CHAPTER 1:

GETTING STARTED WITH MULTIDRIZZLE

In this chapter. . .

MultiDrizzle can process a set of ACS observations to generate a distortion-free, cosmic-ray clean final image. The case presented in this chapter only represents one fairly common situation for ACS users which requires the use of most of MultiDrizzle's functionality.

This chapter delves right into the usage of MultiDrizzle with modest explanation. For a more detailed description of the algorithms used in MultiDrizzle, see Chapter 4: Description of the MultiDrizzle Algorithms.

The use of MultiDrizzle using the Python interface requires that the MultiDrizzle code be installed in a directory accessible by Python. If that directory is not already included in the PYTHONPATH or default Python system path, you can point to it using:

```
>>> sys.path.insert(1,'/directory/with/MultiDrizzle/code')
```

MultiDrizzle can then be loaded using the usual import mechanism in Python:

```
>>> import multidrizzle
```

At this point, MultiDrizzle can be run on the calibrated images in the local directory. Interactive help can be obtained for MultiDrizzle using the built-in help mechanism:

```
>>> multidrizzle.help()
```

This wilc provide basic information on the methods available for use and their syntax, including the method used for bringing up the Python GUI interface for editing all the parameters.

1.2First Session...

Alc ACS data processed by the standard calibration pipeline has been processed by MultiDrizzle to remove geometric distortion, and to remove cosmic rays when combining associated images. Data taken using a standard dither pattern, using CR-SPLIT on a long

This example relies on data taken as part of the ACS ERO program. The association table, however, was manually created for this example to represent how the

this37.918.72 0 couleated3

spending the majority of time actually doing the image combination.

MultiDrizzle creates an instance of a MultiDrizzle object which contains all the methods necessary for editing the input parameters, computing the image combination parameters, then performing the actual cosmic-ray detection and image combination. All these functions get wrapped into one call through the PyRAF interface, making for simpler operation at the cost of the ability to inspect the object and its contents.

The processing starts with the creation of a MultiDrizzle object, which for the sake of this example will simply b inalled 'md':

```
-->  md = multidrizzle.MultiDrizzle('j8cw030x0_asn.fits',mdriztab=yes)
```

The specification of the 'mdriztab' parameter in the initialization demonstrates how any of the input parameters whose values need to

Both ways of running MultiDrizzle will produce the exact same output products using the same input _flt.fits files.

1.3MultiDrizzle Syntax

Operation of MultiDrizzle can be performed using either (or both) an IRAF parameter-based interface or a native Python command-line interface. The IRAF provides the capability to use PyRAF's graphical EPAR editor to set all the parameters necessary for running MultiDrizzle, then execute it all the way to completion _f one step. This method makes it easy to manage the input parameters through the graphical EPAR editor, but prevents introspection into or modification of the values

Table 2:Parameters for MultiDrizzle

Parameter	*Default Value*	*Description*
input	flt.fits	Input files: filename, wildcard suffix, or @list
output		Rootname for output drizzled products
mdriztab	no	Use table with multidrizzle parameters?
refimage		Name of image to use as reference WCS
runfile	multidrizzle.run	File for logginm (run jinedsTzzled commandTj -179.76 -17.04 Td (mdworkputlac)T

Parameter	Default Value	Description	inpdegrees)e-179.76 -17.041.44 driz_s

Parameter	*Default Value*	*Description*
expkeyword		**Exposure time keyword in header**

- **the IDCTAB reference table specified in the input image headers**
- **any**

group: *[format: string]*
A single FITS extension or

Value used to override instrument specific default gain values. The value is assumed to be in units of electrons/count. This parameter should not be populated if the *gainkeyword* parameter is in use.

gainkeyword*: [format: string]*

Keyword used to specify a value to be used to override instrument specific default gain values. The value is assumed to be in units of electrons/count. This parameter should not be populated if the *gain* parameter is in use.

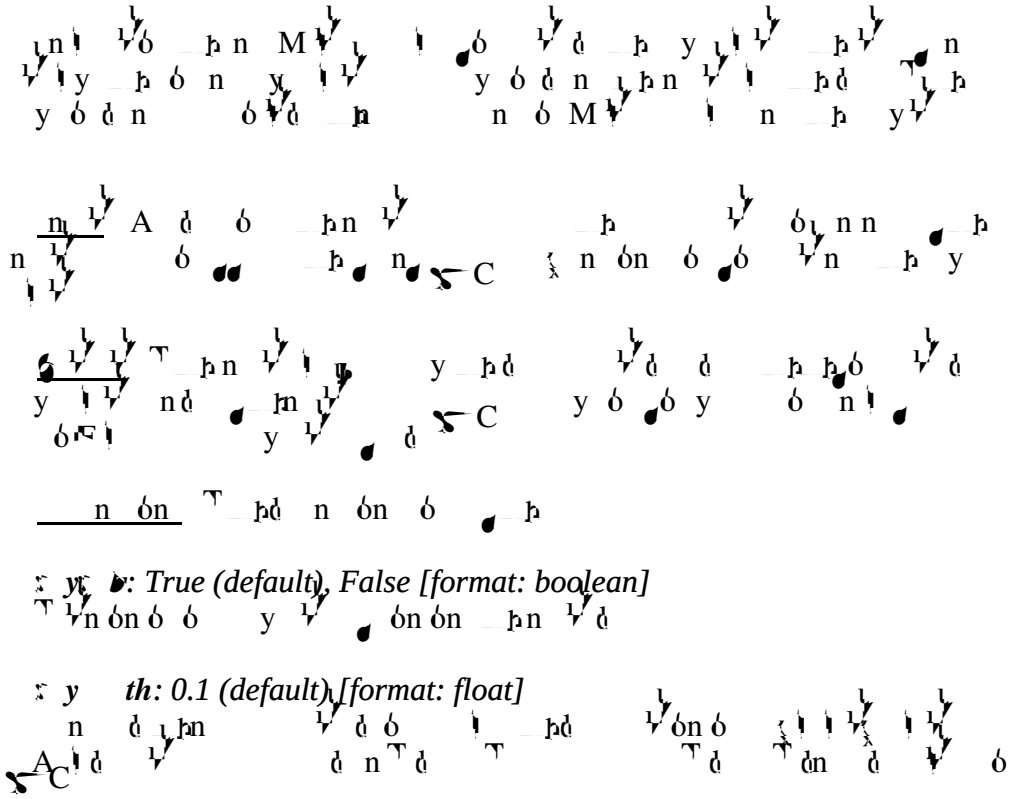
no : [format: float]

Value used to override instrument specific default readnoise values. The value is assumed to be in units of electrons. This parameter should not be populated if the *rnkeyword* parameter is in use.

rnkeyword*: [format: string]*

Keyword used to specify a value to be used to override instrument specific default readnoise

example, each



: True (default), False [format: boolean]

th: 0.1 (default) [format: float]

Name of header keyword which records the sky value already subtracted from the image by the user.

Parameters: driz_separate, driz_sep_outnx, driz_sep_outny, driz_sep_kernel, driz_sep_scale, driz_sep_pixfrac, driz_sep_rot, driz_sep_fillval

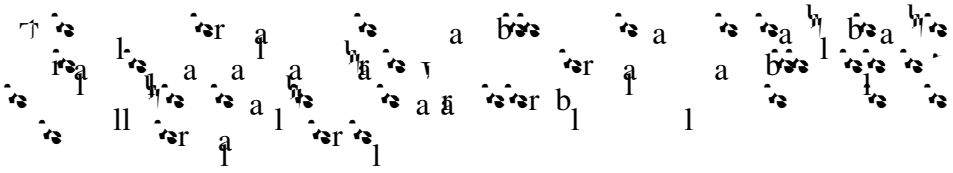
Processing: Each input image gets drizzled onto separate copies of the output frame. These copies when stacked would correspond to the final combined product. As separate image, though, they allow for treatment of each input image separately in the undistorted, final WCS system.

These images provide thematiod fo(n)T0theTj 24 0 Tnecessaryned

***median_newmasks**: True (default), False [format: boolean]*

The user can specify whether or not to create new mask files when creating the median image. These masks are generated

Sets the upper threshold for clipping input pixel values during image



- ***in c** : 1.0 (default) [format: float]*

Scaling factor applied to the derivative in 'driz_cr' when detecting cosmic-rays. This parameter gets passed directly to 'driz_cr'; see the help file for 'driz_cr' for further discussion of this parameter.

8 Final image combination

Parameters: driz_combine, final_wht_type, final_outnx, final_outny, final_kernel, final_scale, final_pixfrac, final_rot, final_fillval

Processing: This step performs the final image combination of the original input images (or their copies) using the updated mask files to remove any cosmic-rays. The output frame, just like the single drizzle

- o
- a
- anczo

and d c on o n zz ng o a a c on

final_scale*: None (default) [format: float]*

L n a z o o c n a c cond o na co n d od c d a a o N c a o nd o d ca o ag a co d y y zz