Scale and Field of View Calculations Using Albireo

Lucas, Miles[1] and Brandon, John[1]
Iowa State University

(Dated: 9 September 2017)

We used images of Albireo with an assumed angular separation of $\Delta\alpha = 35.3''$ to determine the pixel scale and field of view for our SBIG CCD camera and Meade 8" telescope. We accomplished this by subtracting median dark frames from our raw images and using the DAO star-finding algorithm to find the stars' centroids. Our pixel scale was determined to be $1.0072(91)\,''\,\text{pix.}^{-1}$ and our field of view was $770.50(20)'' \times 513.67(13)''$. Analyzing our images further there was a difference in the mean and median brightness between 0.1 s, 0.5 s, and 1.0 s exposure times but very little difference in brightness between photometric V, B, and I filters.

I. INTRODUCTION

This lab seeks to quantify the pixel scale and field of view of the images taken using the SBIG CCD camera and Meade 8" telescope at the Zaffarano Hall observation deck. To do t his we will take images of sources with a known angular separation. The source we used was Albireo (β Cygni), a binary system in the northern sky. Albireo's coordinates are ($19^h30^m43.286^s$, $27°57'34.84''$) with angular separation between the two stars $\Delta\alpha = 35.3''$.

To determine the pixel scale from an image, we must find the distance between the two sources. If we have two centroids and we allow $\vec{r}$ to be their separation, then the pixel scale will be

$$pixel\ scale = \frac{\Delta\alpha}{||\vec{r}||} \tag{1}$$

and the field of view in one direction will be

$$FOV = N\frac{\Delta\alpha}{||\vec{r}||} \tag{2}$$

where N is the number of pixels in the given direction.

We will also compare the images from each filter and exposure time using their statistical moments to analyze the differences in brightness.

II. DATA ACQUISITION AND SETUP

Observations were made on at the Zaffarano Hall observation deck in Ames, Iowa. The night was very clear and the ambient temperature was around 32 °C. Observations were made using a Meade 8" reflector telescope with an SBIG ST-402ME CCD camera with internal V, B, and I filters.

To setup the telescope we used the Meade's GPS functionality to automatically calibrate its position. This calibration involved using two point sources as guides and orienting the telescope centered on the point source. For these measurements, the calibration sources were Arcturus and Altair. We used a 32 mm eyepiece for our calibration alignment. Using the Meade selector, we navigated to the β Cygni system We then switched to a 9 mm eyepiece for fine-alignment on our target. Once we were well centered on our target we removed the eyepiece and inserted our CCD camera.

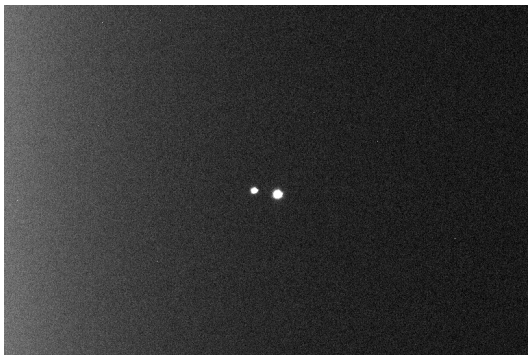
Using CCDops we selected the V filter and began focusing the CCD camera with a manual knob on the telescope. The exposure time for these focusing grabs was 0.05 s. We based our focus on maximizing the peak amplitude read out in CCDops. We took multiple images at multiple exposures with different filters, all recorded in Table III. For each image grab, we set up an autograb using CCDops and took the frames. At each exposure level, we also took dark frames using autograb in order to reduce systematic error.

III. DATA ANALYSIS

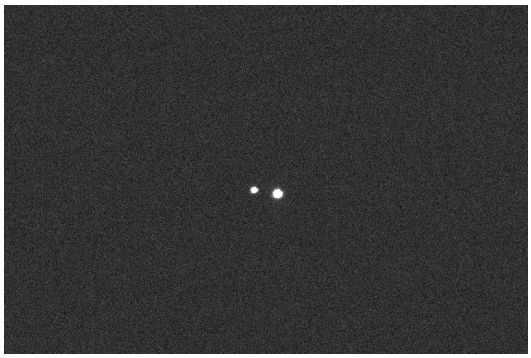
Our analysis included some simple image processing and then used machine learning to determine the pixel scale and field of view.

Our processing pipeline involved finding median dark frames for each exposure length and then subtracting the dark frames from each science image. We use median dark frames rather than mean because we wanted to avoid outliers in the form of cosmic rays or dark pulses. To create the median images, we used a python script that utilizes numpy's array operations for getting the median of the data arrays from each FITS file. Once we saved the three median dark frames (0.1 s, 0.5 s, 1.0 s) we used AstroImageJ to automate the subtraction for the 26 science images. A comparison of the science images before and after subtraction is shown in Figure 1. As you can see, the thermal excitation caused by the camera's electronics is removed.

To calculate the pixel scale of our image, we need to find the center of each star. To do this, we turned to a machine-learning algorithm that uses Gaussian mixture models for detecting stars. The DAO algorithm was used as part of a python package photutils. For more information on this algorithm, see Stetson[1]. The centroid of each star is reported and was then used to calculate separation and the subsequent measurements of interest. The script is shown in Listing B. An example of the centroids calculated using the DAO algorithm is shown in Figure 2.



(a) Raw science image



(b) Processed science image

FIG. 1: Comparison of the raw and processed science images of Albireo in photometric V at 0.1 s exposure.

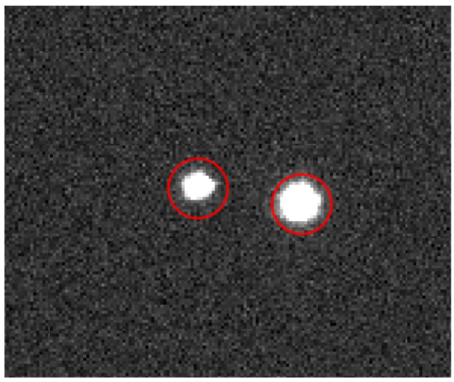


FIG. 2: Results of the DAO algorithm on one of the photometric V 0.1 s exposure

IV. RESULTS

The results of the script are shown in Table I. The mean, median, and standard deviation for each photometric filter and exposure time are listed in Table II.

TABLE I: Pixel scale and field of view values from the 10 photometric V 0.1 s exposures

Variable	Value
Pixel Scale	$1.0072(91)\,''\,\mathrm{pix.}^{-1}$
FOV x	$770.50(20)''$
FOV y	$513.67(13)''$

TABLE II: The statistics for each photometric filter at each exposure level

Filter	Exposure	Max	Mean	Median
V	0.1 s	19800	1.84	1.00
V	0.5 s	64500	7.56	3.00
V	1.0 s	64500	13.7	6.00
B	0.1 s	23100	1.95	1.00
I	0.1 s	23100	1.95	1.00

V. CONCLUSIONS

The results in Table I show that our CCD camera is able to capture almost exactly one arcsecond per pixel, which gives a field of view of $771'' \times 514''$.

Looking through the different processed science images do not show a huge difference between the three photometric filters and the three different exposure times. However, the statistics shown in Table II show that the longer exposures are brighter, but that there is not much difference between the three filters. We also noted that the 0.5 s and 1.0 s exposures were overexposed as shown by the maximum values corresponding with the maximum values of the CCD.

ACKNOWLEDGMENTS

Thank you to Dr. Charles Kerton and Brandon Marshall for their guidance and assistance in this work.

REFERENCES

[1] P. B. Stetson, "DAOPHOT - A computer program for crowded-field stellar photometry," **99**, 191–222 (1987).

Appendix A: Observation Log

TABLE III: Observed 30 August 2017 by Miles Lucas and John Brandon

Time	File	N Frames	Object	Filter	Exposure	Camera Temp.	Notes
20:52	Albireo_V_tenth_	10	Albireo (β Cyg)	V	0.1 s	5 °C	
20:56	Albireo_V_tenth_dark_	5	Albireo (β Cyg)	V	0.1 s	5 °C	Dark frames
20:59	Albireo_V_half_	3	Albireo (β Cyg)	V	0.5 s	5 °C	
21:01	Albireo_V_half_dark	5	Albireo (β Cyg)	V	0.5 s	5 °C	Dark frames
21:03	Albireo_V_full_	3	Albireo (β Cyg)	V	1.0 s	5 °C	
21:04	Albireo_V_full_dark_	5	Albireo (β Cyg)	V	1.0 s	5 °C	Dark frames
21:05	Albireo_B_tenth_	5	Albireo (β Cyg)	B	0.1 s	5 °C	
21:06	Albireo_I_tenth_	5	Albireo (β Cyg)	I	0.1 s	5 °C	

Appendix B: Analysis Scripts

../src/measurements.py

```
from glob import glob
import argparse

import numpy as np
from astropy.io import fits
from astropy.stats import sigma_clipped_stats
from photutils import DAOStarFinder


def get_sep(img):
    mean, median, std = sigma_clipped_stats(img, iters=5)
    daofind = DAOStarFinder(fwhm=3.0, threshold=5. * std)
    sources = daofind(img)
    positions = np.array([sources['xcentroid'], sources['ycentroid']]).T
    return np.linalg.norm(positions[0] - positions[1])

def get_pscale(seps, dalpha):
    pscales = dalpha / np.array(seps)
    pscale = np.mean(pscales)
    pscale_std = np.std(pscales)
    return pscale, pscale_std

def get_fov(pscale, std, dalpha, leng):
    fov = leng * pscale
    std = leng * pscale**2 / dalpha * std
    return fov, std

if __name__ == '__main__':
    parser = argparse.ArgumentParser('Gets measurements from fits files')
    parser.add_argument('filename', help='files to process')
    args = parser.parse_args()
    filenames = glob(args.filename)
    imgs = [fits.getdata(files) for files in filenames]
    seps = [get_sep(img) for img in imgs]
    dalpha = 35.3
    pscale = get_pscale(seps, dalpha)
    x_len, y_len = imgs[0].T.shape
    fovx = get_fov(*pscale, dalpha, x_len)
    fovy = get_fov(*pscale, dalpha, y_len)
```

```
print(pscale)
print(fovx)
print(fovy)
```