PyDrizzle and ACS Pipeline Calibration
Standard ACS Calibrations

The standard ACS pipeline calibration software was designed from the STIS pipeline software. This software performs a wide-range of calibration steps, including bias subtraction, dark subtraction, flat-fielding, and even cosmic-ray rejection.

ACS observations have some properties which standard pipeline processing performed for STIS or NICMOS could not address; specifically, significant optical distortion.

In addition, a 50 pixel wide gap between the chips in the Wide-Field Camera (WFC) would require the use of dithering in order to get data from behind the gap.

The tasks in the STSDAS Dither package provide the tools to address both of these issues.
Dither and the Pipeline

Drizzle was developed to combine the dithered observations of the Hubble Deep Field into a mosaic for analysis. It can also be used to remove distortion as well.

However, there are many problems with getting this software to run in the pipeline. A large number of parameters need to be determined, as documented in the 164-page HST Dither Handbook. IRAF CL scripts do not support error handling necessary for a pipeline. IRAF scripts tend to be fragile and can be hard to maintain. Drizzle only understands how to work on a single chip at a time. Drizzle can not properly apply distortion to subarray data at this time.

A new task was written to control the operation of ‘drizzle’ for use in the calibration pipeline for ACS: PyDrizzle.
What is PyDrizzle?

*PyDrizzle* \(\text{'Py" driz" zle'}\), n. 1. A program written in Python to be run under the PyRAF environment that automates the use of the Dither task ‘drizzle’.

PyDrizzle automates the use of ‘drizzle’ through the following features:

- interprets association tables to determine input images to be combined into a final product.
- understands how all the chips in an observation need to be combined into a single image for ACS (and WFPC2)
- computes the shifts between exposures from the WCS header keywords
- converts distortion coefficients for ACS into a format understood by ‘drizzle’
- only requires one command to run drizzle on all input images
- allows single exposures to be drizzled to separate outputs if desired
- also controls the use of blot, for use in cosmic-ray rejection
PyDrizzle and Association tables

ACS observations can be associated in the pipeline for a number of reasons:

- CR-SPLIT or REPEAT-OBS data
- Dither pattern specified

A simple CR-SPLIT, 2-point dither association table, *JNXXAA10T_ASN.FITS*:

<table>
<thead>
<tr>
<th>MEMNAME</th>
<th>MEMTYPE</th>
<th>MEMPRSNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNXXAAB1T</td>
<td>EXP-CR1</td>
<td>yes</td>
</tr>
<tr>
<td>JNXXAAB2T</td>
<td>EXP-CR1</td>
<td>yes</td>
</tr>
<tr>
<td>JNXXAAC1T</td>
<td>EXP-CR2</td>
<td>yes</td>
</tr>
<tr>
<td>JNXXAAC2T</td>
<td>EXP-CR2</td>
<td>yes</td>
</tr>
<tr>
<td>JNXXA11T</td>
<td>PROD-CR1</td>
<td>yes</td>
</tr>
<tr>
<td>JNXXA12T</td>
<td>PROD-CR2</td>
<td>yes</td>
</tr>
<tr>
<td>JNXXA10T</td>
<td>PROD-DTH</td>
<td>no</td>
</tr>
</tbody>
</table>

1\textsuperscript{st} CR product

2\textsuperscript{nd} CR product

Dither product
PyDrizzle Processing Steps

PyDrizzle performs a number of operations in preparation for running drizzle.

1. Determine input image(s) and name of output product
2. Interrogate each input image to determine the detector
3. Read each extension of each input image to extract WCS information
4. Read in distortion coefficients for each observation
5. Build a virtual ‘meta-chip’ which combines all chips from an exposure into a single image after applying distortion coefficients
6. Compare WCS information for each exposure to determine any shifts, scaling, and/or rotation needed to create final combined product
7. Sets up drizzle parameters for each chip based on info from previous step
8. Run drizzle for each chip
9. Combine all three products from drizzle to create a single distortion-corrected, multi-extension FITS image.
PyDrizzle and Exposures

Drizzle was only designed to operate on a single group or extension of an image at a time. Thus, it falls on the user to figure out (if at all possible) the shifts, output image size and plate scale for each chip or extension to combine a multi-chip observation into a single combined image.

A single distortion-corrected image was desired as the final product for the ACS calibration pipeline, therefore, PyDrizzle needed to know how to combine the chips into a single image. The IDCTAB, distortion coefficients reference table, not only contains the distortion coefficients, but also the absolute position of each chips reference point. This allows PyDrizzle to compute the offsets between the chips. This offset gets recorded in the coefficients file passed to drizzle for each chip.
PyDrizzle and Shifts

Image registration can pose serious problems when drizzling images, as slight offsets in pointing can result in multiple images. PyDrizzle addresses this in two ways:

• Determination of the shifts by PyDrizzle start with the WCS information read in from the image headers.
  – HST observations taken as part of a dither-pattern rely on the same guide stars, thereby minimizing pointing errors.
  – The same assumptions can not be made, however, for mosaics or images taken with different instruments at different epochs.

• PyDrizzle can read in delta’s from optional columns in the association table.
  – These columns provide the offset from the header WCS values necessary to accurately combine the images.
  – These offsets can be computed using any method desired by the user.
Running PyDrizzle

The actual operation of PyDrizzle has been designed to be as simple as possible. The default case only requires the name of the input image or association table as input, information the pipeline has when processing the data.

Taking the association table presented earlier as an example, we can run PyDrizzle to create a dither-combined product in just one command.

```bash
xena> pyraf
>>> stsdas
>>> analysis
>>> dither
>>> pydrizzle jnxxaal0t_asn.fits
(PyDrizzle messages printed here...)
>>> display jnxxaal0t_drz.fits[sci,1] 1
```
Where it is being used...

The ACS calibration pipeline currently uses PyDrizzle for:
- geometric distortion correction for single images
- dither-combining, and distortion correction for associated data

In addition, the Dither Working Group has been providing input into further development of PyDrizzle. Development has begun on:
- shift refinement with PyDrizzle
- cosmic-ray detection and removal using the dither package algorithms

A simple, proof-of-concept Python script has already been written by Anton Koekemoer using PyDrizzle to perform cosmic-ray rejection and dither-combination of ACS and WFPC2 data.