École Polytechnique Fédérale de Lausanne

Rotation estimation of detected satellites

by Noah Kaltenrieder (301368)

Bachelor Thesis

Mathieu Salzmann
Thesis Supervisor

June 2021

Acknowledgments

I really would like to thank my supervisor Prof. Mathieu Salzmann who helped me a lot and all the people that participated in a way to this project, namely Dr. Cameron Lemon, Prof. Jean-Paul Kneib, Prof. Frédéric Courbin, and the SSA Team. I also want to express my gratitude to Yann Bouquet who helped me with pleasure when I had questions. Finally, I also would like to especially thank Alexandre Di Piazza and Marcellin Feasson with whom I have collaborated through all the semester in different ways.
Thanks a lot to Prof. Mathias Payer for the thesis template that I used to write this report [1]

Lausanne, June 2021 Noah Kaltenrieder (301368)

Abstract

I present an approach to extract the intensity along the satellite streaks that can be found in images that were obtained by OMEGACAM on the VLT Survey Telescope. In this project, I only analyzed the long tracks that were created by high velocity objects, to try to get an estimation about their rotation. I used Fourier transform to get the periodicity of the intensity along the tracks. Due to the lack of real data about rotation, I first had to create fake streaks with sinusoid intensity and different angles. I manually get the tracks to run my code on them, so ideally this project should be integrated with DetectSat [2] so that all the process can be done automatically. I found very great result with it and run the pipeline on real tracks. By making assumptions about the satellite, like their altitude, I get possible results, but since we don't have access to the real values, we can't say if it is close to reality or not.

Contents

Chapter 1

Introduction

This project is about estimating the rotation of spatial objects detected by the OMEGA-CAM on the VLT Survey Telescope. In these times, we have a lot of flying objects, it can be debris, useful satellites or dead satellites, around the Earth and it becomes more and more necessary to have a clear view of all these objects. The project of Yann Bouquet "DetectSat"[2] was focused on detection of the satellite tracks on fits files, but with both, small (low velocity) and long (high velocity) streaks. To this project I had to focus on the long tracks, so that the variation of intensity along the line could be better detected. The rotation estimation can be used to determine which kind of object the streak is and, if it is a satellite, it may be used to better find out which one it is. Ideally, this should be run on the lines detected by DetectSat so all the process could be done with only the fits file as input.

The main challenge is that we don't have access to data about rotation of satellites, so I had to first create my own streaks with sinusoid intensity. This lack of resources is very restrictive, since even if I get a result, we cannot know the precision of this method and we cannot be sure that it is right or completely wrong. This is why I had to begin with creating lines that seem realistic and try with random angles.

The second part of this project was to retrieve the periodicity of the sinusoidal function along the streaks that I generated, and finally try this on real tracks. Since my project does not recognize the lines, I had to find the coordinates of the streaks by hand and give them as input to the pipeline. For this part, there are many factors that can

decrease the precision of the pipeline, e.g., if the line crosses a really bright star, it gives a peak of intensity and can lead to wrong result, or if the brightness of the streak is too small compared to the brightness of the background of the sky on the image.

Chapter 2

Inputs

In this project, I used fits images that were generated by the OMEGACAM on the VLT Survey Telescope, and I used a script to create a mosaic of 32 images, separated by NaN values, that are 4000x2000 pixels. So the image with all the mosaic's dimension is 16'000x16'000 pixels. A mosaic is approximately 1.5-2 GB of data.
I principally used the fits file "*OMEGA.2020-02-22T02:11:36.295_fullfield.fits*", corresponding to the captured image February 22, 2020, at 02:11:36, and with the pixels scaled using ZScale to be able to see the streaks as wanted.

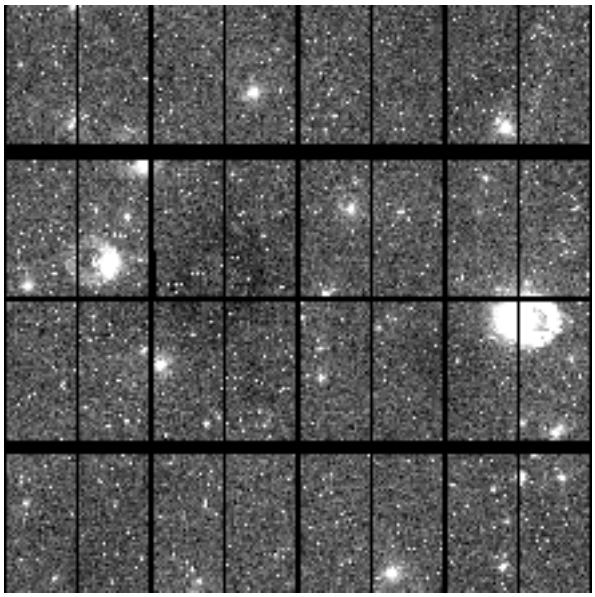

Figure 2.1 – The main FITS image that I used

The pipeline is being run on the entire file, with the coordinates that are relative to the entire file : the origin is at the bottom left of the fits file. The first part of this project, which is about line generation, is completely based on this file Figure 2.1. I chose this file because we can clearly see a really interesting tracks that goes through the image :

Figure 2.2 – The interesting streak

This track on Figure 2.2 was clearly visible and the variation of intensity can also be notified, so it was a great way to test the pipeline with a not too tricky streak. Thanks to Marcellin, he could identify this and it turned out that this streak is actually a debris of the European carrier rocket Ariane 2 ! We know that it is at a geostationnary altitude, so ~36'000 [km] and has a speed of ~3065 [m/s]. I'll use these informations later to analyse the coherence of the result obtained by the pipeline.

Chapter 3

Line generation

When I started this project, we had no data about rotation of satellites and their corresponding tracks on a file, so I had to first generate streaks on the image myself, to be able to test the pipeline. First, to generate a line I had to chose a existing star in the file and cut it out to have a PSF to convolve my line with. The PSF : "Point Spread Function", is like the quantity of blurring of a point object. Here, I had to use it so that my generated line seems more realistic in the image.

Figure 3.1 – The line without using the PSF

Figure 3.2 – The line using the PSF

I will now explain the method I have used to achieve this result. First, I have to chose a star by hand and cut it out to use it as a PSF. The dimension of the star cutted out is 30px, so I initialized an array of height's dimension of 30px and an arbitrary width, with zeros everywhere, except at the center of the array, the center line of 1 pixel of height at the center is set with sinusoid values. Once, I have this two pieces, I can convolve the two arrays together to have the final line with an adequate PSF. I finally rotated the final convolved line by a random angle and add it to the image by summing up the values with the image's array. The final result on the image is pretty satisfying and coherent.

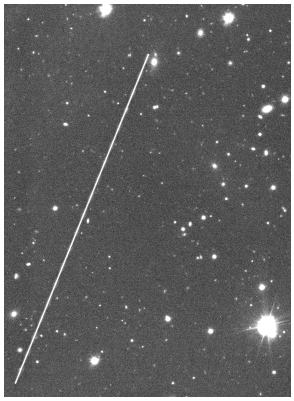

Figure 3.3 – The generated line on the image

The inconvenient of this method is that I had to chose a star by hand and it will work great for this specific file, if we want to make this code works with another image we

have to chose a star from the new file to have a coherent PSF to integrate the line better on it.

Chapter 4

Implementation

The implementation covers some of the implementation details of your project. This is not intended to be a low level description of every line of code that you wrote but covers the implementation aspects of the projects.

This section is usually 3-5 pages.

Chapter 5

Evaluation

In the evaluation you convince the reader that your design works as intended. Describe the evaluation setup, the designed experiments, and how the experiments showcase the individual points you want to prove.

This section is usually 5-10 pages.

Chapter 6

Related Work

The related work section covers closely related work. Here you can highlight the related work, how it solved the problem, and why it solved a different problem. Do not play down the importance of related work, all of these systems have been published and evaluated! Say what is different and how you overcome some of the weaknesses of related work by discussing the trade-offs. Stay positive!

This section is usually 3-5 pages.

Chapter 7

Conclusion

In the conclusion you repeat the main result and finalize the discussion of your project. Mention the core results and why as well as how your system advances the status quo.

Bibliography

[1] Mathias Payer. *Template for EPFL (BSc, MSc, or doctoral) theses and semester projects.* 2020. URL: https://github.com/HexHive/thesis_template (visited on 2021).

[2] Yann Bouquet. *Detectsat Repository.* 2020. URL: https://github.com/YBouquet/detectsat (visited on 2021).