Artificial Intelligence application to Image Processing for autonomous navigation around small bodies

Project preview

Date: 19/05/2020
Stardust-R WPs involved: WP6, WP7, WP1
Mentors:

- Pablo Hermosín (Deimos)
- Pelayo Peñarroya (Deimos)
- Victor Rodriguez (UPM)
- Emma Stevenson (UPM)

Programming languages: Python

Abstract & objectives

The use of image processing techniques for autonomous navigation is one of the main features of space missions to small bodies. At Deimos Space, there is a high degree of expertise developed during missions like NEO-Shield, where typical techniques like detection and tracking algorithms were used in hovering scenarios. These techniques, however, are more and more often replaced by novel learning-based methods that outclass the former in computational time, a key feature when it comes to autonomous operations in space. In the frame of the Stardust-R first Local Training Workshop, Deimos Space will talk about the basic concepts of image processing and how the company adapted to these new techniques.

After that, a hands-on lecture will be given to the attendants to the talk so they can experiment and replicate the results that a basic autonomous navigation system would obtain when using machine learning techniques to process the images taken by the onboard instruments.

Backgrounds for attendees

Navigation sections

Attendees should be familiar with basic concepts related to the exercise, such as:

- Asteroid mission phases.
- Autonomous relative navigation.
- Filtering: sequential and batch estimation techniques (least squares, Kalman filters…).
- Orbit determination and parameter estimation.
- Asteroid environment modelling.
- Observation generation and processing.

A great starting point for this part of the session would be Simone's talk on "Deep space navigation & Navigation around small irregular bodies" (parts 1 and 2), given during the Opening Training School for Stardust-R in Glasgow.

The introductory part of the session will feature traditional feature detection and tracking methods[1] and their performances. The high-level concepts listed above should suffice to get a common understanding of the navigation algorithms used during the practical session.

For the final part of the session, a simple navigation filter will be programmed using FilterPy, a very robust and extended library for filtering applications. A very recommended lecture that encompasses both basics about the library and different filtering techniques (third bullet above) is this book (given as jupyter notebook in the link) by the author of the library, containing very well-explained examples that apply very nicely to the orbital navigation problem we will encounter in the session.

Machine learning section

Attendees should be familiar with basic concepts around deep learning and neural networks, such as:

- Supervised vs unsupervised learning. In this, we'll do a regression task, which is a type of supervised learning.
- Training and validation data
- Overfitting.
- Performance of a regression model.
- Parameters (weights, biases,...) & activations in a neural network
- Batch learning, learning rate

[1] E. Johnson and H. Matthies, "Motion Estimation for Autonomous Small Body Exploration," p. 8.

A good resource to review all of these concepts can be found in the talk "*Deep Learning for Space Guidance*", given by Roberto Furfaro & Richard Linares during the Stardust-R Training School II [2].

In practical terms, we will make use of the Python deep learning library fastai2 [3]. A good introduction to this library, and to practical deep learning in general, can be found in the first chapter of the fastai book, which is freely accessible here. In this exercise, we will work with image data, so attendees should make special focus on reading about that type of learning.

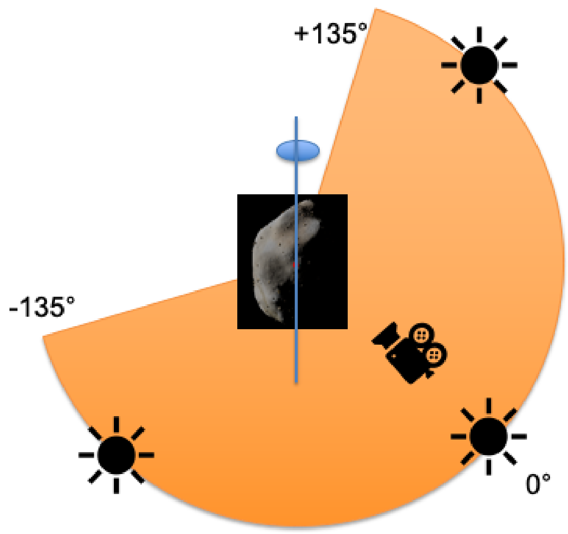
As a development environment, any Python 3 setup is suitable for this exercise. The use of Jupyter Notebooks is recommended due to its exploratory nature for drafting code and results. In this sense, we encourage you to use services like Google Colaboratory, a free[4] Jupyter notebook environment that runs in the cloud and stores its notebooks on Google Drive or Github. It has everything pre-installed and ready-to-use and provides easy sharing & collaborative capabilities for working in teams. Most importantly for this exercise, it also has a freely available GPU which will greatly improve the computational times required for training on an image dataset. A short tutorial on how to use Colab, and enable the GPU, can be found here.

Data description

The data can be find in here (Change to Google Drive repo instead of OneDrive) and is made by:

- "**Centroid**" folder. Containing all the images in .png format enumerated from 00000 to 01367.
- "**Par.txt**" file. Containing the ground truth with the details of each image.

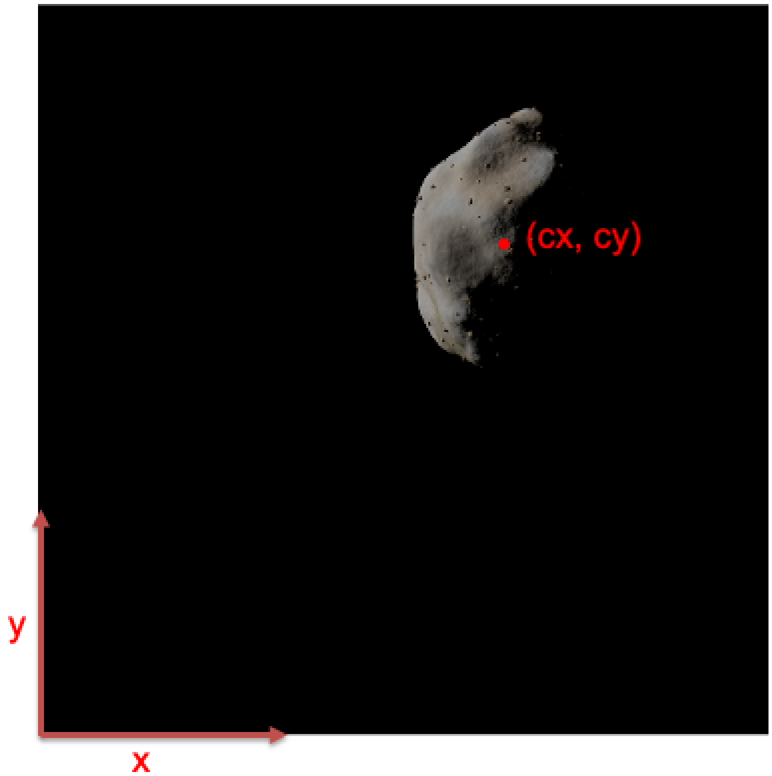
[2] This material has restricted access for the Stardust-R network. If needed, contact the administrators of the workshop for getting the access code.
[3] fastai: A Layered API for Deep Learning: https://arxiv.org/abs/2002.04688
[4] A google account is needed to access Google Colab.



The dataset of images has been generated by 2 degrees of freedom, the rotational angle of the asteroid (Blue) and angle of the illumination vector (Orange). The asteroid rotational angle has been sampled from 0° to 355° with an interval of 5° to capture a variety of different orientations. For each of these, the illumination vector has been varied from -135° to +135° with an interval of 15° with respect to the camera pointing vector. Both camera and Sun are assumed to be coplanar.

The combination of the 72 different orientations with 19 illumination conditions each provide a total of 1368 cases.

For each case a random displacement of the camera within the image plane and a random rotation of the camera boresight axis are added to augment the variability of the asteroid position in each frame.



Each image is 1024x1024 pixels wide and in RGB format. The exact centroid coordinates of the 3D model of the asteroid are listed in the Par.txt file under the cx and cy columns. The coordinates are expressed in pixels from the reference frame illustrated above.

Each column of the Par.txt file contains the following information:

- **ID**: ID of the image name
- **theta_Ast**: rotational angle of the asteroid in degrees
- **theta_Sun**: angle of the illumination vector from the camera point of view in degrees
- **y_CAM**: Camera random displacement in the y-axis
- **z_CAM**: Camera random displacement in the z-axis.
- **theta_CAM**: Camera random rotation around the boresight-axis
- **cx**: x component of the centroid expressed in pixel
- **cy**: y component of the centroid expressed in pixel

```
ID,theta_Ast,theta_Sun,y_CAM,z_CAM,theta_CAM,cx,cy
0,0.0,-135.0,0.08230276354473054,-0.22124172720219526,17.61524095661548,443.7994384765625,599.2282104492188
1,0.0,-120.0,0.08093077357868639,0.19475925816726192,27.22351674601936,520.0352172851562,413.3974609375
2,0.0,-105.0,0.11426805291919984,0.05029548374292758,24.74240423397408,473.1955871582031,468.1398620605469
3,0.0,-90.0,-0.3172555745438347,0.29057476083117617,29.001715641113826,708.237548828125,464.9421081542969
4,0.0,-75.0,0.12635843769828414,-0.38726597208507874,24.121570865977265,383.6676025390625,653.5697631835938
5,0.0,-60.0,-0.35007659183100404,-0.038341192560405135,18.57988978254808,661.9216918945312,581.3687133789062
6,0.0,-45.0,-0.39054756329467943,-0.049454292341675865,12.873957766631465,685.4207153320312,575.4307861328125
7,0.0,-30.0,0.3525754214319994,0.48964639869544,13.811170096965537,406.2292175292969,249.48159790039062
8,0.0,-15.0,-0.46995355141164374,-0.09579612388130365,18.61654546430601,706.5621337890625,624.9553833007812
```

For what concern the task the parameters to be considered are the **ID**, **cx** and **cy**. The other parameters are inserted to provide context.