Observational Astrophysics

Practice 2: Calibration and analysis of astronomical images

1 Introduction

The object of this practice is the analysis and the elimination of the instrumental signature in astronomical images. We will use images obtained with the Apogee CCD camera on the 60 cm. TROBAR telescope at the Aras de los Olmos Observatory. The work will be done using the IRAF software package.

We can access IRAF on the university's Ubuntu Linux virtual desktop, following the link:

https://escritorio.uv.es/portal/webclient/index.html

2 IRAF tools

To initialize IRAF, click on the corresponding icon in the Linux virtual desktop applications. A terminal will open with the "command language" prompt

```
ecl>
```

First, type the instruction:

```
set stdimage=imt2048
```

Next, go to the working directory, where you have previously downloaded the images to be analyzed, and open a graphical window to view them with the instruction:

```
!ds9 &
```

Analysis and correction techniques:

- Image statistics:

  ```
  imstat image_name
  ```

- Image analysis: we use the IRAF command "imexamine". To do so, we type:

  ```
  display image_name
  imexam
  ```

  and then we left-click on the top bar of the window containing ds9. To exit "imexamine" type `q`.

  Within imexamine, we can perform the following operations:

c plot a vertical cut of the image
e contour map
j one-dimension gaussian fit in the horizontal direction
k one-dimension gaussian fit in the vertical direction
l plot an horizontal cut of the image
r radial gaussian fit
s surface plot
q quit imexamine

- For arithmetic operations with images we use the "imarith" task, whit the following syntax:

  `imarith operand1 op operand2 result`

  where:

  - "operand1", "operand2" are images or constant values.
  - "op" is the operator to be applied. The posible operators are: +, -, *, /, min y max.
  - "result" is the resulting image.

To end the session with IRAF type:

```
logout
```

3 Development of the practice

The first image to be displayed is an image of the polarization current, or "bias". We open it with the command

```
display Bias-001 1
```

First, we perform the statistics of the image, with the instruction

```
imstat Bias-001
```

Next, we analyze it with the program "imexamine"

```
imexam
```

Once this command is executed, we activate the graphic window with a mouse click, and we execute any of the commands described above. With the bias image, horizontal and vertical cuts are useful, with the commands "l" and "c". We exit "imexam" with "q".

We perform a similar study of the dark current images, whose name begins with "dark", and of the uniformly illuminated fields, whose name begins with "flat".

Finally, we analyze an astronomical image. Any of those that begin with "NGC" in the practice data. In its study with imexam we will use, in addition to "l" and "c", the commands "e", "j", "k", "s" and "r", centering the cursor on different stars in the image.

4 Astronomical image calibration with IRAF

The collection of tasks that we will use to remove the instrumental signature is found in the "ccdred" package. We access it by typing:

```
noao
imred
ccdred
```

First of all, we must check that the system is able to recognize the main characteristics of our images, such as the type of image corresponding to each file. We can check this by typing

```
setinstrument direct
<ctrl>+d
<ctrl>+d
```

and then

```
ccdlist
```

and answering "*" to the name of the image. A list like the one in Figure 1 will appear. We must check that the system recognizes the image types, which must appear within the last brackets.

When everything is in order, we proceed to correct the instrumental signature. We do this in three steps:

1. Bias current correction:

   First, we obtain an average of the bias current images. We do this with the command:

   ```
   zerocombine
   ```

   which calculates an average bias current image called Zero. We can see with "imstat" that its standard deviation is much smaller than in the individual images (see examples in Figure 2).

   Next, we subtract the average bias current from the rest of the images using "ccdproc". To do so, we edit the parameters of "ccdproc"

   ```
   epar ccdproc
   ```

   and indicate that the only correction to be performed in this first step is the bias correction. We therefore configure the parameter file as it appears in Figure 3.

   Once configured, we exit the editor with

   ```
   <ctrl>+d
   ```

   and run the task:

   ```
   ccdproc
   ```

```
ccdred> ccdlist
CCD images to listed (*):
Warning: Cannot open image (Documents)
Flat1.fit[1024,1024][real][flat][][T]:
Flat2.fit[1024,1024][real][flat][][T]:
Flat3.fit[1024,1024][real][flat][][T]:
Flat4.fit[1024,1024][real][flat][][T]:
Flat5.fit[1024,1024][real][flat][][T]:
Flat6.fit[1024,1024][real][flat][][T]:
Flat7.fit[1024,1024][real][flat][][T]:
Flat8.fit[1024,1024][real][flat][][T]:
bias001.fit[1024,1024][real][zero][][T]:
bias002.fit[1024,1024][real][zero][][T]:
bias003.fit[1024,1024][real][zero][][T]:
bias004.fit[1024,1024][real][zero][][T]:
bias005.fit[1024,1024][real][zero][][T]:
bias006.fit[1024,1024][real][zero][][T]:
bias007.fit[1024,1024][real][zero][][T]:
bias008.fit[1024,1024][real][zero][][T]:
bias009.fit[1024,1024][real][zero][][T]:
bias010.fit[1024,1024][real][zero][][T]:
Warning: Cannot open image (bin)
dark001.fit[1024,1024][real][dark][][T]:
dark002.fit[1024,1024][real][dark][][T]:
dark003.fit[1024,1024][real][dark][][T]:
dark004.fit[1024,1024][real][dark][][T]:
dark005.fit[1024,1024][real][dark][][T]:
dark006.fit[1024,1024][real][dark][][T]:
dark007.fit[1024,1024][real][dark][][T]:
dark008.fit[1024,1024][real][dark][][T]:
dark009.fit[1024,1024][real][dark][][T]:
dark010.fit[1024,1024][real][dark][][T]:
i01.fit[1024,1024][real][object][][T]:
i02.fit[1024,1024][real][object][][T]:
i03.fit[1024,1024][real][object][][T]:
i04.fit[1024,1024][real][object][][T]:
i05.fit[1024,1024][real][object][][T]:
i06.fit[1024,1024][real][object][][T]:
i07.fit[1024,1024][real][object][][T]:
i08.fit[1024,1024][real][object][][T]:
i09.fit[1024,1024][real][object][][T]:
i10.fit[1024,1024][real][object][][T]:
i11.fit[1024,1024][real][object][][T]:
i12.fit[1024,1024][real][object][][T]:
```

Figure 1: Image list with `ccdlist`

```
Oct  1 15:59: IMCOMBINE
  combine = average, scale = none, zero = none, weight = none
  reject = minmax, nlow = 0, nhigh = 1
  blank = 0.
                Images
            bias001.fit
            bias002.fit
            bias003.fit
            bias004.fit
            bias005.fit
            bias006.fit
            bias007.fit
            bias008.fit
            bias009.fit
            bias010.fit

  Output image = Zero, ncombine = 10
ccdred> imstat bias001
#               IMAGE      NPIX      MEAN    STDDEV       MIN       MAX
              bias001   1048576     954.6     15.86      886.     4861.
ccdred> imstat bias007
#               IMAGE      NPIX      MEAN    STDDEV       MIN       MAX
              bias007   1048576     943.2     14.05      883.     2123.
ccdred> imstat Zero
#               IMAGE      NPIX      MEAN    STDDEV       MIN       MAX
        _        Zero   1048576     943.7     5.136     921.2     1502.
```

Figure 2: Mean bias current with `zerocombine`

The correction is performed automatically on all images. If we list them again with

`ccdlist`

We see that the images are listed as bias corrected.

2. Dark current correction:

   The procedure is very similar. First we obtain a medium dark image with

   `darkcombine`

   This image is called "Dark". In the averaging process it has been taken into account that the average level of the dark current in the images depends on the exposure time. The average Dark image is the dark current per unit of time. When correcting for dark current, this average current will be multiplied by the exposure time of each image.

   Next we correct for dark by going through "ccdproc" a second time. Before we have to edit the parameters again, putting "no" in the bias correction, "yes" in the dark correction and entering the name of the mean dark file "Dark". We run

```
                          IRAF
       Image Reduction and Analysis Facility
PACKAGE = ccdred
   TASK = ccdproc

images  =                       List of CCD images to correct
(output =                      ) List of output CCD images
(ccdtype=                      ) CCD image type to correct
(max_cac=                     0) Maximum image caching memory (in Mbytes)
(noproc =                    no) List processing steps only?

(fixpix =                    no) Fix bad CCD lines and columns?
(oversca=                    no) Apply overscan strip correction?
(trim   =                    no) Trim the image?
(zerocor=                   yes) Apply zero level correction?
(darkcor=                    no) Apply dark count correction?
(flatcor=                    no) Apply flat field correction?
(illumco=                    no) Apply illumination correction?
(fringec=                    no) Apply fringe correction?
(readcor=                    no) Convert zero level image to readout correction?
(scancor=                    no) Convert flat field image to scan correction?

(readaxi=                  line) Read out axis (column|line)
(fixfile=                      ) File describing the bad lines and columns
(biassec=                 image) Overscan strip image section
(trimsec=                 image) Trim data section
(zero   =                  Zero) Zero level calibration image
(dark   = []                   ) Dark count calibration image
(flat   =                      ) Flat field images
(illum  =                      ) Illumination correction images
(fringe =                      ) Fringe correction images
(minrepl=                   1.) Minimum flat field value
(scantyp=             shortscan) Scan type (shortscan|longscan)
(nscan  =                     1) Number of short scan lines

(interac=                  yes) Fit overscan interactively?
(functio=            chebyshev) Fitting function
(order  =                     1) Number of polynomial terms or spline pieces
(sample =                     *) Sample points to fit
(naverag=                     1) Number of sample points to combine
(niterat=                     1) Number of rejection iterations
(low_rej=                    3.) Low sigma rejection factor
(high_re=                    3.) High sigma rejection factor
(grow   =                    0.) Rejection growing radius
(mode   =                    ql)
```

Figure 3: `ccdproc` parameter file to perform bias correction

`ccdproc`

and then

`ccdlist`

to verify that the correction has been carried out correctly.

3. Correction of sensitivity and illumination variatios ("Flat field")

   We proceed as in the previous cases. The program that calculates the average flat is

   `flatcombine`

   Before running it we have to edit its parameter file:

   `epar flatcombine`

   and indicate the values of the gain and the read noise, which for the CCD used (Finger Lakes Instruments) are 4.0 e^-/ADU y 10.0 e^- respectively. We also have to make sure that the "process" parameter has the value "no".

   We do the flat correction again with "ccdproc". Before we edit the parameters, putting "no" in the dark correction, "yes" in the flat correction and introducing the name of the average flat file "Flat.fits".

   In the case that we are doing photometry with several filters, "flatcombine" will create several average flats, which it will call "FlatB.fits", "FlatV.fits", etc. If so, in the "ccdproc" configuration, we will enter "Flat*" as the name of the average flat. After this, we run

   `ccdproc`

   and then

   `ccdlist`

   to check that the correction has been performed correctly. Finally, we examine the corrected images, comparing them with the raw images.