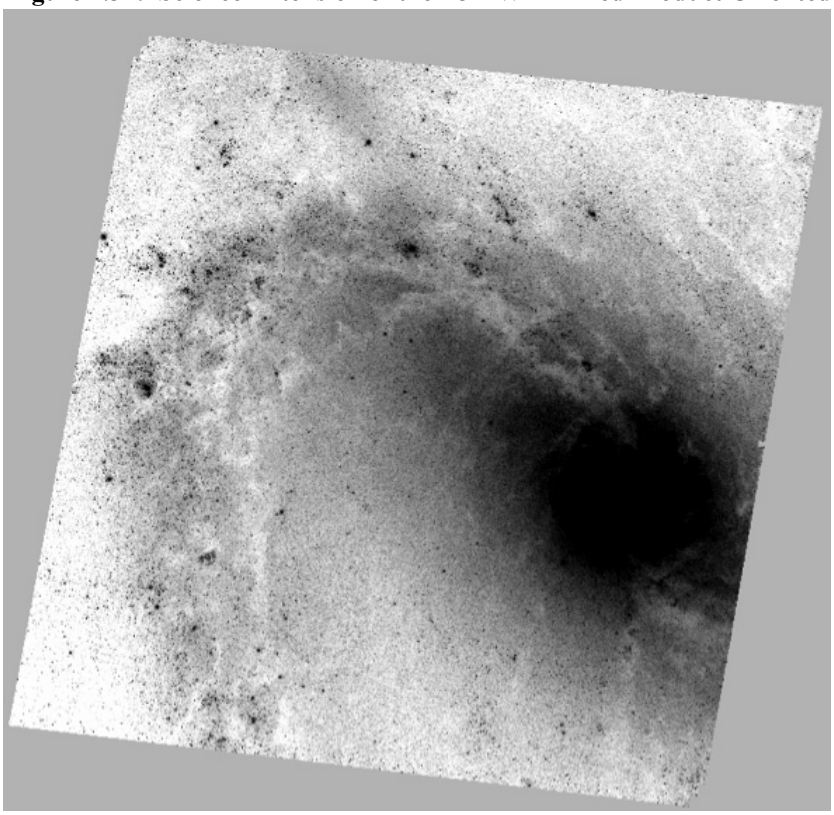
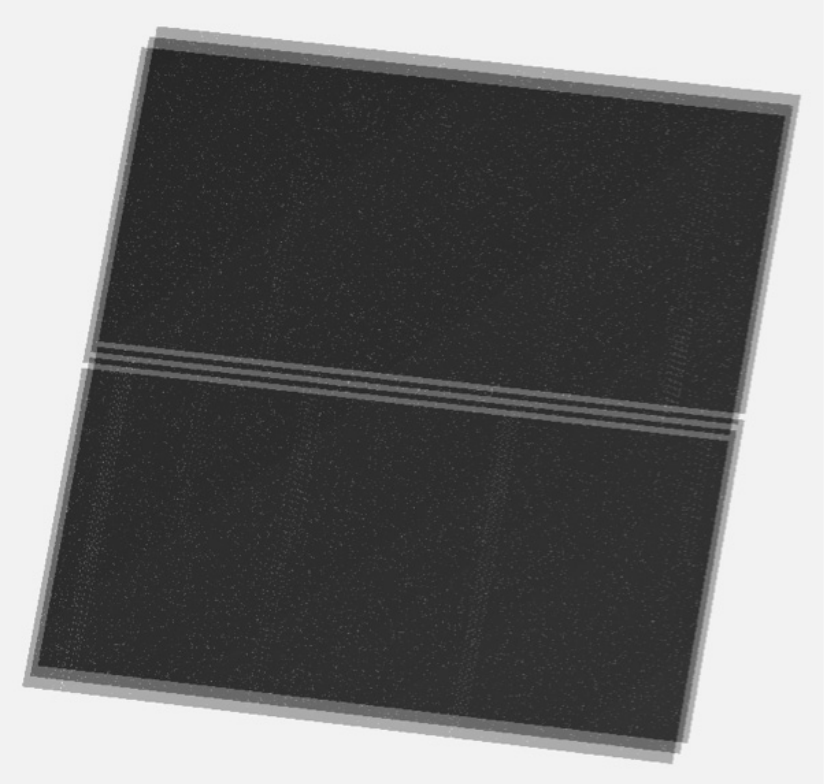


Figure 7.32: Weight Extension of the F814W Drizzled Product



The gap dither pattern can be seen, as well as pixels with DQ flags and cosmic rays which were treated as "bad" in the final drizzle step.

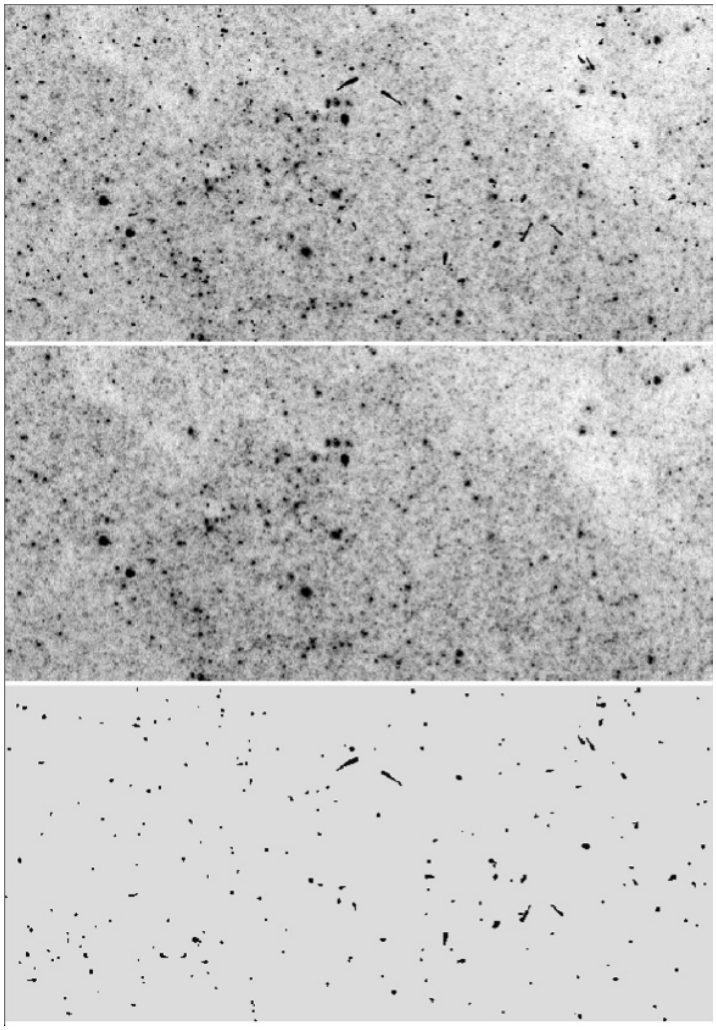
The quality of the cosmic ray masks should be verified by blinking the original `flt.fits` image with both the cosmic ray-cleaned image (suffix `cr_clean.fits`) and the cosmic ray mask (suffix `cr_mask.fits`). When testing various parameter values, the user may find that the cores of stars have been improperly masked, and this indicates that either the ***driz_cr_scale*** parameter should be increased or that the alignment computed by **tweakreg** was not optimal. If not enough cosmic rays were detected, the ***driz_cr_snr*** parameter should be lowered to detect fainter objects. The example below is for a single chip, but this should be repeated for both chips.

```
--> displ f814w_pos1_01.fits[sci,2] 1 zs-zr-z1=-0.2 z2=0.5 ztran=log fill+

--> displ f814w_pos1_01_crclean.fits[sci,2] 2 zs-zr-z1=-0.2 z2=0.5 ztran=log fill+

--> displ f814w_pos1_01_sci2_crmask.fits 3 zs+ zr+ fill+
```

Figure 7.33: Close-up of the Same Region in Three Types of Input Products



The top image is a "zoomed-in" region of a flt.fits image, a cosmic ray-cleaned version of that image (middle), and the corresponding CR-mask image (bottom). Blinking these products is recommended to ensure that the cores of stars are not flagged as cosmic rays.

7.5.4 Align the Header WCS of the Two Filter Images

tweakreg is run on each drizzle-combined filter image to compute the offsets that were previously seen in the **ds9** display:

```
--> tweakreg.TweakReg('f438w_pos1_drz_sci.fits',\
refimage='f814w_pos1_drz_sci.fits',conv_width=3.5,threshold=5,\
shiftfile=True,outshifts='shift_438ref_pos1.txt',nclip=10,\
updatehdr=False)
```

Note that the ***threshold*** parameter was selected to produce a matched catalog with a few thousand (3,900) sources. Before matching, the lists contain 6,900 sources in F438W and 51,100 sources in F814W. When the input image and the reference image have very different signal-to-noise values, it may be useful to create a reference catalog a priori and provide this to ***tweakreg*** via the ***refcat*** parameter. This will help

ensure that the two filters (or detectors) have similar numbers of objects for matching. The example below shows the syntax for providing a prepared reference catalog:

```
-> tweakreg.TweakReg('f438w_pos1_drz_sci.fits',\
refimage='f814w_pos1_drz_sci.fits',\
refcat='f814w_pos1_drz_sci_sky_catalog.ref',conv_width=3.5,\
threshold=5,shiftfile=True,outshifts='shift_438ref_pos1.txt',\
nclip=10,updatehdr=False)
```

The contents of the resulting shift file are given below. Notice that the fit produces an offset similar to the one predicted from the earlier visual inspection of the drizzled products in **ds9**.

```
Image              dX       dY       drot       scale     xfit_rms  yfit_rms
f438w_pos1_drz_sci  0.973094  5.995664  359.999049  0.999970  0.105581
0.110062
```

To update the image headers with the new solution, run **tweakreg** with ***updatehdr=True***.

```
--> tweakreg.TweakReg('f438w_pos1_drz_sci.fits',\
refimage='f814w_pos1_drz_sci.fits',conv_width=3.5,threshold=5,\
nclip=10,updatehdr=True,wcsname='TWEAK_814DRZ')
```

IRAF's **hedit** command shows that the keyword `WCSNAME` has now been updated to reflect the new solution:

```
-> hedit f438w*drz_sci.fits wcsnam* .
f438w_pos1_drz_sci.fits,WCSNAMEA = DRZWCS
f438w_pos1_drz_sci.fits,WCSNAMEB = TWEAK_814DRZ
f438w_pos1_drz_sci.fits,WCSNAME = TWEAK_814DRZ
```

The WCS of the F814W and F438W drizzled images are now aligned. No further processing is necessary, as long as the user defines photometric source catalogs in R.A. and Dec. coordinates.

If, however, the user requires the physical pixels of the two frames to align (for example, to subtract a broad-band image from a narrowband image in order to remove the continuum), the images must be drizzled one more time with a pre-defined output frame of reference.

7.5.5 Propagate Improved Solution to Original flt.fits Images with tweakback

After **tweakreg** has been used to align drizzled products[10], the **tweakback** task can be used to propagate the updated WCS back to the original `flt.fits` images. AstroDrizzle may then be used to re-drizzle the `flt.fits` images frames; the

drizzle products will be aligned, and can be verified by displaying and blinking them in **ds9**. Note that **tweakreg** only aligns the WCS in the image headers. To ensure that the actual pixels are aligned in the final drizzled product, a output frame of reference must be defined, as described in the next section.

```
--> tweakback.tweakback('f438w_pos1_drz_sci.fits',\
input='f438w_pos1_??.fits',verbose=True)
 Processing f438w_pos1_01.fits[('SCI', 1)]
Updating header for f438w_pos1_01.fits[1]
WCS Keywords
CD_11 CD_12: -1.09949173435E-05 -1.97018332034E-06
CD_21 CD_22: -1.27983613307E-06 1.08908389688E-05
CRVAL : 204.271406112 -29.8720133266
CRPIX : 2048.0 1026.0
NAXIS : 4096 2051
Plate Scale : 0.039851380502520579
ORIENTAT : -10.25406939504851

--> hedit f438w_pos?_??.fits[1] wcsnam* .
f438w_pos1_01.fits[1],WCSNAMEO = OPUS
f438w_pos1_01.fits[1],WCSNAMEA = IDC_v5r1512gi
f438w_pos1_01.fits[1],WCSNAMEB = TWEAK_TILE1
f438w_pos1_01.fits[1],WCSNAMEC = TWEAK_814DRZ
f438w_pos1_01.fits[1],WCSNAME = TWEAK_814DRZ
```

As a sanity check, one can rerun **tweakreg** on the original list to ensure that the intra-visit alignment is still optimal. This step is not necessary, but it is shown for completeness, along with the output from the shift file.

```
--> tweakreg.TweakReg('f438w_pos1_??.fits,updatehdr=False,\
conv_width=3.5,threshold=200,shiftfile=True,\
outshifts='shift_438_pos1_iter2.txt')
```

Image	dX	dY	drot	scale	xfit_rms	yfit_rms
f438w_pos1_01.fits	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
f438w_pos1_02.fits	0.000023	-0.000038	0.000024	1.000000	0.067504	0.066542
f438w_pos1_03.fits	-0.000098	0.000042	0.000015	1.000000	0.067033	0.061176

10. In this example, **tweakreg** was used to align two drizzled images, each with a different filter. The same concept can be applied to aligning drizzled images of different detectors and mosaic pointings.

7.5.6 How to Drizzle the Images to a Common Pixel Frame

This section demonstrates how the drizzled images for the two filters can be aligned in pixel space. Some reasons for doing this include creating color images, subtracting one image from another, or combining them to create an object detection "white light" image (i.e., adding images taken with different filters).

This maybe done in one of two ways:

Option A:

Set the parameter ***final_refimage*** to point to the F814W drizzled product. This will automatically define the size, orientation, scale, and R.A./Dec. of the central pixel. To retain the drizzled product from Section 7.5.5 for comparison, it can be renamed as a "version 1" image prior to reprocessing.

Parameters specifying the final output grid are shown in bold in the examples below.

```
--> imrename f438w_pos1_drz_sci.fits f438w_pos1_drz_sci_v1.fits
--> imrename f438w_pos1_drz_wht.fits f438w_pos1_drz_wht_v1.fits

--> astrodrizzle.AstroDrizzle('f438w_pos1_??.fits,\
output='f438w_pos1',driz_sep_bits='64,32',driz_cr_corr=yes,\
final_bits='64,32',final_wcs=yes,\
final_refimage='f814w_pos1_drz_sci.fits')
```

If ***final_refimage*** is not defined, the F438W drizzled product will not be aligned with the F814W image in *pixel space* because **tweakreg** operates on the images' header WCS, not the physical pixels.

AstroDrizzle automatically computes the image reference frame for each set of input images based on the orientation of the reference image (`f814w_pos1_drz_sci.fits` in the example above) and the distortion solution of each filter. The product dimensions will be the smallest-sized output image which incorporates the full set of input frames.

Option B:

Instead of defining a ***final_refimage***, set up a pre-defined output grid based on keywords in the original F814W drizzled product. The **hselect** command, shown below, queries the value of the reference position in pixels and in sky coordinates (R.A./Dec.), and the **imheader** command returns the size of the array. The user may wish to allow for additional "padding" in the final drizzled product to account for offsets in pointing or differences in geometric distortion with respect to the reference image. This may be done by specifying the ***final_outnx*** and ***final_outny*** parameters in AstroDrizzle. Note that the output products have been given unique rootnames, `f814w_pos1r` and `f438w_pos1r`, to reflect the fact that they were drizzled to a pre-defined reference grid.

```
-> hselect f814w_pos1_drz_sci.fits \
$I,crpix1,crpix2,crval1,crval2 yes
# output:
f814w_pos1_drz_sci.fits 2462.24 2597.76 204.268999 -29.860656

--> imheader *drz_sci.fits
# output:
f814w_pos1_drz_sci.fits[4924,5195][real]:
f814w_pos2_drz_sci.fits[4581,4885][real]:

--> astrodrizzle.AstroDrizzle('f814w_pos1_??.fits,\
output='f814w_pos1r',driz_sep_bits='64,32',driz_cr_corr=yes,\
final_bits='64,32',final_wcs=yes,final_scale=0.0396,final_rot=0,\
final_outnx=5000,final_outny=5200,final_ra=204.268999,\
final_dec=-29.860656)

--> astrodrizzle.AstroDrizzle('f438w_pos1_??.fits,\
output='f438w_pos1r',driz_sep_bits='64,32',driz_cr_corr=yes,\
final_bits='64,32',final_wcs=yes,final_scale=0.0396,final_rot=0,\
final_outnx=5000,final_outny=5200,final_ra=204.268999,\
final_dec=-29.860656
```

Following the same logic, an ACS/WFC image may be aligned to a WFC3/UVIS image, though one may need to think about how to set the ***final_scale*** parameter since the native ACS/WFC plate scale, at 0.05 arcseconds/pixel, is larger than the WFC3/UVIS plate scale of 0.0396 arcseconds/pixel. If the frames have been dithered, it may be possible to set the ***final_scale*** to a value smaller than the native plate scale. For more information, refer to the examples on optimizing image sampling in Section 7.2 and Section 7.4.

The drizzled products are now aligned to the same reference pixel, as shown in the result below. These products can be compared to those from Section 7.5.5 which used a different reference pixel.

```
--> hselect *drz_sci*.fits $I,crpix1,crpix2,crval1,crval2 yes
#
f438w_pos1_drz_sci_v1.fits 2461.82 2597.75 204.269016 -29.860724
f814w_pos1_drz_sci.fits 2462.24 2597.76 204.268999 -29.860656

# Option A
f438w_pos1_drz_sci.fits 2462.24 2597.76 204.268999 -29.860656
f814w_pos1_drz_sci.fits 2462.24 2597.76 204.268999 -29.860656

# Option B
f438w_pos1r_drz_sci.fits 2500.00 2600.00 204.268999 -29.860656
f814w_pos1r_drz_sci.fits 2500.00 2600.00 204.268999 -29.860656
```

In **ds9**, the frames may now be aligned by physical pixels rather than by WCS. Comparing the position of a few stars in each image using **imexamine** will verify the alignment.

```
--> displ f438w_pos1r_drz_sci.fits 1 zs-zr-z1=-0.2 z2=0.5 \
ztran=log fill+
--> displ f814w_pos1r_drz_sci.fits 2 zs-zr-z1=-0.2 z2=0.5 \
ztran=log fill+
```

An optional step: to verify that there are no remaining residuals in the two aligned filter images, **tweakreg** can be used to prove the integrity of the results:

```
--> tweakreg.TweakReg('f438w_pos1r_drz_sci.fits',\
refimage='f814w_pos1r_drz_sci.fits',updatehdr=False,\
conv_width=3.5,threshold=5,shiftfile=True,\
outshifts='shift_438ref_pos1r_iter2.txt',nclip=10)
```

The drizzled science products are shown for comparison in Figure 7.34 and the 4-panel fit residual plot is shown in Figure 7.35. The images, as shown in the shift file below, are now aligned to ~0.1 pixel accuracy,

```
Image                dX         dY         drot      scale     xfit_rms  yfit_rms
f438w_pos1r_drz_sci  0.001162   -0.003424  0.000099  1.000000  0.105570  0.110072
```

Figure 7.34: A Comparison of the F438W and F814W Products

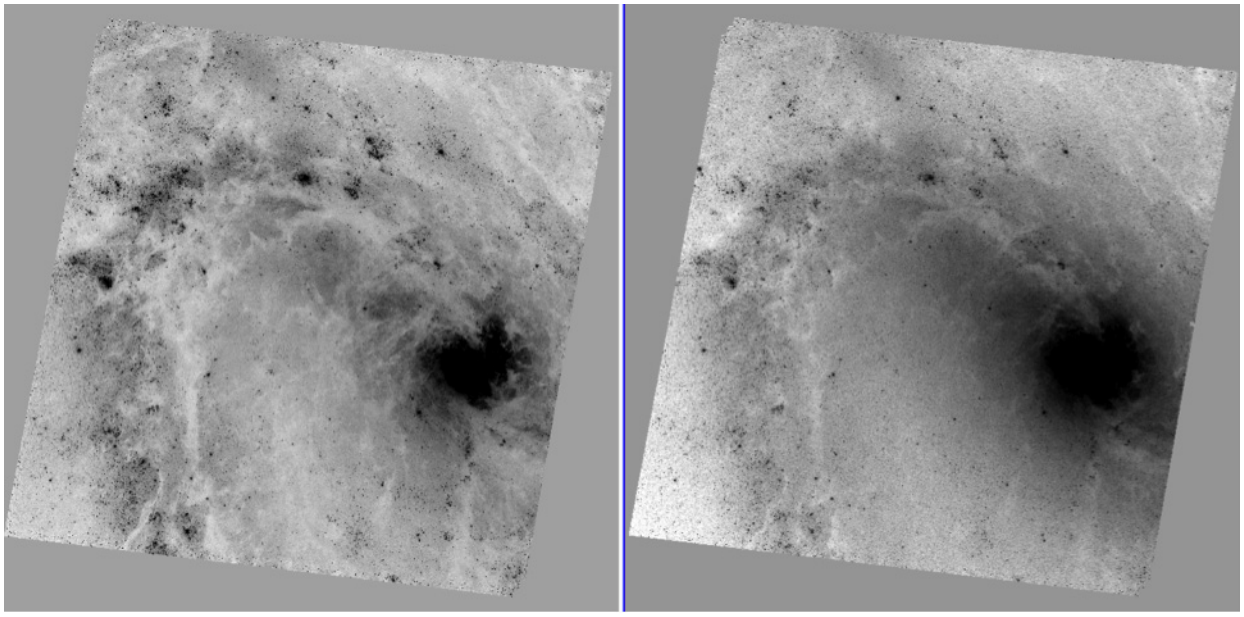
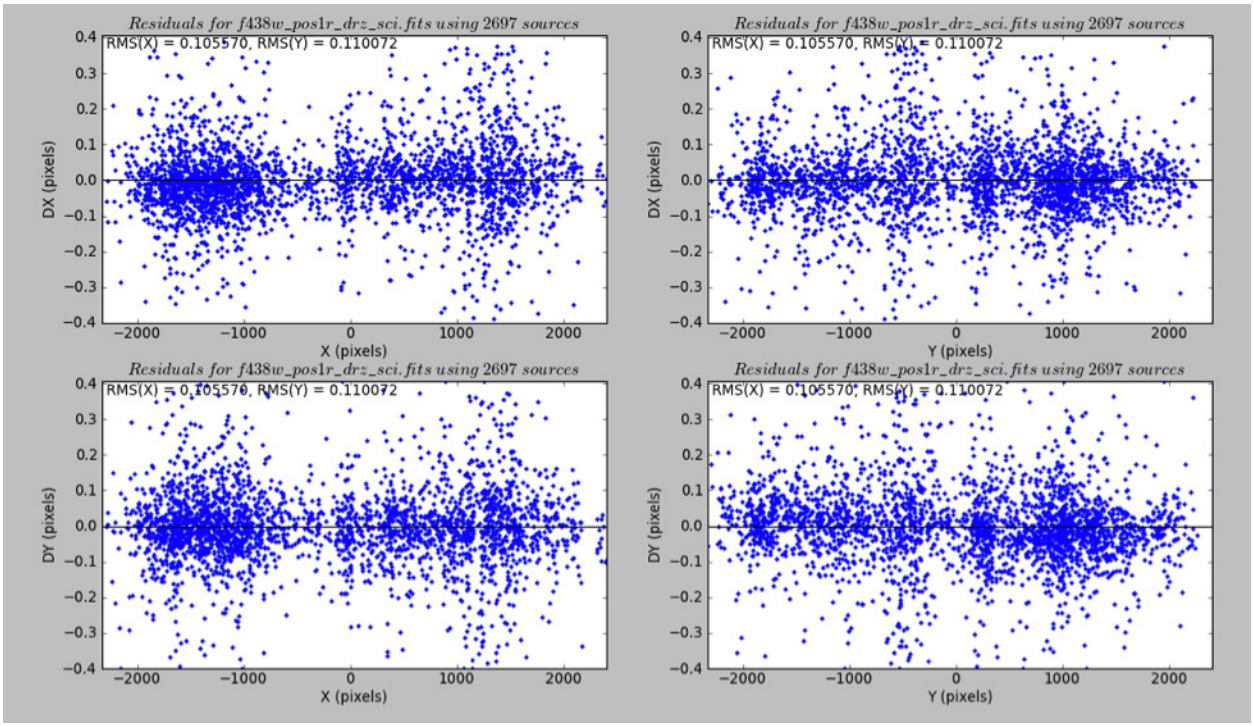


Figure 7.35: Astrometric Residuals for F438W with Respect to F814W



Relative alignment is accurate to ~0.1 pixels.