POLITECNICO
MILANO 1863

# Analysis and validation of spaceborne synthetic imagery using a Vision-Based pose initialization algorithm for non-cooperative spacecrafts

**Francescodario Cuzzocrea**

April 18, 2021

D-ORBIT
NEW SPACE SOLUTIONS

# Introduction

# Table of contents





**Aim of the thesis** $\longrightarrow$ Develop a data-set of images representative of in-orbit real operative conditions and test them using a Computer Vision (CV) algorithm.

# Motivation



**Nowadays, different classes of missions envision a major role for close-range proximity operations.**

Purposes:

- Formation Flying (FF);
- On-Orbit Servicing (OOS);
- Active Debris Removal (ADR).

The main challenge when performing close-range operations in OOS and ADR missions is when the target S/C may be uncooperative (no markers or other specific supportive means). Thus, when operating in close-proximity the time evolution of the target's pose must be estimated in autonomy by exploiting the sensors available on the S/C actively performing the servicing maneuvers.

# Motivation



## Technological Aspects.

Available options:

- LIDAR;
- Stereo Cameras;
- Monocular Camera.

If a **monocular solution** is chosen:

**Advantages**

- low hardware complexity;
- low cost;
- low weight;
- low power consumption.

**Disadvantages**

- increased algorithmic complexity;
- less robust to adverse illumination.

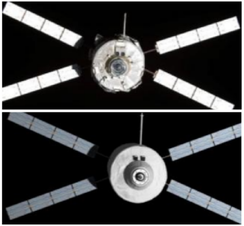
⟶ Need a representative data-set to develop complex CV algorithm.

# State-of-the-Art and Limitations

## Commercial Solutions

ESA PANGU

Initially developed to simulate synthetic planetary surface, has been extended to simulate also S/C.



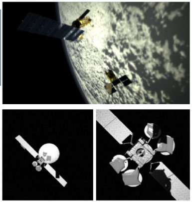
- OpenGL image rendering;
- Can make use of GPU cores;
- Closed-Source.

→ Need to be ESA Project contractor to have a license, or it needs to be negotiated per case.

## Commercial Solutions

Airbus SurRender

Can handle various space objects such as planets, asteroids, satellites and S/C. Can accomodate solar system-sized scenes without precision loss.



- Ray tracing image rendering;
- Fully parallelized;
- Allows to run RT HIL simulations;
- Closed-Source.

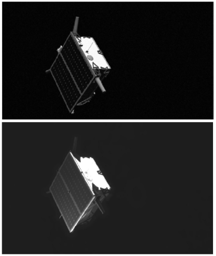
⟶ Need to buy a (costly) license from Airbus.

# Synthetic Image Generation

## Academic Solutions

SPEED Data-Set

Developed at Stanford University. Comprehends both real and synthetic images of the Tango S/C from the PRISMA mission.



- TRON facility for real images;
- Uses true Earth images;
- OpenGL image rendering;
- Closed-Source.

→ Only the Tango S/C is represented in the images.

## Academic Solutions

URSO Data-Set

Developed at University of Surrey. Comprehends synthetic images of Soyuz and Dragon S/C orbiting the Earth.

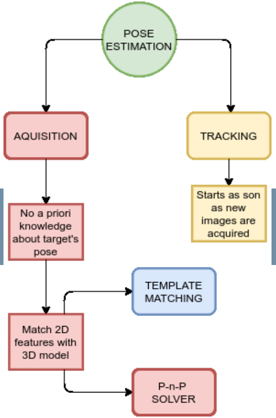
- Implemented using Unreal Engine 4;
- Developed to train neural networks;
- Open-Source rendering engine.

⟶ Lack of photometric accuracy.

## Monocular Solution

### Requirements

- Target 3D model availability
- On-Board real-time execution
- Robustness to illumination
- Robustness to background
- Robustness to variability of pose



## Monocular Based Pose Estimation



### Feature Extraction

Different strategies could be employed, among which:

Harris Corner Detector: Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas

Hough Transform: Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas Bananas

## Matching Procedure

Template Matching:

Based on the matching of an assigned template with specific features and/or image sections detected in the acquired data:

- needs pre-computed renderings of the target;
- changes in the target orientation and position may significantly affect its appearance.

Solve the P-*n*-P problem:

Uses a set of 2D/3D point correspondences to solve true perspective equations:

- need to solve nonlinear equations;
- can have infinite solution in under-constrained situations.

## Requirements

→ Emerges the need for a tool capable of rendering images for CV algorithms training which must:

- be affordable for small research center or small companies;
- be capable of rendering different kinds S/C;
- be able to accomodate solar-system sized scenes.
- be able to also render composite background.

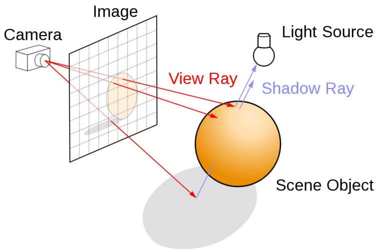
→ The images have to be representative of a real data-set taken by a true camera!

→ A lack of realism can lead to wrong assumptions and thus incoherent results!!!

## Ray Tracing

### POV-Ray represents a viable solution

- Support **mesh** and objects;
- Customizable optical properties;
- Customizable light properties;
- Customizable textures support;
- Customizable camera parameters;
- Used by ESA before PANGU.

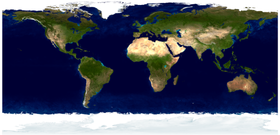



→ It simulates the physics of light!

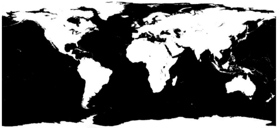
→ Noise can be modeled and added *a posteriori* to enhance fidelity.

# Earth Modeling

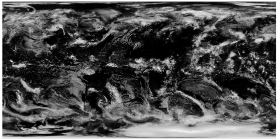
→ Use three different layers to differentiate optical parameters and textures for seasonality variations.



True Color mercator image

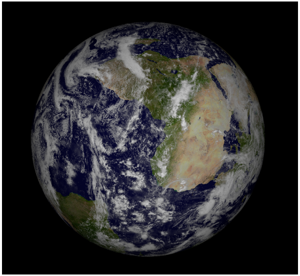


Landmask mercator image

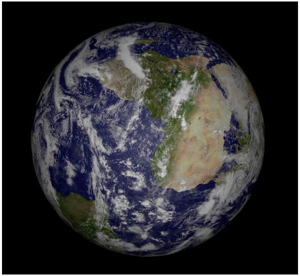


Cloud layer mercator image

## Earth Modeling



No differentiation of optical parameters

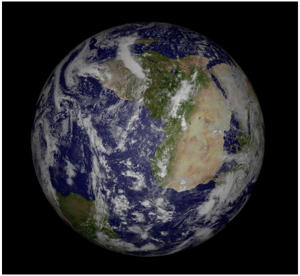


Differentiation of optical paramters

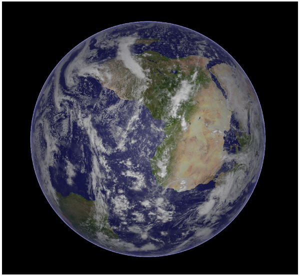
→ Lack of atmosphere model.

## Earth Modeling

→ Add 25 km thick atmosphere augment realness and challenge edge detection CV algorithms.



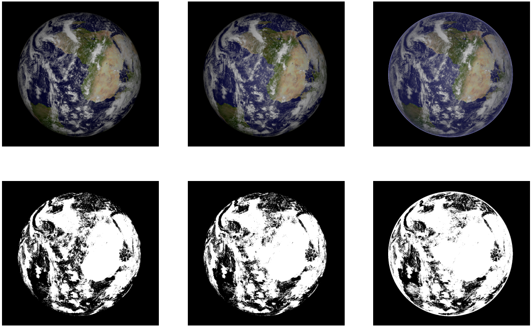
No atmosphere



Atmosphere

## Earth Modeling

→ Binary image Comparison.



## Light Modeling

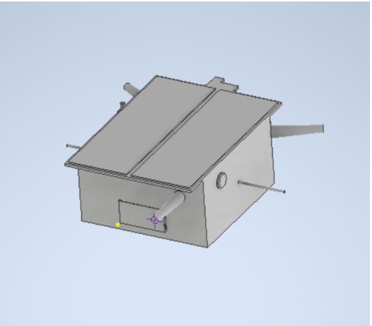
→ The Sun can be modeled using POV-Ray **area_light** feature:

- it create a cluster of point-like light sources distributed on a disc;
- the radius of the disc is set to the radius of the Sun, and placed at the exact distance which the Sun has from the Earth;
- the **orient** option makes every object see the Sun’s disc as oriented toward it, from any position around it.

## 3D Model of the Spacecraft

→ Based on the TANGO spacecraft from the PRISMA mission:

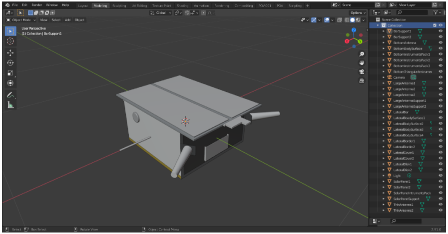
- 570 mm x 759 mm solar panel;
- 560 mm x 550 mm x 300 mm body;
- 4x 204 mm antennas;
- 1x xxx mm bar;



## 3D Model preparation with Blender

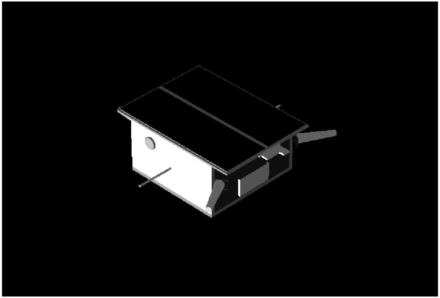
⟶ Exploit Blender POV-Ray add-on:

- define surfaces
- apply textures
- apply optical parameters
- get raw SDL code



## 3D model rendering with POV-Ray

⟶ Remanipulate Blender POV-Ray add-on SDL code for imposing the pose more easily and render the image



## Simulate a real camera

⟶ POV-Ray allows to set camera aperture angle:

$$A.R. = \frac{N_u}{N_v}$$

$$CCD_{size} = d_u \cdot N_u$$

$$\alpha = 2 \cdot \arctan\left(\frac{CCD_{size}}{2 \cdot f_x}\right)$$

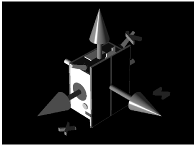
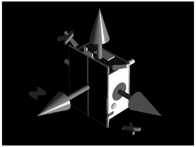
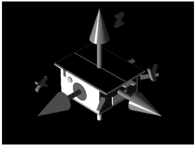
| | |
|---|---|
| $N_u$ | Number of horizontal pixels |
| $N_v$ | Number of vertical pixels |
| $f_x$ | Horizontal focal length |
| $f_y$ | Vertical focal length |
| $d_u$ | Horizontal pixel length |
| $d_v$ | Vertical pixel length |

⟶ Under the assumption of square CCD sensor having square pixels.

## Reference Frames and Ground Truth Pose

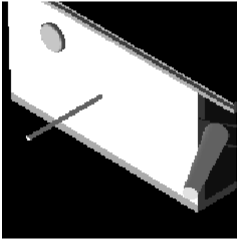
→ POV-Ray by default works using a Left Handed Coordinate System

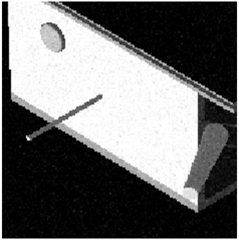
- Flip it to use a Right Handed Coordinate System
- Compute camera attitude by the knowledge of camera position and camera looking direction
- Compute ground truth pose by knowledge of camera attitude and target attitude

## Add Gaussian withe noise and speckle noise

$\longrightarrow$ To enhance the fidelity noise is added to the image





Speckle: $\sigma^2 = 0.004$
Gaussian: $\sigma^2 = 0.003$

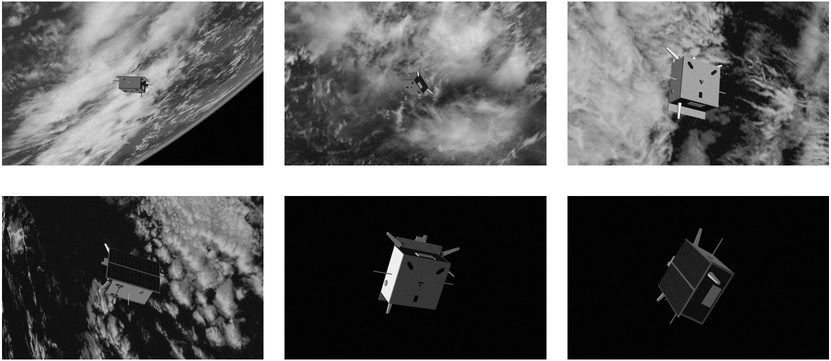
## Simulink Model

- Simulate uncontrolled behavior on LEO orbit
- Easily extendable to include controlled behavior too
- Chaser constrained to be between 3 to 18 meters far from the target
- Extendable to simulate precise chaser approach

## MATLAB Functions

- Write `.pov` SDL files for each image
- Annotate ground truth pose for each image
- Sets relevant POV-Ray options and run it
- Can re-parse ground truth pose for each image

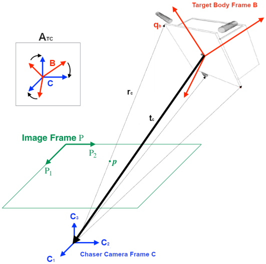
## Samples



**WARNING**: Custom POV-Ray version needed to replicate results!

# The SVD Architecture for Pose Initialization

## Pose estimation problem




2D/3D true perspective equations

$$\mathbf{r_C} = \begin{bmatrix} x_C & y_C & z_C \end{bmatrix}^T = \mathbf{A_{TC}}\,\mathbf{q_B} + \mathbf{t_C}$$

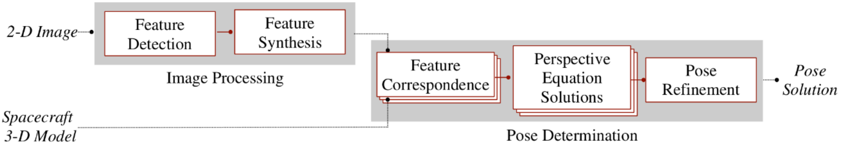
$$\mathbf{p} = \left[ \frac{x_C}{z_C} f_x + C_x,\ \frac{y_C}{z_C} f_y + C_y \right]$$

$$\longrightarrow (\mathbf{A_{TC}}, \mathbf{t_C}) \longleftarrow$$

- Highly non linear and can have $\infty$ solutions if undercostrained
- No *a priori* knowledge of the correspondence between $\mathbf{q_B}$ and $\mathbf{p}$
- Image has to be corrected for pixel's non-quadratism and lens distortion

## General architecture

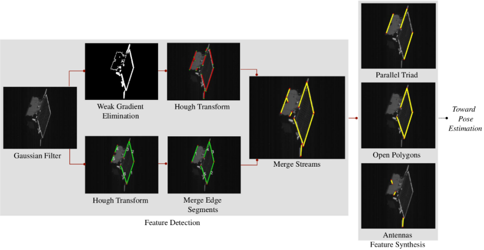
→ **Custom re-implementation of SVD algorithm from the ground up**




→ **Enhance current State-of-the-art**

- Introduces WGE technique which allows to distinguish S/C from background and allows to determine a ROI
- Uses 2D/3D feature groups to solve feature correspondence problem

# Image processing subsystem - 1



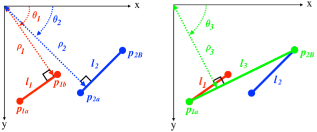

→ Two streams of features are extracted using Hough transform:

- WGE stream
- S&H stream

→ Multiple truncated edges are merged into one line

→ Features are organized in perceptual groups

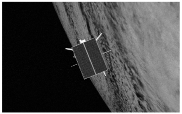
- The knowledge of the ROI allows to tune Hough hyperparameters adaptively to obtain desired behavior
- Edges outside the ROI are rejected



## Image processing subsystem - 2

### → Weak Gradient Eliminator

Most of the gradient intensities are weak and corresponds to the feature in the background or on the spacecraft surface.



Original image

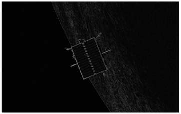
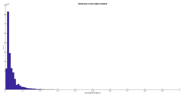
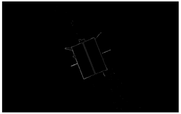


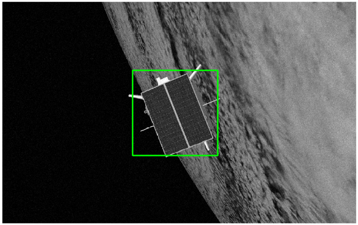
Image gradient



Histogram



Filtered image

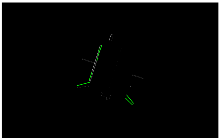


## Image processing subsystem - 3

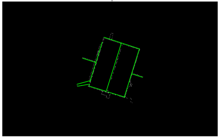
→ **Custom implementation**

- ROI it’s enlarged by adding the 5% of the mean detected edge length to not penalize images presenting a black background
- Prior applying Hough transform to the Sobel image, to eliminate the pixel chunks belonging to smaller reflective elements from the output of Hough, all connected components (objects) that have fewer than P pixels from the input binary image are removed
- The edge merging procedure is applied to both streams

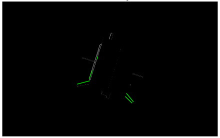
# Image processing subsystem - 4



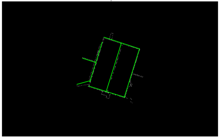
WGE stream



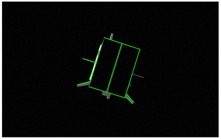
S&H stream



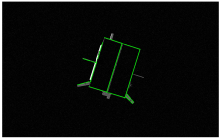
WGE merge



S&H merge



WGE + S&H

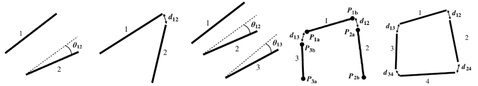


Merge stream

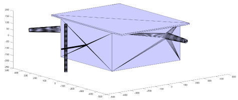
# Perceptual Grouping

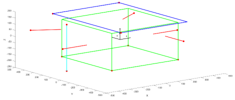
⟶ **Allows to reduce the search space of the correspondence problem**

Perceptual groups are defined on the basis of some geometrical constraints



Parallel Pair | Proximity Pair | Parallel Triad | Proximity Triad | Closed Polygonal Tetrad

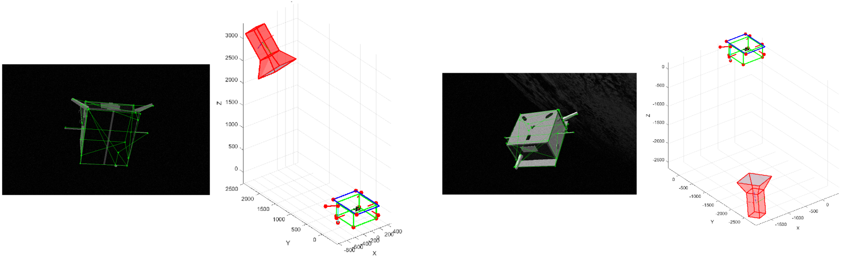




- Carried out on a wireframe model too by applying 3D geometry definitions instead of 2D ones

## Successful pose initialization

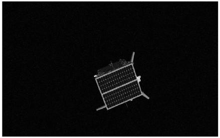
⟶ Obtained using P3P solver coupled with a RANSAC algorithm for outlier rejection instead of ePnP



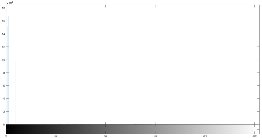

Figures are showing the map superposed to the image and the 3D camera pose in map coordinates

# Results

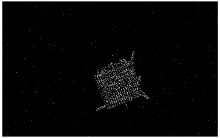
# SPEED VS Implemented Tool



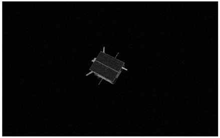
Unfiltered Image



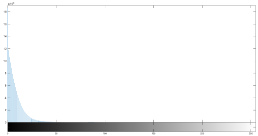
Histogram



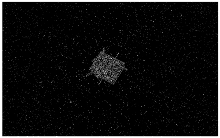
Sobel Image



Unfiltered Image

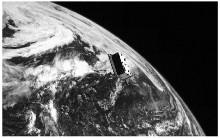


Histogram

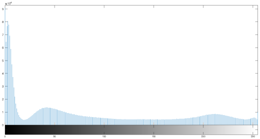


Sobel Image

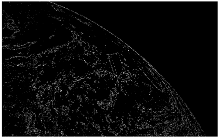
# SPEED VS Implemented Tool



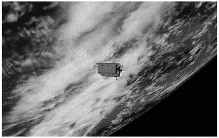
Unfiltered Image



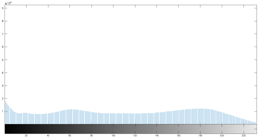
Histogram



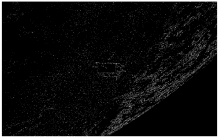
Sobel Image



Unfiltered Image



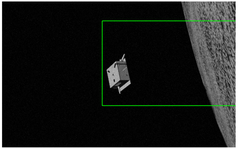
Histogram



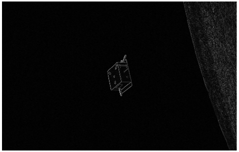
Sobel Image

## Failures

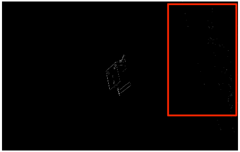
ROI Detection ⟶ Due to composite background



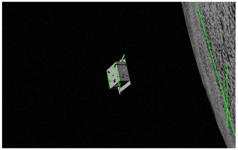
Detected ROI



Normalized Gradient



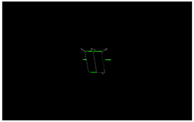
Tresholded Gradient



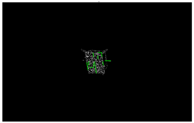
Edge Detection

## Failures

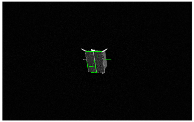
Edge Detection → Due to solar panels or distance



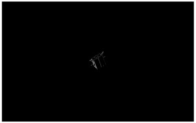
WGE Stream



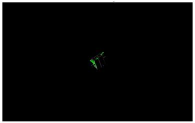
S&H Stream



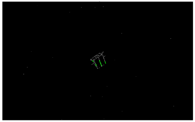
Merging streams



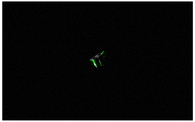
Tresholded Gradient



WGE Stream



S&H Stream

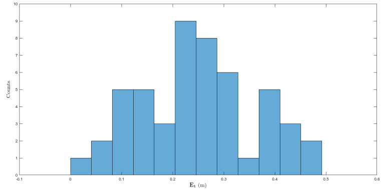


Merging Streams

## Rotational and translational errors

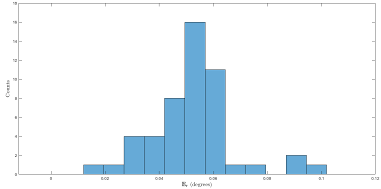
Translational Error

→ Mean error: $0.2506$ m




Rotational Error

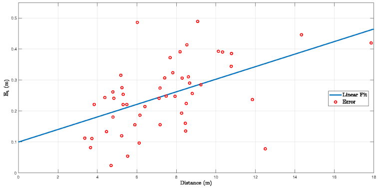
→ Mean error: $0.0528^{\circ}$



# Errors WRT inter-spacecraft distance

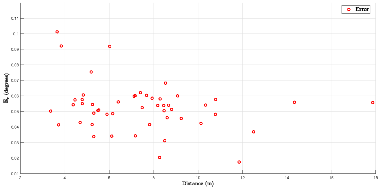
Translational Error

→ Increases with distance



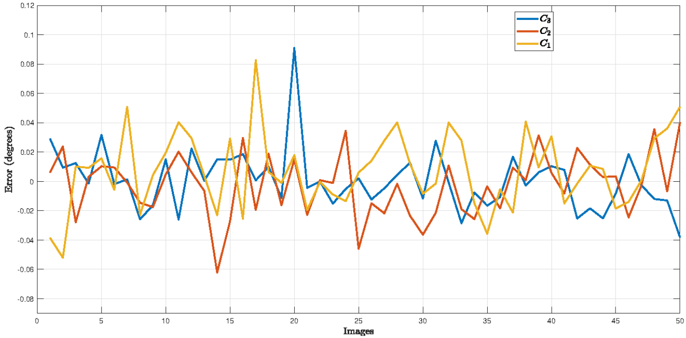

Rotational Error

→ No recognizable trend



# Camera axis error

→ No preferred axis



# Conclusions

## Image Generation Tool

- → Image toolbox is fully automated
- → Controlled and uncontrolled behavior can be simulated as well as well as approach
- → Generated images seems are promising

## SVD Algorithm

- → Successfully tested SVD algorithm
- → Results seems to be promising

## Image Generation Tool

→ Use better textures for the S/C and the Earth

→ Use a more rigorous method to set optical parameters

→ Improve noise model

→ Patch POV-Ray code to allow GPU cores usage for rendering

## SVD Algorithm

→ Employ separated Hough transforms to detect different geometric shapes

→ Improve edge detection procedure (solar panels, distance are an issue)

→ Compare different pose solvers

→ Hardware-in-the-loop experiments to evaluate the computational time on real H/W.

*Thanks for your attention*