AUSAT

Delphini-1: Software And Satellite Operations

Astronomy with a CUBESAT

Andreas Kjaer Didriksen (201303627)
Nicholas Emborg Jannsen (201205962)

Department of Physics and Astronomy
Aarhus University
14. February, 2018

1 Introduction

Here we present how astronomy can be carried out with Delphini-1. Andreas present a short description of the considerations behind any current and future target list for Delphini-1 regarding astronomy, and Nicholas present how aperture photometry can be carried out for highly distorted stellar images. In the last section of each chapter we point out areas of possible future work to improve astronomy done with Delphini-1.

2 Targets – Andreas

2.1 Exposure Time

Since the ADCS of Delphini-1 does not alow us to change orientation and maintain it on a short timescale, we can assume that the direction of the Nanocam will be affected by some drift. This will cause any stars observed by Delphini-1 to have a trail on the camera. This startrail is still usable for photometry if the image contains the full startrail, ie. the trail cannot extend the CMOS sensor. The trailling of the observed stars puts a limit on the maximum exposure time. First a pixel-scale (p_{scale}) of Delphini-1 is calculated, this gives the amount of arcsec that each pixel covers. To calculate this the pixel size (p_{size}) and focal length(f_{length}), both found in the data sheet for the NanoCam, are used. along with a conversion factor from μm to mm and radian to degree. $K = 206.265$.

$$p_{scale} = \frac{p_{size}}{f_{length}} \cdot K \tag{1}$$

To calculate a meaningfull maximum exposure time, the velocity of the field, which the nanocam covers, can be calculated in pixel pr. second, assuming that the Nanocam is constantly pointing radially away from the Earth.

$$v_{cam} = \frac{\frac{360^\circ \cdot 3600\frac{''}{\circ}}{2\pi(R_\oplus + H_{ISS})} \cdot v_{ISS}}{p_{scale}} \tag{2}$$

with $R_\oplus$, H_{ISS} and v_{ISS} being the radius of Earth, orbit height of ISS and the velocity of ISS [1]. Since the size of the camera is known (2048 by 1536 pixels)[2], the time the Nanocam covers an area of the sky can then be calculated

$$\frac{1536p}{v_{cam}} \approx 125s \tag{3}$$

Since the full startrail must be within the CMOS sensor the recommended maximum exposure time will be half this value.

$$t_{exp} \approx 60s \tag{4}$$

2.2 Observable Magnitudes

Being a CMOS sensor, the Nanocam will be less sensitive than a regular CCD, and with the limited exposure time due to the drift, these effects will be causing some limitations for the magnitude which are observable with Delphini-1. Adding that the CMOS sensor is not being cooled acording to the datasheet[4], which introduces a lot of noise on the images due to thermally excited electrons, the magnitude limit for this proposed target list is set to $Vmag = 6$.

2.3 Coordinats

continuing with the assumption that Nanocam is pointing radially away from earth, and adding that delphini-1 will be in an ISS orbit, the declination of the ISS orbit can be used as a constraint on the coordinats for possible targets. The inclination of the ISS orbit is 51.6°.

2.4 Target List

using the criteria described in 2.1 – 2.3 a target query can be made from Simbad. In http://simbad.u-strasbg.fr/simbad/sim-fsam, the following query is written, which lists all stars with a declination between ± 51.6° and a V-magnitude brighter than 6.

> otype = star & dec < 51.6 & dec > -51.6 & Vmag < 6

This give a target list containing ≈ 4100 targets. If the startype is specified to be variable, the search query is written as

> otype = V* & dec < 51.6 & dec > -51.6 & Vmag < 6.

and results in a list of ≈ 1300 targets.

2.4.1 Possible Targets

working from the target list with variable stars describes in section 2.4. A short list with some potential targets is listed in table 1, with emphasis on listing stars which are not situated in crowded areas. This is done in order to assure the best analysis of the images taken.

2.5 Future Work

- Change magnitude range according to experiments with extra camera.
- Create a target query from terminal or python, maybe by using astroquery

Table 1: Target candidates.

Name	Ra	Dec	V_{mag}	type
* alf CMa	06 : 45 : 08.91728	−16 : 42 : 58.0171	−1.46	Binary
* alf Boo	14 : 15 : 39.67207	+19 : 10 : 56.6730	−0.05	Red Giant
* alf Lyr	18 : 36 : 56.33635	+38 : 47 : 01.2802	+0.03	Delta Scuti
* bet Ori	05 : 14 : 32.27210	−08 : 12 : 05.8981	+0.13	Blue Supergiant
* alf Ori	05 : 55 : 10.30536	+07 : 24 : 25.4304	+0.42	Red Supergiant
* alf Vir	13 : 25 : 11.57937	−11 : 09 : 40.7501	+0.97	Beta Cep
* alf PsA	22 : 57 : 39.04625	−29 : 37 : 20.0533	+1.16	Double or multiple
* alf Cyg	20 : 41 : 25.91514	+45 : 16 : 49.2197	+1.25	Evolved Supergiant
* bet Aur	05 : 59 : 31.72293	+44 : 56 : 50.7573	+1.90	Eclipsing Binary
* bet Cet	00 : 43 : 35.37090	−17 : 59 : 11.7827	+2.01	Variable star

- Reading input catalog in `delsimi` that our collaborative Jonas have made [3].

3 Data Analysis – Nicholas

3.1 Class Overview

The final part of this project is to perform image reduction and photometry. This is done with the Python Class called `Delphini_Photometry`

```
class Delphini_Photometry(object):
    def __init__(self, path, LF_name, plot
        , save):
```

The class takes the following inputs: directory path (`path`), names of the light frames (`LF_name`), if the data should be plotted (`plot`), and if the data should be saved to a image-file (`save`). From the path and image names all relevant images are loaded into Python. An example of how to use the software will be illustrated in section 3.4 and all code can be found at Github [5].

3.2 Image Reduction

The image reduction python utility looks as following

```
def image_reduction(self, FF_name, BF_name
    , DF_name):
```

where `FF_name`, `BF_name`, and *DF_name* is the name of the flat, bias, and flat images, respectively. Just before lunch 50-100 flat, bias, and dark images should be saved for optimal image reduction. The light frames (`LF`) is a global variables. The image correction is done with

$$\mathrm{CF} = \frac{S_{\mathrm{raw}} - B_{\mathrm{M}} - D_{\mathrm{M}}}{F_{\mathrm{M}}}, \tag{5}$$

where CF is the corrected frames, S_{raw} is the raw science frames, B_{M} is master bias frame (median combined from all bias frames), D_{M} is the master flat frame (the dark frame with the master bias subtracted – a so-called dark current image), and F_{M} is the master flat (also dark current subtracted and normalized flat image). To do a proper calibration all image settings (CCD temperature, binning, and focus) needs to be the same for all images. Although this image reduction routine seems to perform and make good result, it should be carefully compared to the results from other softwares (e.g. IRAF).

3.3 Photometry

The photometry utility can be seen below

```
def aperture_photometry(self, x_stars,
    y_stars, aperture, background):
```

Given the x and y coordinates to first the target star and then the reference stars in a list, the photometry utility perform aperture photometry with either a *elliptical* aperture or a aperture called *trace* which traces the Center Of Flux (COF) for startrails. By choice the stellar flux can either be corrected by the local sky background flux close to the star, or by a global sky background flux summed from pixel values all around the image.

3.4 Code Results

To explain the software in more detail we here present the results from it. The following code assumes that all data is placed in the same folder, and the science, flat, bias, and dark frames are likewise called so.

```
from Delphini_Photometry import
    Delphini_Photometry
path = '~/path/to/data/'
# Call class
XX = Delphini_Photometry(path, 'science',
    1, 1)

# Image Reduction:
XX.image_reduction('flat', 'bias', 'dark')

# Aperture photometry:
# Ellipse:
x_coor = [146, 201,  87, 213]
y_coor = [ 97,  52, 171, 208]
```

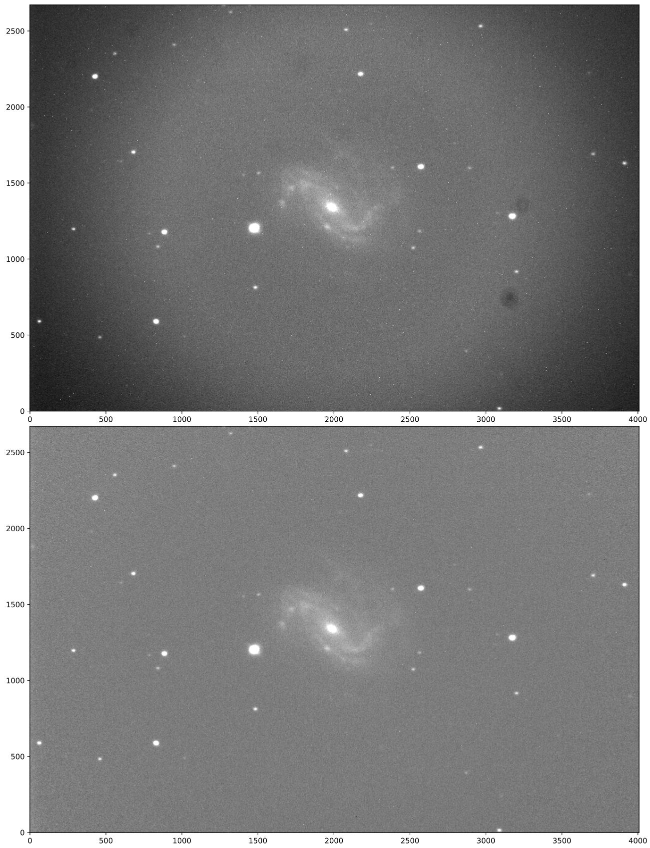

Figure 1: Test of image reduction software on images of NGC 4051 observed from the Ole Rømer Observatory (ORO). Top: The raw pixel data. Bottom: the reduced image.

```
XX.aperture_photometry(x_coor, y_coor,
    ['ellipse', 6, 48, 8, 172], 'local')
# Trace:
x_coor = [107,  165,  49, 176]
y_coor = [103,   55, 176, 212]
XX.aperture_photometry(x_coor, y_coor,
    ['trace',   3, 78, 8, 172], 'local')
```

First a result of the image reduction software can be seen in figure 1. The top and bottom image shows the raw pixel data and the corrected image of the galaxy NGC 4051, respectively.

For the photometry software the first 2 entries in `aperture_photometry` are the stellar coordinates. The next entry is the `aperture` entry that takes 5 arguments: [`aperture`, `a`, `b`, `q`, `phi`]. Here `aperture` is either 'ellipse' or 'trace' corresponding to the two apertures. Because we are working with startrails, `phi` is a tilt angle of the aperture between 0-180 degrees defined by the zero-point of the unit circle (hence counter clockwise from first quadrant).

Using the ellipse as aperture `a` is the semi-minor axis of the ellipse, `b` is semi-major axis of the ellipse, `q` is the width of the local background flux from the ellipse. Figure 2a shows an highly distorted image of M42 (the Orion Nebula) and the red mark indicates the center of a target star. Figure 2b shows the result

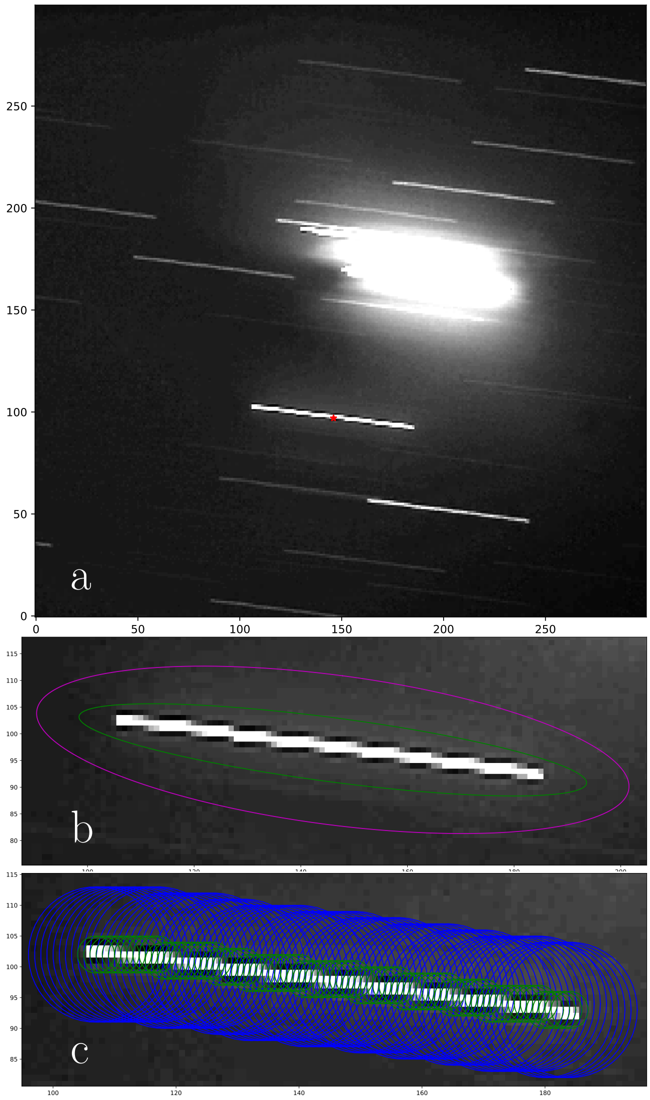

Figure 2: Using the aperture photometry pipline on real data. a) full distorted image showing the Orion Nebula and startrails found at [6], b) photometry using a elliptical aperture and c) photometry using the trace aperture.

of the code example using a elliptical aperture on the target star. The green ellipse is the stellar aperture, and the area between the purple and green aperture is the local sky background flux.

As mentioned the trace aperture is a mission-specific aperture that uses a circular aperture of radius `a`. Given the coordinates of the most left part of the startrail, the COF is found inside this initial circular aperture. Next the circular aperture will be moved one pixel at a time in either the positive x or y direction depending on ϕ

$$\text{step direction} = \begin{cases} x & \text{if } 0 \leq \phi < 45 \text{ or } 135 \leq \phi \leq 180 \\ y & \text{if } 45 \leq \phi < 135 \end{cases}$$

For each pixel step in the x,y direction the opposite y,x pixel coordinate is determined by the COF. Figure 2c shows the result of the code example using the trace aperture on the target star. From our code example the aperture is moved in a x pixel step direction, which means for each step the belonging y coordinate is determine by the COF from the total circular aperture. Seen from figure 2c common center for each green or blue circle represent directly the COF for each pixel step. Just as for elliptical aperture, `q` is here the width of the sky background aperture. The advantage of the trace aperture is, if it turns out that the satellite is very unstable, as long as the Signal to Noise Ratio (SNR) is sufficiently high, this routine will still follow the perhaps strange pattern of the COF for the stars. As all the stellar objects inside one frame will then automatically have the same appearance.

The third argument for the utility `image_reduction` is if a local or global sky background flux should be used to correct the stellar flux. As mentioned above the local sky background flux is define by a band of width `q` around the stellar aperture. As the factor of stellar contamination and crowding is very hard to predict for our mission, the sky background flux can also be determined globally. This is done simply by slicing the image into s number of subframes. Inside each subframe n number of pixels having the lowest flux is found, hence, $s \times n$ is the total number of sky background pixels and the robust $3 \times \text{median(sky-pixels)} - 2 \times \text{mean(sky-pixels)}$ value of the $s \times n$ number of pixels with lowest flux is then the sky background flux. When a high level of vignetting or other image artifacts is present the local sky background flux should be used. Also, the global sky background routine do not work for all subframes at the moment.

All in all the aperture photometry pipline returns stellar flux and SNR for all stars that assigned a coordinate to it. For several code runs with different apertures, it is evident that the trace aperture provides a much higher SNR compared to the elliptic aperture. This is at least very intuitive because the trace aperture is much better to only in-close the star.

3.5 Future Work

- The image reduction routine may be used as long as no effects of vignetting or other image artifacts are visible in our data. However, image reduction needs to be done very carefully, hence, we recommend using IRAF or another software as well to calibrate the data.
- The tilt angle θ could in principle be determined by image analysis. This is a future project.
- From previous studies of mine, I have found that the highest SNR is obtained if the stellar aperture is slightly smaller than whats seems to inclose the whole star. However, the star will contaminate the sky background flux if a too small stellar aperture is choosen. Thus, an easy improvement of the software will be to determine local flux further away from the star.
- First the bug where the global background flux only takes certain subframes s as input needs to be fixed. Moreover, having an crowded stellar image and using a lot of subframe in the global background subtraction, a stellar pixel could in principle be misinterpret as a sky background pixel. A simple threshold value could reinsure that this would not happen.

Bibliography

[1] ESA. Iss: International space station. http://www.esa.int/Our_Activities/Human_Spaceflight/International_Space_Station/ISS_International_Space_Station, 2014. Accessed: 2018-02-13.

[2] GOMSpace. Nanocam c1u - datesheet. https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanocam-c1u-17.pdf, 2017. Accessed: 2018-02-13.

[3] Jonas Svenstrup Hansen. Delsimi - a a simulation code for astronomy images with delphini-1. http://sweaglesw.com/cs448/, 2018. Accessed: 2018-02-13.

[4] Micron Technology. Delphini-1 sensor description. https://www.1stvision.com/cameras/sensor_specs/Aptina-MT9T031_3100_PB.pdf, 2006. Accessed: 2018-02-06.

[5] Nicholas Emborg Jannsen. Delphini photometry: A aperture photometry pipline for astronomy image with delphini-1. https://github.com/jonasshansen/delsimi, 2018. Accessed: 2018-02-13.

[6] Woodley Packard and Steven Stanek. Removing motion blur from astrophotographic images. http://sweaglesw.com/cs448/, 2014. Accessed: 2018-02-13.