Delsimi: A Simulation Code for Delphini-1 Astronomy Images

Jonas Svenstrup Hansen
201205674@post.au.dk
Study no. 201205674

Department of Physics and Astronomy
Aarhus University, Denmark

February 13, 2018

1 Introduction

This protocol forms the documentation of the author's work completed during the course *Delphini-1, Software and Satellite Operations* which the author followed as an extra-curricular project. Delphini-1 is the first satellite from Aarhus University. Its purpose is a proof-of-concept, but remote sensing and, hopefully, astronomy are project goals.

Delsimi is designed to generate astronomy images with which to evaluate the performance of various photometric methods and thereby address the capabilities of Delphini-1 as a space telescope. This work describes the contemplations behind the crucial elements of the simulation.

1.1 Source Code

The Delsimi source code, written in Python 3.5, is available at the online repository `https://github.com/jonasshansen/delsimi`. Here a README file explains the basic structure of the code and how to run it. A requirements file is included in the repository with a list of packages required for running Delsimi. These packages can e.g. be installed in a virtual environment.

2 Simulation

The main idea of Delsimi is to add pixel-integrated, smeared PSFs of stars at given positions and of given Bayer filtered RGB flux to an image that is subsequently binned. The output is a FITS file containing the image and with a WCS solution in its header.

Most of the methods implemented in the Delsimi code are inspired by the TASOC photometry pipeline (TASOC, 2018, source code) for the TESS satellite which the author is doing a master's thesis on. The TASOC photometry code is currently in development, but includes a FITS image simulator as well as PSF fitting. The author's development of these parts forms the basis for the ideas in Delsimi.

2.1 PSF, Smear and Pixel Integration

The point spread function (PSF) of Delphini-1 is unknown. As a crude approximation, a two-dimensional Gaussian is used for Delsimi. The Gaussian PSF is sampled at subpixel resolution and interpolated. Smear due to the lack of pointing on Delphini-1 is introduced by convolving the subpixel PSF with a line kernel of the same dimensions. The line end and starting points are limited to integer positions in the subpixel resolution due to the package (scikit-image, `skimage.draw.line_aa`) used, but this error is reduced with oversampling.

The subpixel, smeared and normalized PSF is then interpolated and integrated at the desired subpixel location and multiplied by the stellar flux and integration time to yield the final smeared pixel response function (PRF) of a star. The Bayer filter is simulated here by scaling the pixel values depending on the color of the star and the position on the filter. Integration of the high resolution PSF is only done if an evaluation of the interpolation yields a significant value, chosen to be 1×10^9. This is done to avoid the scaling of inaccurate parts of the interpolation that do not represent the star. A sketch of this method is shown in fig. 1.

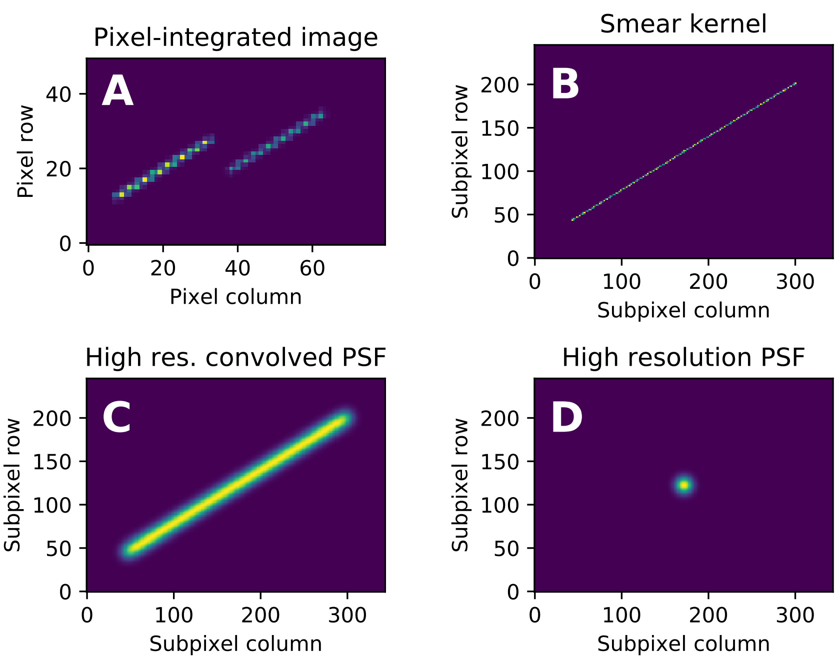

Figure 1: *The pixel-integrated image (A) is created by a convolution (C) of the smear kernel (B) and a high resolution PSF (D) that is interpolated and integrated, shown here without binning.*

The basis of this structure, apart from the smear convolution, is inspired by the TASOC photometry pipeline (TASOC, 2018, source code). It allows for the easy inclusion of a more accurate PSF in the future if this should be measured.

The drawback of the smear convolution method is, however, its very long execution time, especially for long star trails. Therefore, an alternative method is proposed as a future extension: The smeared PSF could also be created, albeit with some errors, using an extensive number of evaluations of an analytical pixel-integrated Gaussian, the method `integratedGaussian` in `psf.py`, along the line of the star trail. The accuracy and efficiency of this method remains to be shown.

2.2 Magnitudes

A realistic simulation of the stellar fluxes observed with Delphini-1 should take account of the Bayer RGB filter on the Aptina MT9T031 CMOS sensor (Micron Technology, 2006) of the NanoCam C1U. While summation binning of the 2×2 pixel Bayer filter structure yields greyscale images, the observed flux in each of the four RGB filtered pixels will be different for a non-uniformly colored source. Dependent on the extent of the PSF and the position of the star, this will most likely influence the flux levels in the binned pixels.

2.2.1 Flux and Filters

In order to accurately simulate the stellar flux counts in Delphini-1 astronomy images an absolute flux level is needed. This does, however, require a measurement of the flux levels of stars observed with the Delphini-1 camera. An easily correctable alternative using a is therefore applied in Delsimi.

For accurate star simulation on the Bayer filtered CMOS sensor, a transformation from the often available Johnson-Cousins UBVRI filters to RGB has to be applied when using catalog stars with color information in these filters. An attempt at using an inversion of the results of Park et al. (2016, eqs. 1-3) yielded very unrealistic results and was discarded. The code is preserved in the `utilities.py` file function `bvr2rgb_discarded`.

As a simple alternative, the current filter transformation assumes a direct translation from the Johnson-Cousins BVR to RGB: B→B, V→G and R→R. While this transformation may not be very accurate, it makes sure that the color effect is implemented. Furthermore, it is easily replaceable with a more sophisticated method, being placed as a function `rvb2rgb` in the `utilities.py` file.

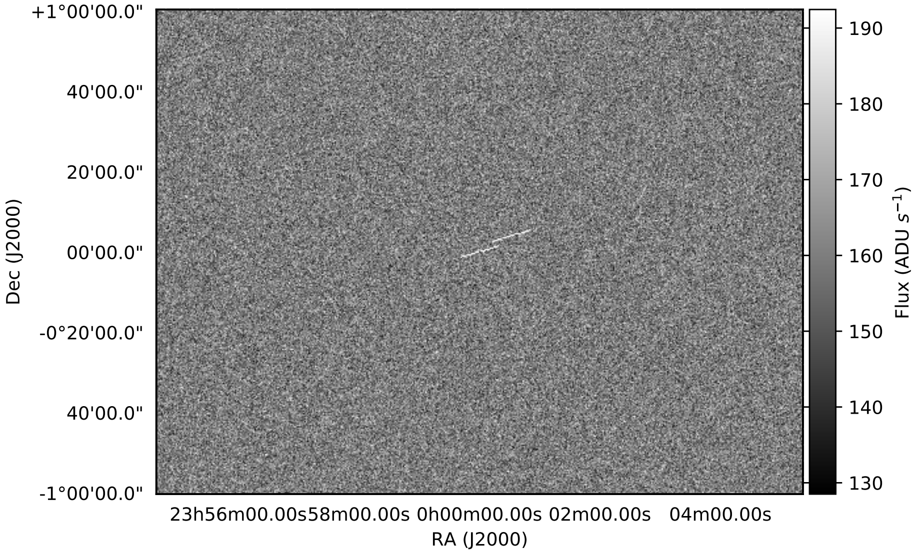

Figure 2: *The simulated image of two fake stars of BVR magnitudes (*6.0, 5.0, 4.0*) and (*4.5, 5.5, 6.5*) integrated for* 10 s *with a movement angle of* 0.3 rad *and speed inferred from approximate ISS orbital parameters, generated using the call* `python run_delsimi.py -t 10 -v 0.3`.

2.3 Noise

An accurate account of the noise in Delphini-1 images is difficult. The only absolutely known noise sources are read noise and dark current, listed in the data sheet from Micron Technology (2006). These two approximate levels are implemented in Delsimi by adding them to the smeared image. Thus, the noise is not smeared. Other noise sources have not been implemented as of yet.

3 Future Extensions

Possible extensions to the Delsimi code are many. Especially relevant is the inclusion of catalog input to the code. A pixel-integrated Gaussian at high time resolution as a replacement of convolution might improve performance. Combined with catalog input this would make Delsimi relevant for target evaluation, since one could make images to simulate whatever target area of the sky is desired. A programmical online catalog search in a given area, perhaps with limits found using the Delphini-1 orbit, would be a relevant implementation of this.

Measurements of the Delphini-1 PSF could provide a very accurate estimation of this. A more easily obtainable solution, albeit with low accuracy, would be PSF sampling from ground-based images with an identical camera. Such an image could also help constrain the magnitude to flux conversion parameter, which is currently a value found using a scaling relation from aperture photometry for the TESS satellite. An image of known stars of different color could also aid the development of a more accurate filter transformation.

Background, focus change, jitter, more accurate noise, readout smear, rolling shutter, saturated pixels and cosmic rays are other less easily estimated possible extensions to Delsimi.

Everyone are welcome to contribute to the Delsimi code. This is most efficiently done using Git by pushing to the development branch of the repository `https://github.com/jonasshansen/delsimi`.

Bibliography

Micron Technology. 2006, Delphini-1 Sensor Description, `https://www.1stvision.com/cameras/sensor_specs/Aptina-MT9T031_3100_PB.pdf`, accessed: 2018-02-06

Park, W., Pak, S., Shim, H., Le, H. A. N., Im, M., Chang, S., & Yu, J. 2016, Advances in Space Research, 57, 509 [ADS]

TASOC. 2018, The basic photometry setup for TASOC, `https://github.com/tasoc/photometry/tree/psf`, accessed: 2018-02-13