

TP – imagerie numérique

Ce TP a pour objectif de vous familiariser avec l'acquisition des images numériques et leurs manipulations. Vous étudierez la phase de pré-traitement des images. Vous vous familiariserez aussi avec certains types d'aberrations propres à l'utilisation de capteurs CCD.

La caméra utilisée pour les TP est une caméra Audine, équipée d'un capteur CCD Kodak AF400.

Matériels utilisés :

- 1 raspberry fonctionnant sous Linux avec le logiciel Audela qui permet de piloter la caméra CCD;
- 1 caméra Audine (Figure 1) avec une optique pour faire de l'imagerie;
- 1 boîtier d'alimentation de la caméra (Figure 2);
- 1 nappe pour raccorder la caméra au raspberry.

Le boîtier d'alimentation possède un interrupteur général placé à l'arrière. La mise sous tension de la caméra se fait dans l'ordre suivant : allumer l'alimentation +/-15 V, puis celui du ventilateur. **Ne pas allumer le refroidissement par le module thermo-électrique Peltier** (qui permet de refroidir le capteur à une température en dessous de 0°C). Pour éteindre la caméra, merci de procéder dans l'ordre inverse i.e. éteindre le ventilateur puis l'alimentation +/-15 V.



Figure 1 – image du boîtier caméra. Pour ce TP une optique est également présente devant le capteur afin de pouvoir faire des images.

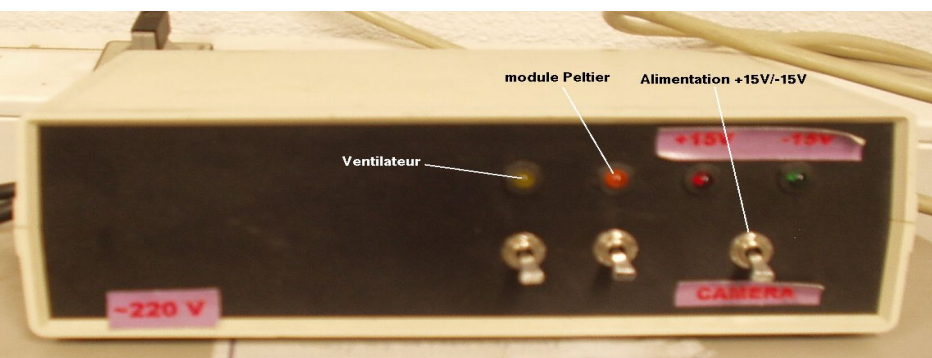


Figure 2 – Boîtier d'alimentation de la caméra (interrupteur général au dos du boîtier).

A la fin du TP, merci de rassembler l'ensemble de vos images et programmes Python dans un répertoire portant vos noms sur le Bureau. Vous indiquerez sur votre rapport à quoi correspondent les différents programmes et images. Pour vos compte-rendus merci d'inclure les images prises et traitées.

1. Acquisition avec la caméra CCD

La caméra est réglée. Il est donc impératif de ne pas la déplacer, ni modifier le réglage de la focale. La caméra étant très sensible à la lumière, vous ferez les images avec la pièce dans l'obscurité (rideaux et lumière fermés, porte entrouverte). **Veiller à ce que les conditions d'éclairage ne changent pas trop au cours de vos prises d'images.**

Pour ouvrir Audela, cliquer sur l'icône **TP CCD** sur le bureau, puis cliquer sur **Execute**. En haut de l'interface utilisateur, sélectionner **Method = full_frame** pour faire des images. Le **temps de pose** (en seconde) et le **binning** (un mode de lecture de la caméra) peuvent être ajustés.

Les images prises sont au format **.fits** (*Flexible Image Transport System* – <http://fits.gsfc.nasa.gov/>) et sont à sauvegarder dans le répertoire **/home/user/Documents/guitastro/products**.

- 1) Faire une image du rond blanc sur fond noir fourni sur la paillasse sur 1 seconde et une autre sur 10 secondes en binning 1x1.

Que constatez-vous sur ces deux images ? Est-ce que le temps de lecture change entre ces deux acquisitions ? Cette aberration s'appelle le "*smearing*". Expliquer son origine.

- 2) Faire une image du rond blanc sur fond noir sur 5 secondes en binning 1x1 et en binning 2x2. Estimer la définition des deux images.

Expliquer ce qu'est le binning et quel est son intérêt.

2. Calibration

Un pixel d'une image brute, pour un temps de pose t , comporte les composantes suivantes

$$\text{Brute} = \text{Bias} + \text{Thermique} + \text{lumière} * \text{réponse} * t$$

Si l'on effectue une pose, de même temps de pose dans le noir absolu, on obtient le dark:

$$\text{Dark} = \text{Bias} + \text{Thermique}$$

Bias = signal artificiel injecté dans chaque puits de potentiel de la matrice active. Ce signal sert à indiquer le niveau zéro (ou de référence) dans chaque puits.

Thermique = électrons produits par agitation thermique.

Si l'on effectue une pose, de même temps de pose sur une plage de lumière uniforme (2 feuilles de papier blanc superposées par exemple), on obtient le flat :

$$\text{Flat} = \text{Bias} + \text{Thermique} + \text{cste} * \text{réponse} * t$$

Le flat permet de corriger l'image acquise de la réponse ("empreinte") spatiale du CCD. Cette "empreinte" est liée au fait que tous les photo-sites de la matrice active n'ont pas les mêmes performances i.e. ne créeront pas le même signal pour le même flux incident.

A partir des images Dark et Flat et de leur expression, en déduire une méthode pour corriger l'image Brute des charges thermiques et de la non uniformité de réponse des photo-sites.

L'image ainsi corrigée s'appelle l'image pré-traitée.

2.1. Acquisition des images de calibration

Vous effectuerez toutes vos acquisitions en **binning 2x2**. La caméra n'étant pas équipée d'un obturateur, les étudiants devront assurer cette fonction durant la phase de lecture en collant le cache fourni sur la paillasse à l'objectif tout en veillant à ne pas endommager le matériel. Pour ce faire, vous chronométrez le temps de pose choisi avant de mettre le cache et ce jusqu'à ce que l'image apparaisse à l'écran.

- 1) Faire une acquisition avec la lumière allumée. L'image obtenue est dite saturée. Noter la valeur de l'intensité maximale renvoyée par le convertisseur analogique/numérique (déplacer le curseur sur l'image à l'aide de la souris). A quoi correspond la saturation ?
- 2) On commencera par faire l'acquisition du flat en optimisant le temps de pose (avoir la plus grande dynamique possible en ne saturant pas). Sauver cette image sous le nom **Flat**. On pourra ouvrir la porte afin de laisser passer suffisamment de lumière tout en évitant la saturation.
- 3) Faire l'acquisition du **Dark** (toujours en binning 2x2). Sauver cette image sous le nom **Dark**. Quelle durée avez-vous choisi pour le temps de pose ? Justifier votre réponse (discuter en avec l'enseignant(e)). Commenter l'image obtenue.
- 4) Enfin, faire l'acquisition d'une image d'un des supports fournis. Préciser le temps de pose choisi et pourquoi. Sauver cette image sous le nom **Image**.
- 5) Commenter l'image brute "**Image**" obtenue avec la caméra CCD en la comparant à l'image d'origine.

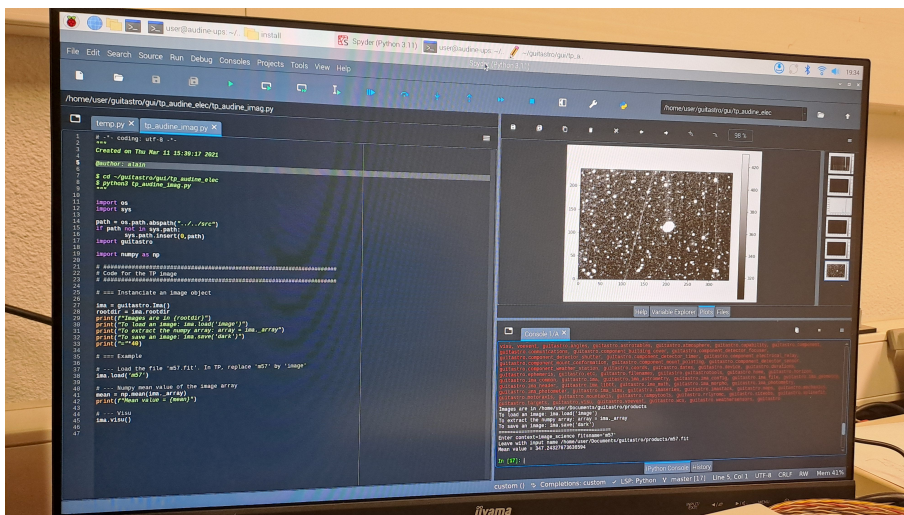
2.2. Utilisation de Python

Ouvrir un terminal sur le Bureau. Taper **cdprog** pour aller dans le répertoire (`~/guitastro/gui/tp_audine_elec`) dans lequel se trouve le fichier **tp_audine_imag.py**.

Ouvrir **Spyder** en allant dans la barre de menu (cliquer sur le "rasberry", puis **Programming** et **Spyder**).

Une fois l'interface ouverte, par défaut le programme **Python tp_audine_imag.py** est ouvert. Si ce n'est pas le cas, l'ouvrir.

Ce programme vous donne les bases pour manipuler les images .fits, les visualiser, etc.



Pour exécuter le programme `tp_audine_imag.py`, cliquer sur l'icône **Play**.



Le programme s'exécute sur la console en bas à droite. Par défaut, il va utiliser l'image `m57.fits` qui se trouve dans le répertoire `/home/user/Documents/guitastro/products`.

Une fenêtre montrant l'image va s'ouvrir. Pour récupérer la main sur la console fermer l'image qui va ensuite s'afficher dans la partie haute à droite.

- Pour afficher une courbe **curve** depuis la console importer la librairie **matplotlib**

```
import matplotlib.pyplot as plt
```

```
plt.plot(curve)
plt.show()    ⇒ Affichage de la courbe
```

- `ima.load('nom_fichier_image')` permet d'extraire l'image du fichier fits
- `array = ima._array` ⇒ Cela permet de copier le tableau 2D dans la variable **array**
- Pour faire des soustractions et des divisions, utiliser les fonctionnalités de la librairie **numpy**.
Soit 2 matrices **A** & **B** de même taille, `img = A + B` et `img = A/B`
- Pour visualiser les images, taper `ima.visu()` ⇒ L'image apparaît en dégradé de N&B avec les seuils min et max pris sur l'image.

Possibilité d'ajuster les seuils en tapant `ima.visu(seuil_min, seuil_max)`

- Sélection d'une ligne de l'image, taper `line = array(200)` ⇒ Intensité pour la ligne verticale 200 de l'image **array**, puis `plt.plot(line)`

```
temp.py x tp_audine_imag.py x
1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Mar 11 15:39:17 2021
4
5 @author: alain
6
7 $ cd ~/guitastro/gui/tp_audine_elec
8 $ python3 tp_audine_imag.py
9 """
10
11 import os
12 import sys
13
14 path = os.path.abspath("../src")
15 if path not in sys.path:
16     sys.path.insert(0, path)
17 import guitastro
18
19 import numpy as np
20
21 # =====
22 # Code for the TP image
23 # =====
24
25 # === Instanciate an image object
26
27 ima = guitastro.Ima()
28 rootdir = ima.rootdir
29 print(f"Images are in {rootdir}")
30 print("To load an image: ima.load('image')")
31 print("To extract the numpy array: array = ima._array")
32 print("To save an image: ima.save('dark')")
33 print("=="40)
34
35 # === Example
36
37 # --- Load the file 'm57.fit'. In TP, replace 'm57' by 'image'
38 ima.load('m57')
39
40 # --- Numpy mean value of the image array
41 mean = np.mean(ima._array)
42 print(f"Mean value = {mean}")
43
44 # --- Visu
45 ima.visu()
46
47
```

2.3. Fonction de calibration

- 1) Expliquer comment vous allez utiliser les images **Dark** et **Flat** pour corriger l'image **Brute**.
- 2) Ecrire un programme Python rassemblant les lignes de commandes permettant de réaliser le pré-traitement de l'image brute acquise en utilisant les images **Flat & Dark**.
- 3) Afficher les images brute et pré-traitée en ajustant les seuils en intensité. Commenter l'image pré-traitée.

3. Aberrations des images numériques

L'objectif dans cette section est de vous amener à reconnaître et à comprendre l'origine de certaines aberrations induites par l'utilisation d'un capteur numérique CCD. Vous vous intéresserez à 2 types d'aberrations supplémentaires que vous essayerez de diagnostiquer. L'enseignant vous indiquera le protocole à suivre.