CCDReduction Documentation

Jurriaan de Gier - s1619179

May 26, 2019

1 Introduction

CCDReduction is a package (in this case a set of python files) that can be imported and used to open and reduce images taken using a CCD camera. Within multiple fields of science, data is recorded using CCD cameras which then are reduced to the so called science image.

2 Dependencies

The classes have the following dependencies:

numpy numpy is used for computations with numpy arrays and is essential to the package.

matplotlib matplotlib is used to create figures and is essential to the package.

os os is used to find the files at a path and is essential to the package.

scipy scipy is used for odd functions such as the bessel functions and the optimize package. It is not essential is depended on in the Laser Reduction files (such as Focus and LaserReducer)

astropy astropy is used to load Fits files and is not essential to the inner workings of this class (only FitsLoader depends on it). This is done by design.

emcee emcee is used for an mcmc optimalization procedure in the FocusFitter classes and is not essential.

corner corner is used to visualize the mcmc optimalization and is not essential.

3 Overview

The CCDReduction package is split in 3 parts: The Core files, the CCD classes, and the Fits classes. An important not is that the CCD classes contain all of the important functions and variables, but cannot be executed by themselves. the Fits classes are subclasses of the CCD classes and enable the opening of Fits files (used a lot in astronomy and related sciences). For more information see the Fits classes.

3.1 Core files

3.1.1 Data

The Data file contains a class which creates the Data object which has the following calling signature:

```
dataobject = Data(data, time, identifiers)
```

with ***data*** a list containing N numpy arrays, ***time*** a list containing N integers or floats and ***identifiers*** a list containing N items (such as strings, numbers, headers from Fits files, etc). The data object takes 3 lists like[1] objects, which must have the same lengths. The Data object has the following properties:

data A function which returns the data list: `Data.data()`

time A function which returns the time list: `Data.time()`

identifiers A function which returns the identifiers list: `Data.identifiers()`

shape A list that contains the shapes of all data arrays: `Data.shape`

medians A function which returns an array containing the median value of every data array: `Data.medians()`

[1] lists, tuples and numpy arrays work

maximums Same as medians but with the maximum values: `Data.maximums()`

minimums Same as medians but with the minimum values: `Data.minimums()`

it also contains a number of "magic" methods such as `len(Data)` returns N.

3.1.2 ErrorGenerator

ErrorGenerator is an abstract class which empty subclasses can derive from to create a Error class.

3.1.3 support_functions

The support_functions python file contains a number of functions that are used throughout the classes. An important note is that the functions contained in this class must always work the same way, and such are mostly mathematical formulas. The following

3.2 CCD classes

3.2.1 CCDReductionObject

The CCDReductionObject file contains the abstract CCDReductionObject class and its CCDBias, CCDDark and CCDFlat subclasses. The classes have the following calling signature:

```
bias = CCDBias(masterpath, filespath=None)
```

The *`masterpath`* is the path to the location where the "master" pcl file is/will be stored, *`filespath`* is the path to where the files containing the corresponding CCD images of the class are located (for example for the CCDBias class *`filespath`* is the folder where all the bias images are located, the `master_bias` function then combines all of the bias images into a single "master" bias which can then be used to reduce the CCD images you want to analyze). If *`filespath`* is not given it is taken to be the same as *`masterpath`*. Both paths must obviously be strings.

The objects created from the CCDBias, CCDDark, and CCDFlat classes can creat and load the bias, dark and flat pcl files. These files contain Data objects used to reduce the CCD images to the science image in CCDReducer.
All 3 subclasses have the following functions (here indicated for the CCDBias subclass):

load A function that loads the corresponding pcl master file: `bias.load()`

create A function that creates the corresponding pcl master file: `bias.create()`

The way that the master file is created differs between the subclasses: CCDBias opens all files at the *`filespath`* and combines these into a single pcl file; CCDDark opens all files AND the "master" bias pcl file at the *`masterpath`* to create a single pcl file; and CCDFlat opens all filesAND the "master" bias and dark pcl files to create a single pcl file.

So each class can create a master file and load it.

3.2.2 CCDReducer

The CCDReducer class loads a single file and optionally reduces with the created master files from the ReductionObjects. Its calling signature is:

```
f = CCDReducer(filepath, masterpath=None, savepath=None)
```

`filepath` is the full path to the file to be reduced. *`masterpath`* is the path to the location of the master files and *`savepath`* is the path to the location where figures can be saved. When one is not given it is taken to be *`filepath`*.
CCDReducer has the following attributes:

Data The data object holding the image: `f.Data`

imshow A function which creates a figure to view the image. It takes the optional *cmap*, *log* and *title* arguments: `f.imshow( cmap=None, log=False, title=''Image of Data'')`

imsave A function which creates a figure and saves it. It takes the same arguments as imshow as well as: *savename* and *extension*: `f.imsave(cmap=None, log=False, title=''Image of Data'', savename=''CCDPic'', extension=''.png'')`

3.2.3 CCDLaserReducer

The CCDLaserReducer is a subclass of CCDReducer specifically create to reduce images captured of laser light. Its calling signature is:

```
f = CCDLaserReducer(filepath, masterpath=None, savepath=None, pixel_size=1)
```

filepath, *masterpath* and *savepath* work as in CCDReducer. *pixel_size* is a float or integer indicating the length of a pixel (pixels are assumed to be square shaped).
CCDLaserReducer has the following atributes:

Data The data object holding the image: `f.Data`

imshow A function which creates a figure to view the image. It takes the optional *cmap*, *log* and *title* arguments: `f.imshow( cmap=None, log=False, title=''Image of Data'')`

imsave A function which creates a figure and saves it. It takes the same arguments as imshow as well as: *savename* and *extension*: `f.imsave(cmap=None, log=False, title=''Image of Data'', savename=''CCDPic'', extension=''.png'')`

slicingshow A function which creates a figure to view the image: `f.imshow()`

slicingsave A function which creates a figure and saves it: `f.imsave()`

powershow A function which creates a figure to view the image: `f.imshow()`

powersave A function which creates a figure and saves it: `f.imsave()`

cum_power_fraction_withing_area A function which creates a figure to view the image: `f.imshow()`

power_within_area A function which creates a figure and saves it: `f.imsave()`