Astrometry from CCD Images

Darya Zanjanpour
Darya.zanjanpourmobarhanfomani@mail.utoronto.ca

December 5, 2019

Abstract

For this lab, by investigating special mathematical methods and principles, which is well-known as "Astrometry", on the NGC1337 and 26 Prosperina CCD images data, that are provided form the Dunlap Institute telescope (DIT), the students can locate the relative astronomical object that they are interested in. Besides, by comparing the different data sets of an astronomical object in a period they are able to evaluate the proper motion of any object, using the relevant position and considering the relevant time properly. By following the steps, first, the students reduced their Dunlap telescope data and applied the systematic corrections including affected fields, second, they compared their image with the digital sky survey and validated the field of their interest using some well-matched method due to cross-correlate the stars in their images with the reference's one which here is USNOB-1 star catalogue. Moreover, by performing the Linear Square Fit they found the relative plate constant for each image and after examining the residuals by comparing the calculated data set and the reference, as a final result, they measured the corresponded position and furthermore the proper motion of astronomical objects which in this case for NGC1337 is up to finding position and for 26 Prosperina is up to finding proper motion per $arcsec/hr$. Briefly, The main purpose of this lab is to provide enough information and pre-data sets to guide the students into track the position of some minor planets and asteroids, based on the celestial coordinate system. Secondly, finding the proper motions of the observed asteroids by the same procedure, using the CCD images. In this lab, the students examined different methods for data analyzing due to determine the properties of the imaging camera and CCD and astronomical databases which improves their ability to analyze and interpret different data sets.

1 Introduction

As a formal definition, Astrometry is one of the astronomy's divisions established on meticulous measurements based on current location and movement of different celestial bodies such as stars and galaxies. The output of astrometrical evaluation delivers useful data sets about the physical and kinematic features from the solar system's origins up to the milky way.

Due to finding the proper motion the several steps must have been done in order. This lab has been done in 3 main sections due to guide the students step by step. First, in order to practice the relevant astrometry methods have performed on the NGC1337 data due to evaluate the position of NGC1337 relative to the provided CCD image. As a next main step, same procedure have redone for 26 Prosperina for 4 different images in order to find the relative positions in 4 different sky night. For the last step, by recombining the 4 positions which evaluated from the 26 Prosperina CCD pictures the proper motion calculated for this astronomical object. For evaluating the position for each of astronomical objects the following steps have done in order:

- Running the FITS files from the CCD images on the $50 - cm$ telescope and displaying the data using the Python PyFITS library (Using python program) (§A)
- Reducing the relative Dunlap telescope data set and applying corrections which will be discussed in the Data Acquisition section completely
- Comparing the image with the USNOB1 Catalogue and measuring the position of the stars which will be discussed in the Calculation and modelling section

- Computing the "Standard coordinates" for the USNOB1 stars and matching the founded list which will be discussed in the Calculation and modelling section

- Performing a "linear least-squares fit" to find the "plate constants

- Checking the residuals between the measurements and the fit

Also for determining the asteroid position, the position data sets will be recombined and some relative calculation will be useful due to finding the proper motion which will be discussed more in the Discussion section.

This lab report has been done by Darya Zanjanpour, although, for a basic observation such as comparing the python codes and discussing about the statistical methods which has used for data reduction, the members of group C worked as a team together.

2 Equipment

The equipment which used in this experiment, are as the bellow list:

1. CCD Image Sensor:

   Briefly, CCD is two-dimensional rows of pixels which is famous as a modern indicator. Even though recently has been become a replacement instead of photomultiplier tubes, it is still using the photoelectric effect as same as the photomultiplier tube to detect the photons. It is noticeable that in this method of photon detecting is not perfect and contains various noises and symmetric errors which is one of the very basic reasons for students to go through all the data reduction process in order to overcome these featureal errors too. This sensor consists of solid semi-conductor part that each singular position on it, translate each single detected photon [from photoelectric effect] to a charge pulse. This charge has a relative amount with the numbers of detected photons which is captured from our astronomical object. The produced charge goes into the capacitor and again retranslate it as a stored charge, then this stored charge would be read by a computer as our final digital signal. [Figure 1]

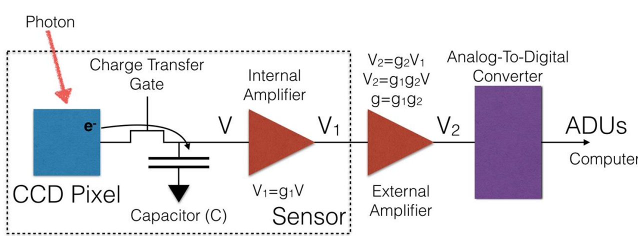


Figure 1: CCD Sensor

2. Python Program:

   Due to analyzing the data, graphing the different data sets and evaluating some statistical components such as Linear square fitting, which has used too match the different data set for this lab, the students used the python program to compile the data and analyze them

3 Data Acquisition

For this lab, 2 different section of data sets had been used.

1. NGC1337 data set
2. 26 Prosperina data set

Both of these data sets which were the CCD Images were provided from DIT (Dunlap Institute Telescope). This telescope is located at New Mexico Skies on Mt. Joy site near May Hill, NM. Also, it is noticeable that it is specially equipped with a 4096×4096 pixel CCD array. So, as a consequence, the resultant field of view is around 36×36 arc minutes. This telescope was designed with a 50-cm robotic telescope to search for optical counterparts of gamma-ray bursts (GRB). Detailed information provided in the appendix. (§B) After collecting the data sets using DIT, the data set provided for students on the website due to upload and use them properly. The format for these provided files is standard astronomical format, known as FITS files. FITS files contain the image data, stored in binary format. For NGC1337 data set since there was only 1 set of; data, flat, bias and dark were provided with no special action needed to upload them in python or taking any statistical method. The only used equation to filtered correctly is as below, which has performed on the data set due to calibrate them.

$$Calibrated\ Data = \frac{(Data - Bias) - (Dark - Bias)}{Flat - Bias} \tag{1}$$

For the 26 Prosperina data set, there was a challenging part. Since the provided data set for each Flat, bias, dark and the actual data were more than one due to upload the file and correctly use them, the students faced some difficulties. As provided on course website[1] according to the below table:

Number of Data set	**Data**	**Bias**	**Dark**	**Flat**
NGC1337	**1**	**1**	**1**	**1**
26 Prosperina Day1	**3**	**10**	**15**	**5**
26 Prosperina Day 2	**3**	**10**	**15**	**5**
26 Prosperina Day 3	**3**	**10**	**15**	**5**
26 Prosperina Day 4	**3**	**20**	**15**	**5**

Table 1: Number of each Data set

For calibrating the Data set using equation (1) and at the same time dealing with different data set, taking Mean or Median for the data set suggested as a solution. This is in our case since we are looking for a trustable value taking Median would provide us with enough accuracy. It is noticeable that the EXPOSURE time for NGC1337 is equal to 240 while 26 Prosperina is 60. Note: In Appendix A there is an example from a part of code in order to check how the Median can be taken and how uploading several files is possible in python language.

After acquiring data sets from the astronomical object, the next step is to get the relevant data from the USNO B1 Catalogue which mentioned before as our reference data to rematch the CCD Images with them. Due to get the data we use the [2] provided link which is the US Navy Observatory catalogue which provides different information for stars such as positions and proper motions. These data sets are based on the scan of photographic plates from Palomar−48 inches Schmidt telescope which took many sky surveys within the past years. Due to run this list into the python program you can check appendix c. It is important to understand that the uploaded data set from the USNO B1 list are in "RA" and "DEC" and has to be converted into Pixel size to be compared with our CCD Images which will be explained more in Data Reduction section.

4 Data Reduction

As a first step for Data reduction, students need to find the centeroid positions for each CCD images. It is obvious that each of this CCD Images contains many irrelevant pixels which has to be exclude from the domain of our interests. In order to find the centeroids I used a mathematical functional way which personally named as a "Couple fencing" method. This method is a simple algebraic algorithm which is applied for each pixel in the Image.

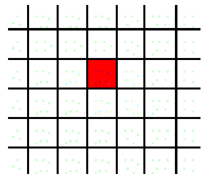


Figure 2: Fencing around a CCD Pixel

As the above figure suppose we have a $n \times n$ pixels. and we are interested to make a fence around a specific one, in order to set the proper value of width fence, we use the Standard normal Distribution method and consider the related data in the "width $= 3\sigma + Mean\ Value$" in order to cover up to 99 percent of our data each time that the relative fence makes. So, by doing that for the entire CCD image we can evaluate all the centroids. Although, there are many exemptions such as 2 couple centroids or unrelated bright pixels which would be excluded while the fences make each time. As a final result by applying this "Couple fencing" we

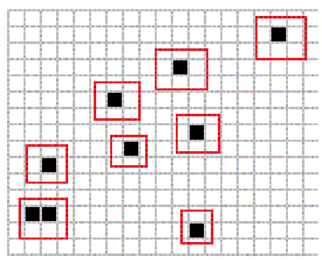


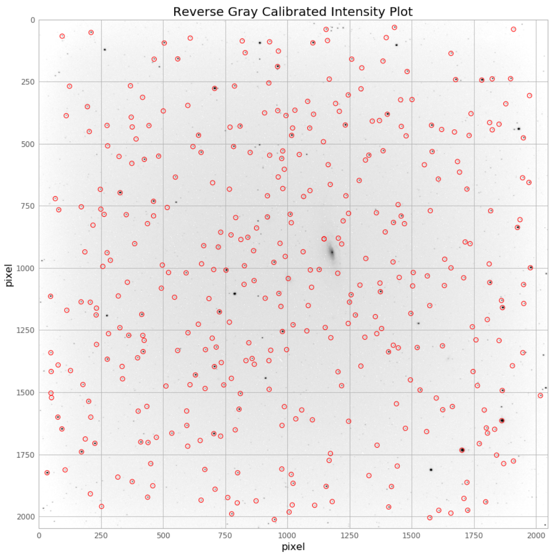
Figure 3: Couple Fencing around a CCD Pixel

can cover up to a well-enough quantity of centeroid data set due to compare with the USNO B1 list. Figure 3 is a visual version of what possibly is happened around each point of interest. As it mentioned before the below table provided the needed width size around each pixel due to make the proper fence. Notice that this width can be extremely critical. By having a Large width scale the function would not work correctly since it is possible to miss match a lot of centeroids and computes them as a one point. By a having a significantly small Width that cannot cover the enough data pixel the relative centeorid would not be detected and you will be face with the lack of enough centeorid due to compare them with USNO B1. Note: in Appendix D there is a cutoff code of this "Couple fencing" provided.

Data for making fence	**NGC2337**	**26 Prosperina 1**	**26 Prosperina 2**	**26 Prosperina 3**	**26 Prosperina 4**
Mean value	**3.74**	**0.0094**	**0.0081**	**0.0096**	**0.00912**
$\pm\sigma$	**0.49**	**0.010**	**0.0078**	**0.011**	**0.021**
$\pm 3\sigma + Mean\ value$	**5.21**	**0.0394**	**0.0315**	**0.0426**	**0.0721**

Table 2: The Width computation for making a fence in each CCD images pixel

After finishing about finding centeroid we would have to compare the evaluated centeroids with the displayed USNO B1 Reference as below:



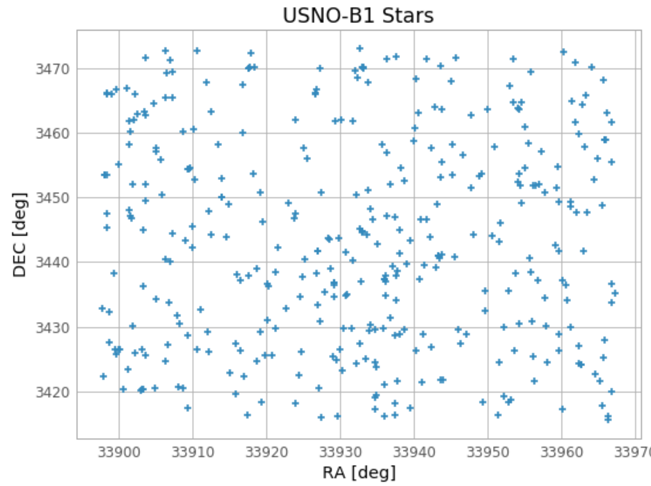


Figure 4: Centeroid location by "Couple fencing"

Figure 5: USNO B1 Catalogue stars

Since the USNO B1 is not in a pixel coordinate we need to use a function to convert the celestial coordinate into the 2D CCD Image. In order to reach this purpose we are using the following equation to find the X and Y pixels:

$$X = -\frac{\cos\delta \sin(\alpha - \alpha_0)}{\cos\delta_0 \cos\delta \cos(\alpha - \alpha_0) + \sin\delta \sin\delta_0}$$
$$Y = -\frac{\sin\delta_0 \cos\delta \cos(\alpha - \alpha_0) - \cos\delta_0 \sin\delta}{\cos\delta_0 \cos\delta \cos(\alpha - \alpha_0) + \sin\delta \sin\delta_0}$$

Notice that, α_0 and δ_0 are obtainable from the FITS le headers. Standard coordinates were then converted into pixel coordinates as a result the data sets now are comparable on the same magnification. One more thing that you have to notice is, the focal length and pixel length must be considered also. As mentioned before, the pixels are binning 2×2 which means the $x_0 and y_0$ values are both equal 1024. Also, in an ideal situation the focal length would be the same number, but since the CCD camera that we are dealing with is not an ideal camera we have to consider different values for each pixel which will be discussed more in the plate solution part. But as an ideal case we consider $f = 3454nm$. So as following we have:

$$x = f(X/p) + x_0 \qquad y = f(Y/p) + y_0 \tag{2}$$

After converting the USN0 B1 data set into Pixel coordinate we plot the result versus the founded centeroids positions as follow:

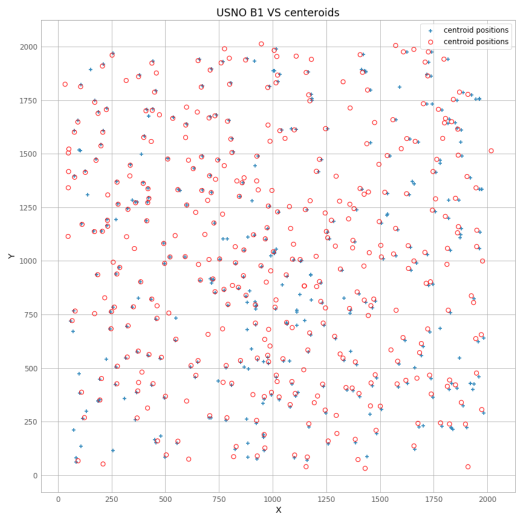


Figure 6: U Vs Cent before reduction

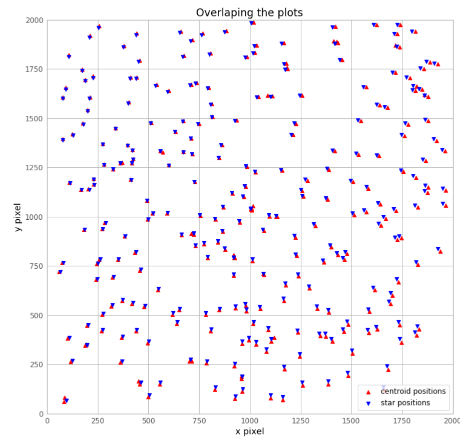


Figure 7: U B1 Vs Cent after reduction

The Blue plus signs represent the stars' location and the red circles represent the founded Centroids, as you can by figure 6, they still not well enough to match. so we simply using the Pythagoras theorem due to finding the closest centroids relative to the nearest stars as follow:

$$Target\ Distance = \sqrt{(|x_{centeorid} - x_{list}|)^2 + (|y_{centeorid} - y_{list}|)^2} \tag{3}$$

By reducing the data according to the method which has provided in equation 3, and set the "Target Distance" no more than "18" pixel unit distance, the reduced data plot as figure 7 would be found which contains a well-enough precise data to build the next step based on. Note: In Appendix E the cutoff Pythagoras method is provided to check.

Based on figure 7, As you can the centeroid position in comparison with star's location still need to be shifted in order to well-matched. By plotting the Residual and observing the Y-offset [figure 8], it is concludable that all the centeroids need to move in a symmetric pattern due to locate at the star's position.

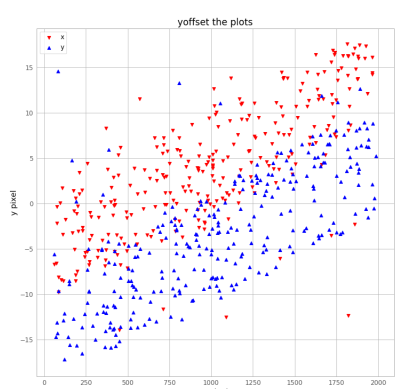


Figure 8: Y-offset plot

Due to overcome and fix the final miss-matched of stars, we are going to use a plate solution method which in a simple definition; is a matrix transformation of all centeroids to the star's position. In order to Transform all the data set we need to rotate them in proper way. As what we have below:

$$x = \frac{f}{p}(X\cos(\theta) - Y\sin(\theta)) + x_0 \tag{4}$$

$$y = \frac{f}{p}(X\cos(\theta) + Y\sin(\theta)) + y_0 \tag{5}$$

The rotation between (x,y) CCD Image and and (X,Y) standard coordinates with θ is calculated as above. As you can see there is a linear transformation between X and x which $X = (X, Y, 1)$ and $x = (x, y, 1)$ So, which is evaluated as below:

$$x = TX \tag{6}$$

$$T = \begin{pmatrix} (\frac{f}{p})a_{11} & (\frac{f}{p})a_{12} & x_0 \\ (\frac{f}{p})a_{21} & (\frac{f}{p})a_{22} & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7}$$

Thus as a result we expect to have a_{11}, a_{12}, a_{21} and a_{22} as the scalars of transformation and x_0 and y_0 as pixel offsets. Also by investigating on this matrix transformation as a result we can also find the effective telescope scale which in an ideal situation must be equal to f/p but since the situation is not ideal also, can evaluate as the determinant of matrix T as below:

$$\frac{f}{p} = \sqrt{Det(T)} \approx 191888.889 \tag{8}$$

So, the number 191888.889 is a good reference to check if the evaluated determinant is approximately acceptable or not. After evaluating the plate constants for each CCD Image we can also evaluate the regression's reduced chi-squared in order to verify its integrity. The equation is as follows:

$$\chi^2_{red} = \frac{1}{\nu}\sum_{i=1}^{N}(\frac{y_i - y(x_i)}{\sigma_i})^2 \tag{9}$$

In tale 3 the final result for plate constants, relative Chi squared and other useful data are provided as below:

	a_{11}	a_{12}	a_{22}	a_{21}	x_0	y_0	${\chi_x}^2$	${\chi_y}^2$	**f/p**
NGC	**-1.4122e-01**	**-3.8356e-03**	**-1.5217e-01**	**-4.1314e-03**	**1051.14**	**1012.11**	**0.27**	**0.02**	**173255.62**
Pro D1	**-2.3452e-01**	**-5.4326e-03**	**-2.6342e-01**	**-5.2681e-03**	**1047.62**	**1065.85**	**0.38**	**0.016**	**187765.25**
Pro D2	**-6.3777e-01**	**-1.29543**	**-6.5328e-01**	**-1.30789**	**1020.82**	**980.43**	**138.66**	**259.87**	**096742.57**
Pro D3	**-1.5321e-01**	**-3.9921e-03**	**-1.7231e-01**	**-4.2014e-03**	**997.56**	**999.34**	**0.43**	**0.13**	**176122.45**
Pro D4	**-1.4391e-01**	**-4.0147e-03**	**-1.6287e-01**	**-4.2153-03**	**1015.58**	**970.76**	**0.52**	**0.17**	**182923.68**

Table 3: Plate constants and the relative evaluations.

Notice that in below table NGC1337 is called as "NGC" and 26 Proserpina is called as "Pro". Also D_i is the same as Day 1.

As a final step, The least-squares solutions help us due to calculate the standard coordinates for any pixel in the CCD. Now for Converting the CCD position into Celestial coordinate we do the following conversion:

$$X = T^{-1}x \tag{10}$$

which can be determined as follow:

$$tan(\alpha - \alpha_0) = \frac{-X}{cos(\delta_0) - Y sin(\alpha_0)} \tag{11}$$

$$sin(\delta_0) = \frac{sin(\delta_0) + Y cos(\delta_0)}{\sqrt{1 + X^2 + Y^2}} \tag{12}$$

So, by using equation 11 and 12 we are able to do the conversion from standard coordinate to Celestial coordinate.

5 Discussion

Proper motion Based on what we have got on previous sections of this report, we can evaluate the proper motion. The steps are rooted in simple physic but as an advanced model. Now, since we evaluated the position of 26 Proserpina in different period, we can consider this motion as simple equation:

$$x = vt \tag{13}$$

$$v = x/t \tag{14}$$

As it mentioned before in the last part of data reduction section, We have to notice to convert the evaluated position to celestial coordinate in order to evaluate the accurate positions. So, by having the relative positions we have to also, notice about the curvature of space which can be evaluated as follow:

$$cos(\delta_0) = sin(a_0)sin(a_1) + cos(a_0)cos(a_1)cos(x_0 - x_1) \tag{15}$$

from the above equation, a_0 is equal to the initial DEC, x_0 represents the initial RA and correspondingly a_1 and x_1 represent the final DEC and final RA. and δ_0 is equal to the arc length in the unit of Radian. Next, by evaluating the position of 26 Proserpina in different days and convert the 2D CCD pixel position into Celestial coordinate we would have a following table:

	X pixel	**Y pixel**	**RA (deg)**	**DEC (deg)**
26 Proserpina D1	**1032.26**	**1030.17**	**82.61**	**25.01**
26 Proserpina D2	**1027.74**	**997.06**	**81.65**	**24.76**
26 Proserpina D3	**1019.26**	**955.46**	**80.47**	**25.96**
26 Proserpina D4	**1544.53**	**1002.24**	**80.34**	**26.87**

Table 4: The relative position of 26 Proserpina in 4 different days

Now, by combing equation 16 and the information from table 4 we can reconstruct the proper motion of asteroid from Day 1 to Day 2, Day 2 t Day 3 and Day 3 to Day 4. We have to notice that by writing Day 1 to Day 2 means, 2 different days and be aware that these days are not necessarily consecutive.

Time **Relative time**	**Proper motion** **arcsec/hour**
Day 1 to Day 2 **[length of 1 days]**	**73.67** $\pm relative\ error$
Day 2 to Day 3 **[length of 2 days]**	**52.81** $\pm relative\ error$
Day 3 to Day 4 **[length of 3 days]**	**27.98** $\pm relative\ error$

Table 5: The final result for proper motion of 26 Proserpina

From table 5, we can see that the proper motion has decreased over the observable time which could be for several astronomical reasons. Notice that in these calculations we supposed that the frame that we are measuring is stable and not in motion which is not true. Since the earth is a moving frame we have to notice that this can be the effect on our entire measurement.

1. Error One of the focus areas on this report would be "Error Correction". Many different factors are contributing to making an error such as atmospheric situation, instrument accuracy and the error made by the observer during the observation. Fortunately, most of these errors can be reduced with different methods. The entire report is one of the effective methods due to reducing the errors but we have to notice in each step we have to consider the relevant sigma and Mean value and the normal distribution which obeys the $s\ sigma + Mean\ value$ and reduce our data set based on that. furthermore, as mentioned before there are some set of symmetric errors that is solvable but is not mentioned in this report. One of these errors is the movement of measuring frame which is possible to be reduced but it was not the focus area of this report.

A

Appendix A: Displaying and Running the Data set by python codes

Here is the cut of python cod for displaying the data using the Python PyFITS library.

```
Data files
fits1='NGC7331-S001-R001-C001-r.fts'
x = fits.getdata(fits1)
hdr = fits.getheader(fits1)
```

The rest of the code is how to run a list of files such as Flat files and then take Median or Mean in order to use them later as follow:

```
d1 = Path('Proserpina/26Proserpina/20120119/Day 1/FLAT')
day1 = os.listdir(d1)
headerd1 = fits.open(d1/ day1[0])
getting data and median for the Proserpina data
for i in range(0,int(len(day1))):
i is the file
zd1 = fits.getdata(d1/ day1[i])
if i == 0:
yd1 = zd1
else:
yd1 = np.dstack([yd1,zd1])
xd1 = yd1.transpose(2,0,1)
FLATDAY1 = np.median(xd1,axis=0)
```

B

Appendix B: Dunlap Institute Telescope information

Site	Mt. Joy, NM
Geographic location	32°54'10"N 105°31'46"W
Primary mirror	50-cm (diameter)
F/ratio-focal length	F/6.8 – 3454 mm
Camera	Apogee Alta U16M (www.ccd.com)
Detector	TE-cooled Kodak KAF-16803 CCD
CCD format	4096 × 4096, 9.0 μm pixels (unbinned)
Nominal pixel scale	0.53 arc seconds/pixel
Field-of-view	36.4 arc minutes (square)
Optical Filters	*g*,*r*,*i*,*z*,*L*,*B*,clear
Limiting Magnitude	$r \approx 17.5/18.5$ mag. (10/60 seconds)

Figure 9: Nominal observatory and instrument properties

C

Appendix C: Relevant code due to running the USNO B1 Cat

This piece of code provides you to upload the USNO B1 Catalogue

```
def usno(radeg,decdeg,fovam):   RA/Dec in decimal degrees/J2000.0 FOV in arc min.
    str1 = 'http://webviz.u-strasbg.fr/viz-bin/asu-tsv/?-source=USNO-B1'
    str2 = '-c.ra=:4.6f-c.dec=:4.6f-c.bm=:4.7f/:4.7f-out.max=unlimited'.format(radeg,decdeg,fovam,fovam)
    final URL: make sure it does not have any spaces or carriage returns/line feeds when copy-pasting
    sr = str1+str2
    f = url.urlopen(sr)
    s = f.read()
    f.close()
    column interpretation compass
    namecol, RAcol, DECcol, rband = 0, 1, 2, 12
    null1, null2 = ' ', ''
    sl = s.splitlines()
    sl = sl[45:-1]  get rid of header
    name = np.array([])
    rad = np.array([])  RA in degrees
    ded = np.array([])  DEC in degrees
    rmag = np.array([])  rmage
    get data from each line
    for k in sl:
    kw = k.decode().split('')
    if kw[0] != '':
    name = np.append(name,kw[namecol])
    rad = np.append(rad,float(kw[RAcol]))
    ded = np.append(ded,float(kw[DECcol]))
    if (kw[rband] != null1) and (kw[rband] != null2):
    rmag = np.append(rmag,float(kw[rband]))
    else:
    rmag = np.append(rmag,np.nan)
    return data
    return name,rad,ded,rmag
    main function
    if name==' main':
    ras= file[0].header['ra']
    des = file[0].header['dec']
    radeg = 15*( float ( ras [ 0 : 2 ] ) + (float ( ras [3:5])/60.) +( float ( ras [6:])/3600.))
    dsgn = np.sign ( float ( des [ 0 : 3 ] ) )
    dedeg = float ( des [ 0 : 3 ] ) + dsgn*float ( des [4:6])/60. + dsgn*float ( des [7:])/3600.
    print 'RA, DEC [ deg ] =', radeg , dedeg
    get stars in a 5'x5' square around (RA, DEC = 30, 30)
    name,rad,ded,rmag = usno(radeg,dedeg,34.55)
    mask=np.where(rmag<15)[0]
    rad=rad[mask]
    ded=ded[mask]
```

D

Appendix D: "Couple Fencing method" relevant codes

This reduced version of code provides due to check the "Couple fencing method"

def Fence(fence, Mean$_v alue, sigma)$:
Border = Mean$_v alue + 3 * sigmacoverupto$
X$_P eak, Y_P eak = Peak_F ence(fence)$
Total$_l ist$ = []
for x in range(len(fence)):
list= 0
for y in range(len(fence)):
if (fence[y][x] > Border):
if (x==0) or (x==len(fence)):
print("Broken fence because of x")
return []
if (y==0) or (y==len(fence)):
print("Broken fence becauswe of y")
return []
else:
list = list + 1 updating the list
else:
Total$_l ist.append(list)$
print("total list=", Total$_l ist$)
width = max(Total$_l ist$)
print("width=", width)
if (width<3):
print("Broken fence / Not enough pixel fitted for width problem")
return []
Render$_F ence = fence[int(Y_P eak - (width/2)$
-1):int(Y$_P eak + (width/2) + 2)$,
int(X$_P eak - (width/2) - 1) : int(X_P eak + (width/2) + 2)]$
if (len(Render$_F ence[0]$) == 0) :
print("Broken fence / render fence is zero shape")
return []
Rad$_R ender_F ence = (len(fence) - width)/2$
if (X$_P eak + Rad_R ender_F ence) > len(fence)$:
return []
if (Y$_P eak + Rad_R ender_F ence) > len(fence)$:
return [] for x in range(len(Render$_F ence[0]$)) :
for y in range(len(Render$_F ence$)) :
if (Render$_F ence[y][x] > Border$) :
if (x==0) or (x==len(Render$_F ence$)) :
print("Broken fence / Over radius")
return []
return Render$_F ence$

E
Appendix E: Pythagoras method codes

Due to reducing the data to the well ones we used the following method:

```
for i in range(len(centx)):
    centym=centy[i]
    centxm=centx[i]
    centym=centy[i]
    for j in range(len(X)):
    Ym=Y[j]
    Xm=X[j]
    Ym=Y[j]
    sepration= np.sqrt(((np.abs(centxm-Xm))**2) +((np.abs(centym-Ym))**2))
    if sepration < 18:
    xstar.append(centxm)
    ystar.append(centym)
    xlist.append(Xm)
    ylist.append(Ym)
    xoffset.append(Xm-centxm)
    yoffset.append(Ym-centym)
```

References

[1] "Python Code for Vizier USNO-B1 catalog, howpublished = http://www.astro.utoronto.ca/ astrolab, note = Accessed: 2019-12-05.".

[2] "Available Instruction By Professor Dae-Sik Moon / Astrometry course website." http://www.astro.utoronto.ca/ astrolab/. Accessed: 2019-12-02.

[3] "USNO B1 Catalogue: 'http://webviz.u-strasbg.fr/viz-bin/asu-tsv/?-source=usno-b1-c.ra=:4.6f-c.dec=:4.6f-c.bm=:4.7f/:4.7f-out.max=unlimited, note = Accessed: 2019-11-05.".

[1] [2] [3]