

MOON LIMB-BASED AUTONOMOUS OPTICAL NAVIGATION USING STAR TRACKERS

C. Balossi^{*}, F. Piccolo[†], P. Panicucci[‡], M. Pugliatti[§], F. Topputo^{**},
F. Capolupo^{††}

Star trackers (STR) are optical sensors that are widely used on spacecraft for inertial attitude determination. The optical characteristics of STR hardware are optimized for the acquisition of star images. To enhance the accuracy of star centroiding algorithms, STRs intentionally operate with defocused images and with exposure times high enough to properly observe faint objects. In recent years, a number of space missions have explored the potential of vision-based navigation (VBN) strategies relying on images from dedicated navigation cameras. This paper presents a limb-based optical navigation strategy specifically adapted to use STR instead of typical navigation cameras. Indeed, using STRs to perform VBN could simplify spacecraft design and reduce costs. The image processing (IP) pipeline relies on portions of the scene captured by the sensor to detect the lunar limb and estimate the position of the Moon relative to the satellite. The full state of the satellite is then reconstructed using an extended Kalman filter. The study examines the variation in achievable performance with STR exposure time and assess the statistical robustness of the algorithm to various sources of uncertainty with a Monte Carlo campaign.

INTRODUCTION

Optical navigation algorithms determine the state of a spacecraft by processing information coming from images of celestial objects, e.g., planets, visible from the satellite orbit. This goal is typically achieved using a dedicated navigation camera. Mid-range optical navigation includes missions that foresee fly-by, close approach, or bounded orbits around moons or planets, at a distance such that celestial bodies are entirely resolved and occupy hundreds of pixels in the image. Numerous ongoing and future missions are focusing on development and testing of mid-range navigation pipelines. Space agencies such as ESA and NASA have ambitious and innovative plans for the exploration of the solar system, with a particular interest on Mars, Jupiter, and the Moon. Examples of missions within this navigation regime include JUICE, NASA's Artemis lunar missions, and LUMIO, a CubeSat designed to investigate meteoroid impacts on the far side of the Moon.

^{*} PhD student, Department of Aerospace Science and Technology, Politecnico di Milano, claudia.balossi@polimi.it.

[†] PhD student, Department of Aerospace Science and Technology, Politecnico di Milano, felice.piccolo@polimi.it..

[‡] Assistant Professor, Department of Aerospace Science and Technology, Politecnico di Milano, paolo.panicucci@polimi.it.

[§] PhD student, Department of Aerospace Science and Technology, Politecnico di Milano, mattia.pugliatti@polimi.it.

^{**} Full professor, Department of Aerospace Science and Technology, Politecnico di Milano, francesco.topputo@polimi.it.

^{††} Guidance, Navigation and Control Systems Engineer, European Space Research and Technology Centre, ESA, francesco.capolupo@esa.int.

JUICE (Jupiter Icy moons Explorer) is a European mission that was launched in 2023 and is scheduled to reach Jupiter in 2031.¹ The mission's objective is to spend 3 years within the Jupiter system, conducting detailed observations of the gas giant and three of its largest moons: Ganymede, Callisto, and Europa. JUICE is equipped with a comprehensive suite of instruments and sensors that includes navigation cameras to perform optical navigation. JUICE's vision-based navigation strategy is horizon-based and mid-range: it exploits the Jovian moon limbs to determine the spacecraft relative distance with respect to the moon.

LUMIO, short for Lunar Meteoroid Impacts Observer is the winner of ESA's SysNova Competition No. 4 "Lunar CubeSats for Exploration".² The mission is designed to work on a quasi-halo orbit around the Earth-Moon L2 point and its main objective is to observe, quantify, and characterize the meteoroid impacts by detecting the impact flashes on the lunar far-side. A navigation camera onboard the CubeSat will be used to perform optical navigation by pointing at the Moon and processing information extracted from the images.

The Artemis I mission launch in 2022 sent the Orion spacecraft to a Distant Retrograde Orbit in the Earth-Moon system. The mission is equipped with an onboard Optical Navigation (OpNav) system that serves as a backup source of the spacecraft navigation state in case of permanent communication loss. By acquiring images of the Earth or Moon and focusing on the reconstruction of the lit limb, the software can estimate the relative position to the observed celestial body. Reference 3 documents the performance of the Optical Navigation System during the Artemis I flight compared to the pre-flight error models.

The state of a satellite is fully characterized when both its translational (position and velocity) and rotational (orientation and angular velocity) components are known. A variety of attitude determination sensors have been designed to date, but they all work on the same principle: when mounted on the spacecraft, the instrument senses the external world in accordance with the spacecraft's orientation. Amongst such sensors, star trackers are perhaps the most widely used. A star tracker (STR) is a camera that takes starfield images to infer the spacecraft attitude. Indeed, by recognizing star patterns in the scene, it is possible to understand where the camera, and thus the spacecraft, is pointing.

ESA's "Star Tracker Autonomous Relative Navigation (STAR Nav)" project, led by Politecnico di Milano in collaboration with Leonardo, aims at exploring the use of star trackers for autonomous optical navigation. The interest in this change of paradigm is twofold: the use of identical hardware units for attitude determination and relative navigation would simplify spacecraft design and at the same time reduce mission costs.

This paper illustrates the vision-based navigation algorithm developed within the STAR Nav framework for a limb-based navigation scenario in the proximity of the Moon. The proposed scenario considers a satellite on the same quasi-halo orbit around Earth-Moon L2 as LUMIO. Given the selected orbit, the star tracker can provide images of the entire lunar disk, making horizon-based (or limb-based) optical navigation the most suitable choice to estimate the position of the vehicle with respect to the Moon. A crucial aspect of the navigation algorithm is its compliance with the hardware limitations of the STR. Indeed, because of memory constraints, the instrument cannot provide full images of the observed scene, as it is usually the case with navigation cameras. Instead, the sensor can output small sub-portions of the original image while retaining the information on their pixel content. As an alternative acquisition mode, the instrument can scan the image to look for groups of pixels with intensity higher than a specified background threshold. This study presents two different image processing strategies developed to work with these particular inputs. Moreover, STR are designed to work at high exposure times, to facilitate the detection of dim stars. Thus, overexposure effects need to be considered when observing big and brighter objects like the Moon.

The following sections describe two OpNav architectures. The first one is tailored on the two STR acquisition modes, but still exploits information coming from ideal correctly-exposed images. The results from this first navigation chain sets the bar for the maximum attainable performance given the limitation in the amount of information extracted from the scene. The second algorithm is instead designed to consider the STR real output, including overexposure, and it is useful to determine the impact of image saturation on the overall performance of the navigation strategy.

OPNAV SCENARIO

The OpNav scenario is similar to that of the LUMIO spacecraft. The selected operative orbit is a quasi-periodic halo orbit around Earth–Moon L2, characterized by a Jacobi constant $C_j = 3.09$.⁴ Along the orbit, the entire lunar disk is visible within the field of view (FOV) of the onboard star tracker. Thus, the most appropriate method for determining the spacecraft’s position relative to the Moon involves a limb-based optical navigation strategy. The performance of limb-based navigation algorithms varies depending on how many pixels in the image are occupied by the celestial body (that is, the distance of the object from the spacecraft), and on the illumination conditions of the scene. The latter is dependent on the relative position of the Sun, the spacecraft, and the observed celestial object. The navigation scenario considers a time interval of 3-days, and a measurement frequency of 5 minutes.

Star Tracker Properties

The primary challenge in star tracker-based optical navigation lies in customizing the navigation algorithms to accommodate the hardware limitations of the star tracker. Within the STAR Nav framework, the selected star tracker is the AASTR2.0 developed by Leonardo S.p.A. The sensor has a field of view of 20 degrees and a focal length of 50.7 mm. The STR camera has a resolution of 1024x1024 px but due to memory limitations within the unit, the sensor is not capable of capturing and storing pixels intensities for the entire image. Instead, information on the observed scene can be reconstructed via two types of outputs: segments and windows.

Segments are grouping of consecutive pixels on a single row, each exhibiting intensity higher than a specified background threshold. These segments are characterized by the row and column index of the initial pixel and the segment’s length. The star tracker can output a maximum number of segments, and it is possible for multiple segments to be identified on the same row. A simplified representation of the segments acquisition of an image is depicted in the left part of Figure 1.

Windows are sub portions of the image for which the pixel intensities’ values can be stored. Also in this case the STR software limits the number of readable windows at each acquisition, but it allows to define the positions of windows to be read. A visual representation of windows is depicted as yellow squares in Figure 1, right: only the content of pixels within the windows are stored.

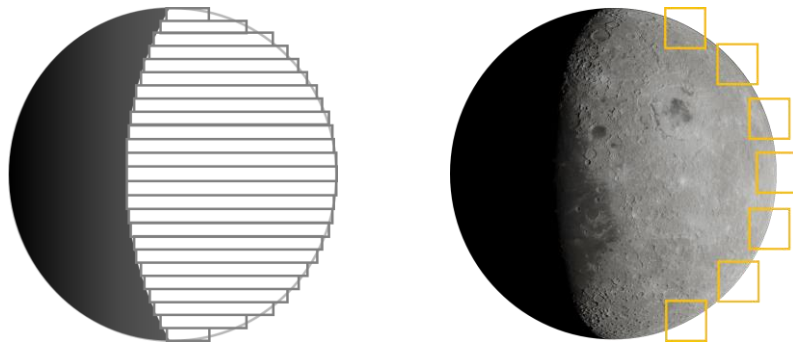


Figure 1: Visual representation of segments (left) and windows (right)

Dataset Generation

The development of a vision-based navigation pipeline for the study needs a dataset of synthetic images capable of reproducing the scene observed from the STR camera along the orbit. The images have been generated in CORTO, a Blender-based rendering engine developed at Politecnico di Milano's DART Lab. CORTO takes as input the spacecraft position and attitude, as well as the Sun position, and produces a realistic image of the scene observed by the camera. The acquisition frequency has been set to 300 s and the dataset comprises 865 renderings of the Moon, with range spanning from 44426 km to 71609 km and phase angle variation from 90 to 40 degrees.

To test the robustness of the strategy, no specific pointing requirement towards the center of the Moon is enforced, allowing the Moon to appear at a random position within the 20-degree field of view of the star tracker. Additionally, the light orientation is randomly directed in each image, ensuring an evaluation of edge detection results in all scan directions. Figure 2 displays several images from the collection.



Figure 2: Sample images from the test dataset

Exposure Time and Image Saturation

Being designed to capture star fields images, the STR usually works at exposure times in the order of hundreds of milliseconds. However, in the significantly off-nominal condition of observing a big and bright object such as the Moon for 200 ms, the STR outputs highly over-exposed images. A reliable proof of this behavior is given by real images of the Moon taken by Leonardo during night sky tests: a scene captured with high exposure time is extremely over-exposed. The minimum exposure time that the sensor can handle is close to 10 ms. In these conditions, the effects of over-exposure are much more contained. It is noteworthy that these acquisitions are degraded by light pollution and by the presence of the Earth atmosphere, and thus images actually taken in space could be slightly different. Nonetheless, to understand to what extent can STR be used in vision-based navigation chains, also the case of overexposed images has to be investigated. Since CORTO is designed to work in nominal illumination conditions, the initial dataset has been processed to reproduce the illumination conditions of the scene captured by the star tracker. This second dataset tries to mimic the STR output in exposure times close to the lower limit. However, the STR behavior in this highly nonlinear working conditions of the detector is not characterized, therefore a validated model of the rendering procedure is still under study. Further hardware-in-the-loop and sensor characterization tests will allow to reproduce over-exposure and distortions at a higher degree of fidelity. Figure 3 displays an example image from the original "correctly-exposed" dataset, as well as its saturated counterpart. The following sections will present the performance of the vision-based navigation procedure for both datasets.



Figure 3: Original image (left) and overexposed version (right)

Detector and Optical Head Effects

Synthetic images are not affected by distortions and noises typically present in real cameras. To test the performance of the navigation algorithm on high-fidelity images, fundamental camera effects are incorporated through post-processing of the datasets. First, the image is distorted according to the characteristics of the camera optics. The STR camera primarily experiences radial distortion and the effect is modeled using the Brown-Conrady distortion model.⁵ Then, focusing errors are reproduced by image convolution with the STR point spread function. Finally, relevant detector noises are added. The most significant illumination-dependent contributions include photo response non-uniformity and photon shot noise. Illumination-independent noises considered in the model are dark current, dark signal non-uniformity, and readout noise.

OPNAV ALGORITHM ARCHITECTURES

Once the synthetic image is generated from the rendering engine of Blender, and modified to account for camera effects, it is processed to extract useful information. This section describes the two OpNav architectures developed for the study. The procedures rely on the assumption that the spacecraft's attitude, as well as the direction of the Sun, are known at each image acquisition time and are composed of three main parts: edge detection, position estimation and navigation filter. The detailed description of these three building blocks is carried out in the following section.

The first navigation chain here presented is developed to work specifically for non-overexposed dataset. As images captured from the STR will always be over-exposed to some degree, this is an ideal scenario and it is presented to set the limit for the best attainable performance given the other STR camera properties and hardware limitations. Having the possibility of exploiting the variation in pixel intensity, the image processing chain combines the two star tracker acquisition modes in a dual-cycle approach. The initial phase involves segments-based acquisition, aiming at acquiring a coarse information regarding the location of the Moon's limb. Subsequently, a second phase employs windows-based acquisition to pinpoint the sub-pixel location of the edge-points. The relative position between the camera and the celestial body is determined from the final set of edge points, employing Christian and Robinson's position estimation algorithm.⁶ The overall translational state of the spacecraft is ultimately reconstructed via an Extended Kalman Filter. The workflow is illustrated in Figure 4.

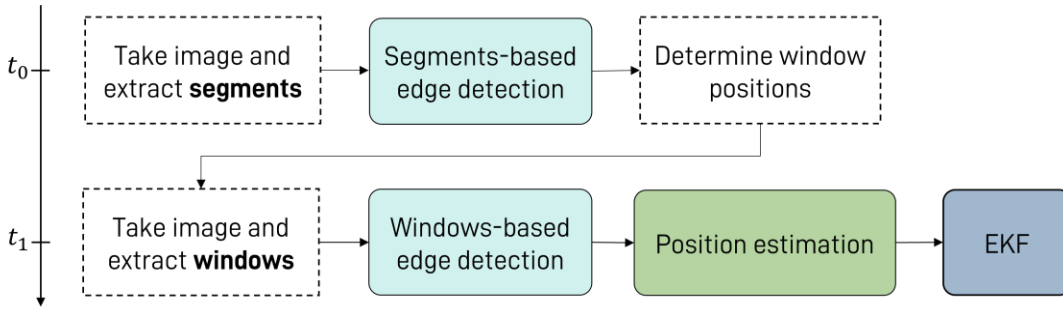


Figure 4: Baseline OpNav algorithm for non-overexposed images

If the pixel content in the image is saturated, no information on the subpixel position of the limb point can be retrieved. Thus, when working with the over-exposed dataset only the segment-based edge detection is meaningful, and points obtained from the processing of segments are directly fed to the position estimation algorithm and eventually to the navigation filter. Figure 5 shows the algorithm architecture.

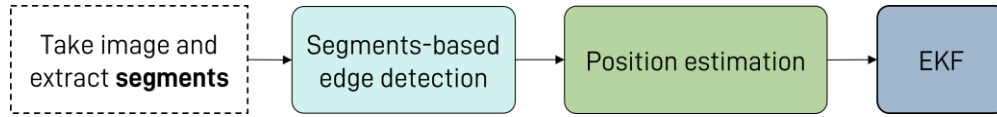


Figure 5: Modified OpNav Algorithm for Over-exposed Images

OPNAV BUILDING BLOCKS

The OpNav procedure proposed in this work is based on three main blocks: edge detection, position estimation and navigation filter.

Edge detection indicates the strategy implemented to recognize the lit horizon of the Moon in the image. Both types of star tracker acquisition modes described in the previous section, i.e. in *segments* or *windows*, can be manipulated to extract information on the position of the Moon's horizon, albeit with different outcomes in terms of accuracy and computational complexity. A detailed description of the two different procedures developed in this sense is described in the first two paragraphs of this section.

Secondly, the set of edge points is manipulated to retrieve the relative position of the Moon with respect to the camera. This step is carried out with the use of Chistian-Robinson's dedicated algorithm. A summary of the position estimation procedure is included in this section.

The last building block described in this section is the Extended Kalman Filter (EKF), that is used to improve the accuracy in the estimated position by merging the value obtained from the image processing with its predicted value propagated via the system dynamics. In addition to this, the navigation filter estimates the velocity of the satellite.

Segments-based Edge Detection

As outlined in the introduction, star trackers can extract illuminated segments in the captured scene. To facilitate the design and testing of the image processing pipeline, a function mimicking the STR output has been developed. Segments extracted from the image can be easily exploited to obtain a coarse estimation (i.e., pixel accuracy) of the coordinates of the points belonging to the illuminated limb. The first step of the algorithm consists in coupling the notions on the Sun position and the spacecraft attitude to define the direction of the light in the image.

Firstly, the projection of the Sun direction onto the image plane is computed as:

$$\hat{r}_{sunproj} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \hat{r}_{sun} \quad (1)$$

where \hat{r}_{sun} is the unit vector describing the direction of the Sun in the camera reference frame. The camera reference frame is defined as follows: X_C and Y_C represent the two in-plane coordinates, whereas Z_C is the out-of-plane axis which points towards the Moon. The angle α_{sun} defining the light direction in the image is obtained from the two components of $\hat{r}_{sunproj}$:

$$\alpha_{sun} = \tan^{-1} \left(\frac{\hat{r}_{sunproj}(2)}{\hat{r}_{sunproj}(1)} \right) \quad (2)$$

From the definition, the angle α_{sun} is zero when the light is directed along the vertical direction, from top to bottom. The light direction is of paramount importance in the edge detection chain as it is needed to distinguish the lit limb from the terminator: indeed, the lit limb is composed by pixels characterized by a sufficiently high intensity change with respect to the background that are found when scanning the image along the direction of light in the image. When working with segments output, to each of the values of α_{sun} is associated a *scan direction*. As this cycle in the algorithm is used to obtain a preliminary solution that will then be refined, it has been chosen to work with 90 degrees intervals and map all the light directions into four scan directions: top-to-bottom, right-to-left, bottom-to-top, left-to-right. Each scan direction is associated with a different processing of the collection of segments:

- Right: when scanning the image horizontally from right to left, the illuminated limb is composed by the collection of the right-most illuminated pixel in each row.
- Left: when scanning the image horizontally from left to right, the illuminated limb is composed by the collection of the first pixel of the left-most segment along each row.
- Top/bottom: when scanning the image vertically, either from top or from bottom, the coordinates to be stored are the ones describing both the first pixel of the first segment and the last pixel of the last segment detected along each row.

A simple visual representation of the scanning directions and consequent point identification is shown in Figure 6, where the considered pixels are highlighted as red dots.

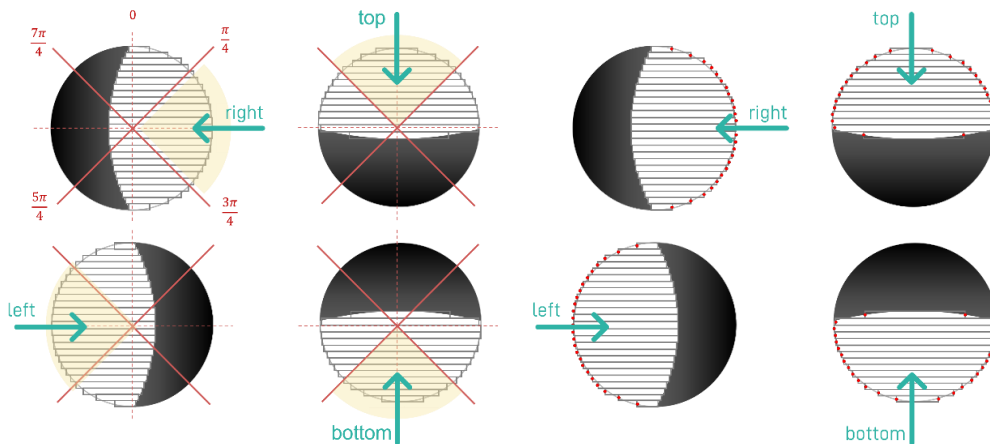


Figure 6: Edge detection from STR segments: definition of scan direction and selection of edge points

Once the collection of points belonging to the limb is retrieved, the last step of the pipeline consists of estimating the Cartesian parameters of a circle that fits in the set of edge points, exploiting a least squares method relying on singular value decomposition.⁷ The outliers are rejected through a Random Sample Consensus (RANSAC) algorithm.⁸

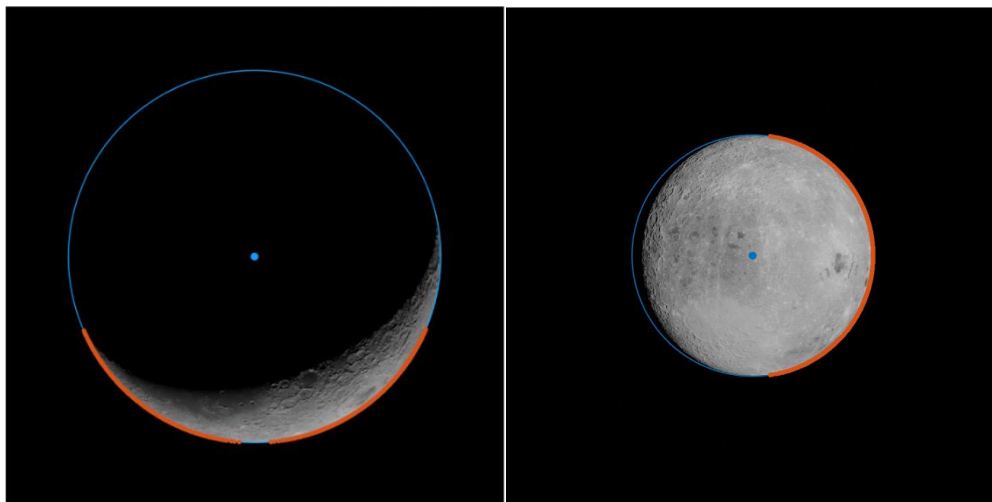


Figure 7: Examples of segments-based edge detection

Figure 7 shows the result of the segments-based edge detection step in two different lighting, apparent Moon size and orientation conditions. The red dots are the points retrieved as inliers from the algorithm, and the blue circle represents the reconstructed Moon limb. The strategy is extremely simple; however, it has the major drawback of granting pixel-level accuracy.

Windows-based Edge Detection

Subpixel precision is required whenever a high level of accuracy is desired. Indeed, an error of one full pixel on the edge detection results in hundreds of km error on the range, depending on the distance to the Moon. The second type of STR output, *windows*, can be exploited to reach sub-pixelic accuracy in the horizon detection, as the information on the pixels intensity contained in the windows is available. However, it is noteworthy that this kind of output is useful only if the pixels content in the image is not saturated. Since the number and size of windows that can be read is limited, the idea is to perform sub-pixel edge detection inside each window and then merge the results through a circle fitting step that rejects the outliers. Firstly, a preliminary edge detection step is performed inside each window. This is done by looking for zero-crossings after filtering the image portion with a Laplacian of Gaussian (LoG) filter. The refinement of every detected limb point is then carried out using Zernike moments as detailed in Reference 9. The algorithm relies on the application of a small image mask (usually 5×5 or 7×7) to refine the position of the boundary inside the pixel itself, by exploiting the relative values of a set of moments.

The main limitation of having the STR working in *windows* mode is the fact that windows' locations are an input to the STR software and must therefore be known a-priori. Thus, since the window must contain a portion of the Moon limb, previous knowledge on the edge position is needed. A set of points is selected by processing the information obtained from segments and windows are opened around these locations. Then, the intensity content of each window is exploited to obtain new edge coordinates. When performing an acquisition in windows mode, several limitations on the number and size of windows must be observed. It is important to select the location of the windows appropriately to ensuring good performance of the IP chain. The workflow proposed

for the determination of windows positions starts with fixing the size of the windows. After a test campaign, windows of dimensions 15 x 15 px are selected. The windows coordinates are computed by considering the radius R_c and center (c_x, c_y) of the circle extracted from the segments-based edge detection cycle, as well as the information on the angle α_{sun} computed previously. Working with radial coordinates, one can easily subdivide the half circle centered in $(0,0)$ and radius R_c in n_{win} equally spaced arcs, where n_{win} is the number of windows. To avoid opening the windows exactly on area where the Moon's limb and terminator meet, an angular margin is considered. The last step consists of the rotation of the set of points according to the angle α_{sun} to match the lit limb orientation, and the final translation of (c_x, c_y) . Two examples of the result of windows extraction are displayed in Figure 8.

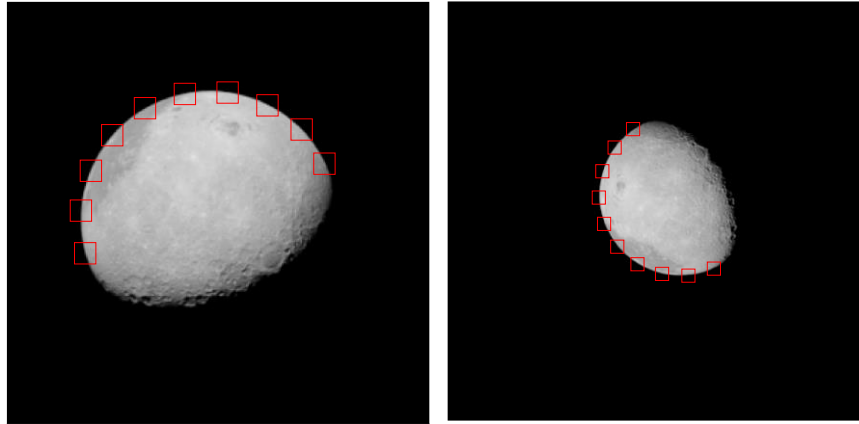


Figure 8: Examples of windows extraction (red boxes)

Figure 9 shows a comparison of the set of edge points retrievable from segment-based (in blue) and windows-based (in red) edge detection. Points extracted from illuminated segments are constrained to be positioned in correspondence of pixels' centers.

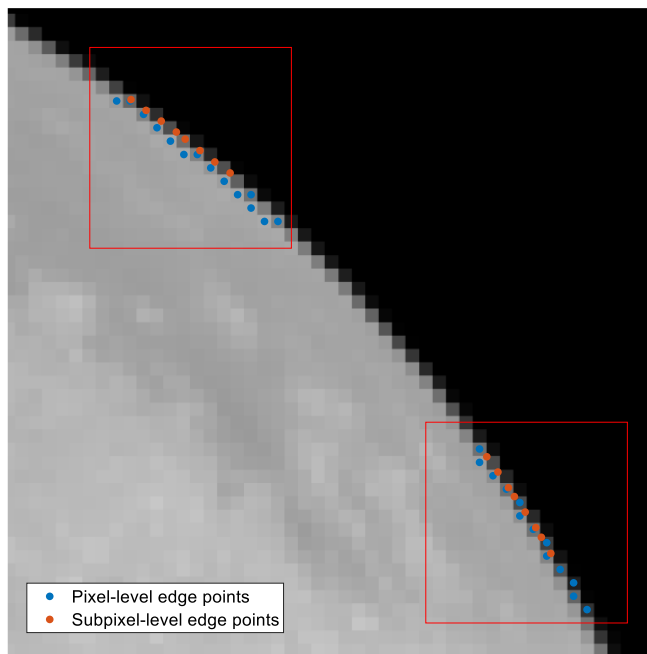


Figure 9: Edge points before (blue) and after (red) sub-pixel refinement

Position estimation

The final set of inliers edge points, either extracted from segments or windows, can be used to determine the position vector \mathbf{r}_c from the camera optical center to the center of the Moon, expressed in the reference frame of the camera. To do so, the algorithm developed by Christian and Robinson is implemented. It considers the factorization of the Moon's shape matrix A_{moon} as:

$$A_{moon} = D^T D = \begin{bmatrix} 1/a & 0 & 0 \\ 0 & 1/b & 0 \\ 0 & 0 & 1/c \end{bmatrix}^T \begin{bmatrix} 1/a & 0 & 0 \\ 0 & 1/b & 0 \\ 0 & 0 & 1/c \end{bmatrix} \quad (3)$$

where a, b, c are the semiaxes of the Moon ellipsoid. Note that $a = b = c = R_{moon}$ in the approximation of spherical Moon. For every edge point i , the following transformation is defined:

$$\begin{aligned} \hat{\mathbf{x}}_i &= DR_{C \setminus P} K^{-1} \mathbf{u}_i \\ \hat{\mathbf{s}}_i &= \frac{\hat{\mathbf{x}}_i}{\|\hat{\mathbf{x}}_i\|} \end{aligned} \quad (4)$$

where K is the intrinsic camera matrix, $R_{C \setminus P}$ indicates the direction cosine matrix (DCM) to go from planet to camera frame. \mathbf{u}_i is the vector containing the coordinates of the edge point expressed in the $u-v$ coordinate frame. It is straightforward to transform the coordinates in \mathbf{u}_i into image coordinates \mathbf{x}_i exploiting the intrinsic camera matrix, as $\mathbf{x}_i = K^{-1} \mathbf{u}_i$. It is possible to construct the linear system:

$$H\mathbf{n} = \mathbf{1}_{n \times 1} \quad (5)$$

with $H = [s_1^T \ s_2^T \ \dots \ s_n^T]^T$. The linear system can be solved in the total least squares sense to retrieve \mathbf{n} . After some manipulation one can compute the position vector \mathbf{r} as

$$\mathbf{r} = (\mathbf{n}^T \mathbf{n} - 1)^{-1/2} R_{C \setminus P} D^{-1} \mathbf{n} \quad (6)$$

Navigation Filter

The navigation filter employed for the study is an Extended Kalman Filter (EKF). The estimated state vector is

$$\mathbf{x} = [\mathbf{r}^T \ \mathbf{v}^T \ \mathbf{a}_r^T \ \mathbf{b}^T]^T \quad (7)$$

\mathbf{r} and \mathbf{v} are the spacecraft position and velocity, expressed in the J2000 reference frame. \mathbf{a}_r and \mathbf{b} are 3x1 vectors of residual accelerations and image processing biases respectively, modelled as first order Gauss-Markov processes. The spacecraft position and velocity are propagated by integrating a dynamical model that accounts for the gravitational attraction of the Moon, the Earth and the Sun and the estimated residual accelerations. The state transition matrix is also integrated together with the equations of motion and the propagation is based on a Runge-Kutta 4 integration scheme. The EKF measurement model is given by:

$$\mathbf{h}(\mathbf{x}) = -R_{C \setminus N}(\mathbf{r} - \mathbf{r}_m) + \mathbf{b} \quad (8)$$

$$H = \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \end{bmatrix} = [-R_{C \setminus N} \quad \mathbf{0}_{3 \times 3} \quad \mathbf{0}_{3 \times 3} \quad \mathbb{I}_{3 \times 3}] \quad (9)$$

where \mathbf{r}_m is the position vector of the Moon, $\mathbf{0}_{3 \times 3}$ is the 3x3 zero matrix and $R_{C \setminus N}$ is the rotation matrix from the J2000 frame to the camera frame, computed as

$$R_{C \setminus N} = R_{C \setminus B} R_{B \setminus N} \quad (10)$$

with $R_{B \setminus N}$ the rotation matrix from J2000 frame to spacecraft body frame and $R_{C \setminus B}$ the rotation matrix from body to camera frame. The two rotation matrices are assumed to be known.

RESULTS

This section illustrates the results obtained from the overall OpNav architectures developed for the study. Uncertainties on input parameters and data must be accounted for when assessing the performance of the vision-based navigation algorithm. Specifically, the initial spacecraft position and velocity are perturbed to account for initial state errors and the spacecraft attitude is modified according to an attitude knowledge of 1 arcsec. Additionally, errors considering star tracker misalignment and perturbations on the camera properties are considered. The uncertainties values that have been considered in the numerical simulations are reported in Table 1.

Table 1: Uncertainties values for limb-based scenario

Uncertainty	Value
Attitude knowledge	10 arcsec (1-sigma).
Initial position	10% (1-sigma) of the true range
Initial velocity	5% (1-sigma) of the true velocity
Focal length calibration residual	1 px (1-sigma)
Optical center calibration residual	0.5 px (1-sigma)
Star Tracker misalignment	100 arcsec (1-sigma).
Radial distortion coefficients	0.1% (1-sigma) of the true value

Baseline OpNav Algorithm for Non-Overexposed Images

The outcome of edge detection on a test image extracted from the dataset is showcased in Figure 10, where the blue set of points represent the limb position extracted through the double-cycle edge detection chain. Since the image suffered from camera distortion, to obtain an accurate estimation of the Moon's position an undistortion step is needed. The red dots in the figure represent the edge points after the distortion removal and they align much better to the actual Moon limb, depicted as a yellow circle in the image. Ultimately, the two crosses denote the estimated and real positions of the Moon's center respectively.

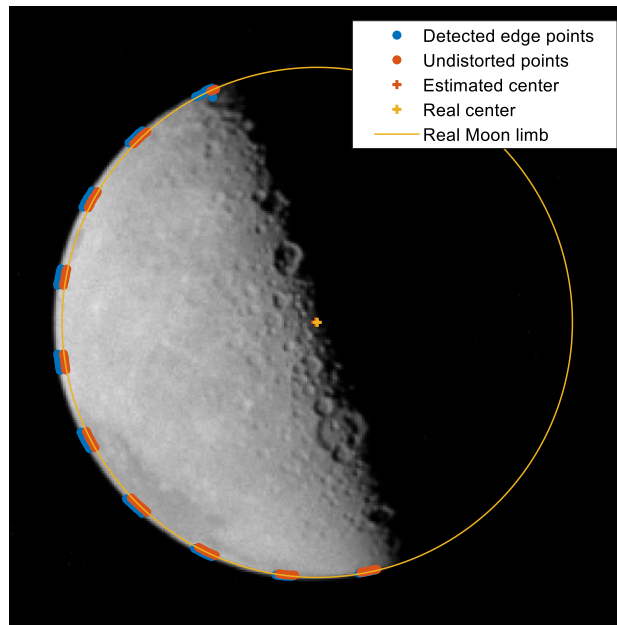


Figure 10: IP results on example image

The numerical simulation carried out on the overall set of 865 non-overexposed images yields a radius estimation error with subpixel accuracy. Figure 11 illustrates the resulting probability density function (PDF) and cumulative distribution function (CDF). It is noteworthy that the circle fitting strategy implemented in the IP chain suffers from a slight positive bias and tends to overestimate the circle radius. This results in a biased estimation of the position of the Moon with respect to the camera, in the camera's boresight direction, as the resulting range is smaller than the actual distance. This effect increases with the distance to the Moon: approaching $t = 3$ days (range > 70000 km) the error on the estimated range is in order of 200-250 km.

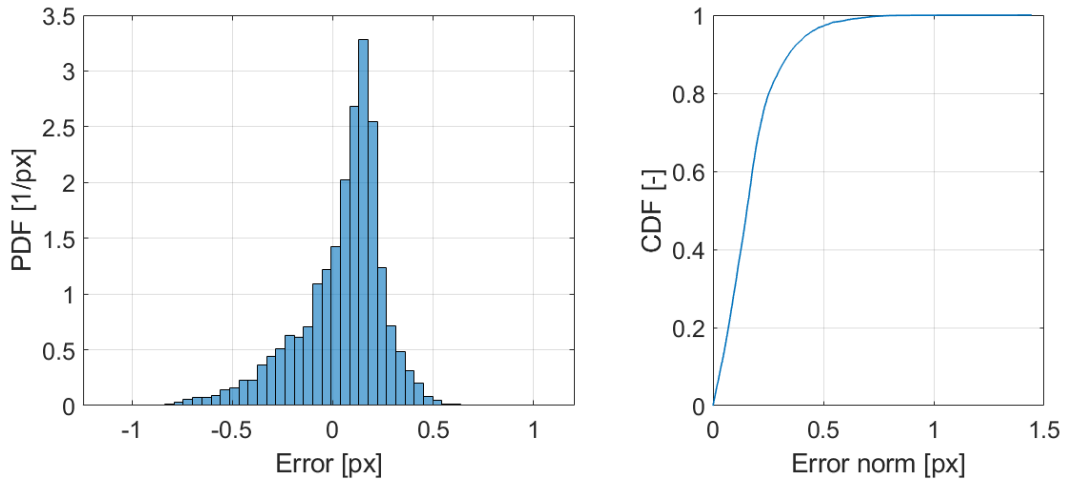


Figure 11: Estimation of Moon radius

The estimation of the position of the center of the Moon in the image is performed with subpixel accuracy. Figure 12 shows the resulting errors with respect to the real location for both the X and Y component in camera frame. We recall that the camera reference frame is defined with X_C and Y_C as the two in-plane coordinates, and Z_C as the boresight direction.

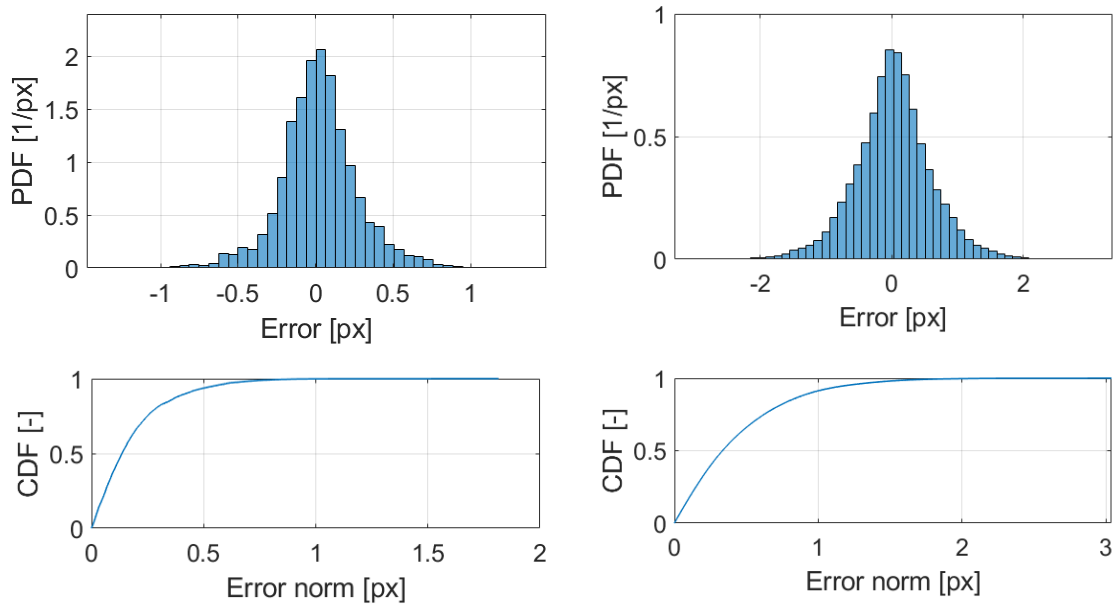


Figure 12: Estimation of circle center

The output from the image processing chain provides the position vector from the satellite to the center of the Moon, expressed in the camera reference frame. The estimation error related to the vector is visible in Figure 13. As expected, the overestimation of the radius leads to an underestimation of the position vector component along the boresight direction.

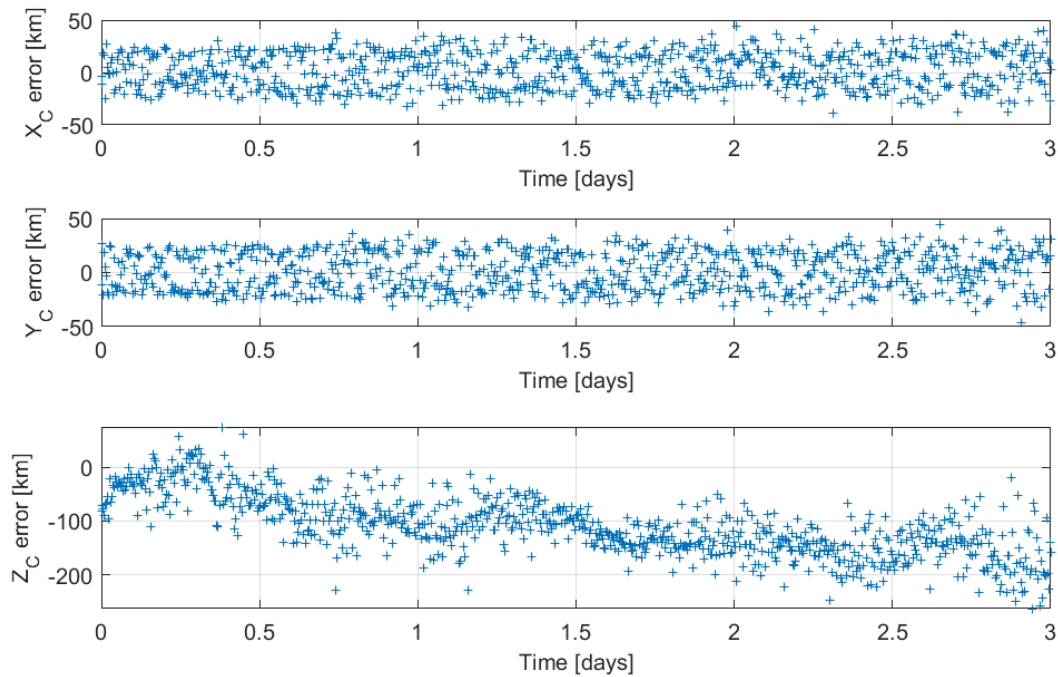


Figure 13: Estimated position vector error

This measurement is then input to the Extended Kalman Filter (EKF) to refine the satellite's position and obtain an estimation of its velocity. The satellite dynamics within the navigation filter

is expressed in the ECLIPJ2000 inertial frame centered in the Earth-Moon barycenter and accounts for the gravitational attractions of the Moon, Earth, Sun, and residual accelerations. The measurement bias coming from the IP is handled as Gauss-Markov process. The statistical performance of the navigation filter is evaluated through a Monte Carlo campaign with 500 runs, and the results are presented in the following plots. The measurement covariance matrix used in the EKF update is provided by the position estimation algorithm, initialized with a covariance value obtained from the image processing results. In the case of the non-overexposed dataset, results are obtained considering a covariance of 1 pixel for edge detection. The analysis incorporates all uncertainties related to initial position and velocity fed to the filter, attitude, as well as calibration errors on focal length, optical center, and radial distortion coefficients.

The EKF outputs the state in the J2000 inertial frame in which the dynamics is expressed but can easily be translated to the Camera Frame. Indeed, the rotation matrix R_{CN} that maps the transformation between the two reference frames is supposed to be known up to the considered uncertainties on attitude knowledge and star tracker misalignment errors.

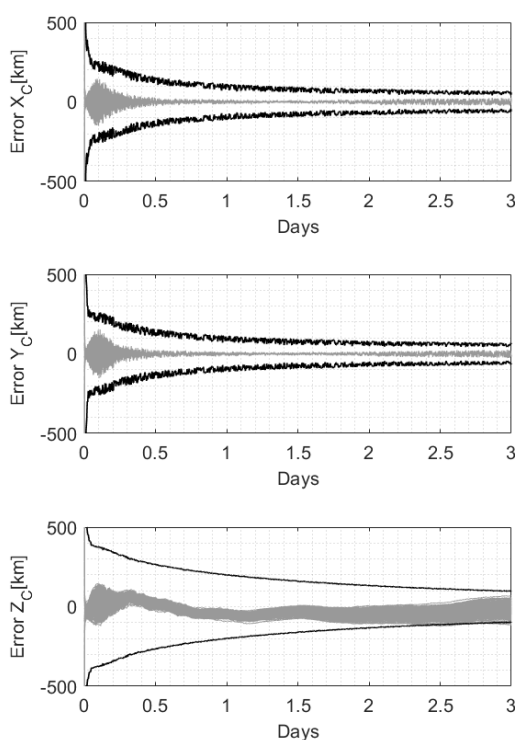


Figure 14: Position error and 3σ for 500 samples of the Monte Carlo simulation, CF

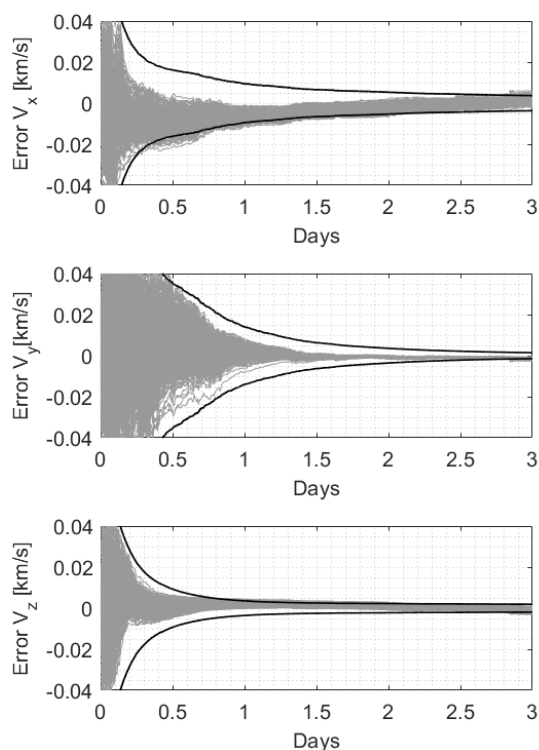


Figure 15: Velocity error and 3σ for 500 samples of the Monte Carlo simulation, CF

The estimation of the position vector in camera frame yields the error and relative 3σ bounds depicted in Figure 14. The bias that appeared in the estimation of the position component along the boresight direction has been mitigated by the introduction of the measurement bias as a Gauss-Markov process in the navigation filter. The error values are consistent with the fact that vision-based navigation strategies can provide accurate measurement in the image plane, but less precise estimations along the camera boresight. Moreover, the high uncertainties on the camera properties considered in the simulations contribute to the spreading the estimation of the position error. The 1σ errors resulting from the VBN expressed in Camera reference frame are in the order of 20 km for the two in plane directions and 80 km in the out of plane axis. From the navigation filter it is

possible to also retrieve an estimation of the spacecraft velocity. The estimation yields to the error shown in Figure 15, where the 1σ errors bounding the estimation are 0.2131 m/s, 0.1891 m/s and 0.1822 m/s respectively.

Modified OpNav Algorithm for Over-exposed Images

The outcome of edge detection on an illustrative image from the over-exposed dataset is presented in Figure 16. The blue set of points depicts the limb position extracted through the edge detection chain, while the red dots in the figure represent the edge points after correcting for optical distortion. The scene is the same as in Figure 10, and the comparison highlights the difficulty in retrieving the edge when the image is heavily overexposed.

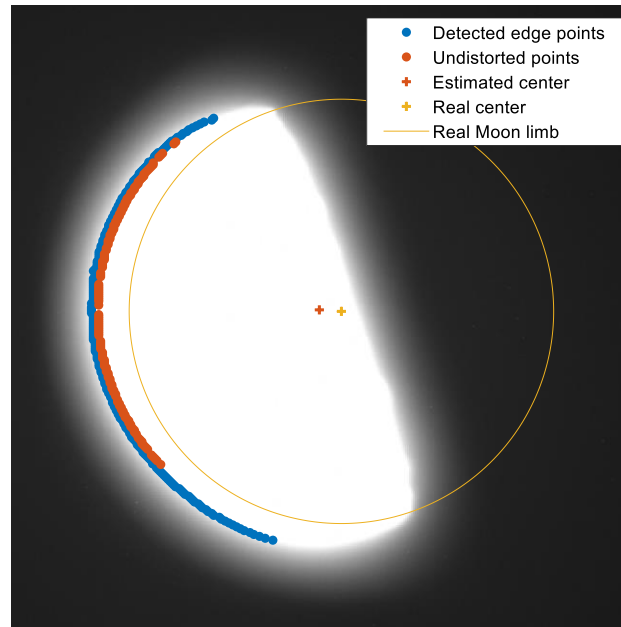


Figure 16: IP results on over-exposed image

It is interesting to highlight how the resulting edge detection performance vary along the dataset. It is worth recalling that the orbit is such that the satellite-to-Moon distance spans from 44426 km to 71609 km, while the phase angle decreases from 90 to 40 degrees. The apparent Moon radius in the image decreases from about 110 px to 70 px. The effects related to image saturation, image ghosts, and infield straylight depend on the number of photons impacting on the detector. For the considered trajectory, this number decreases with the increasing range from the Moon. The described variations are visible in the collection of images from Figure 17.

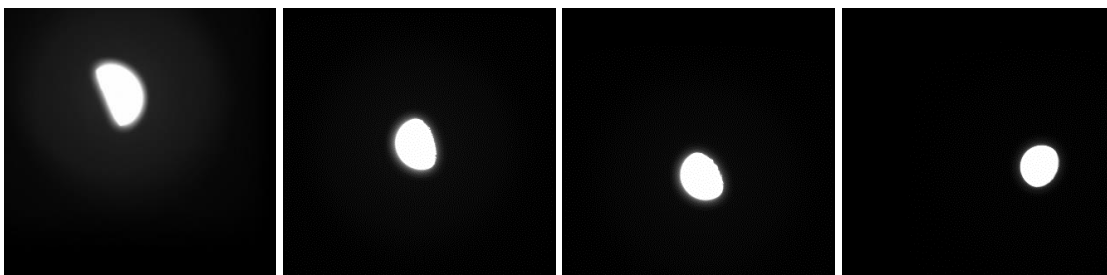


Figure 17: Images from over-exposed dataset

Figure 18 illustrates the variation in the estimation error of the Moon's radius along the timespan covered by the scenario's trajectory. The figure of merit experiences a substantial improvement after ~1 day, that corresponds to a distance to the Moon higher than 50000 km. The performance of the IP chain improves substantially at higher distances from the Moon.

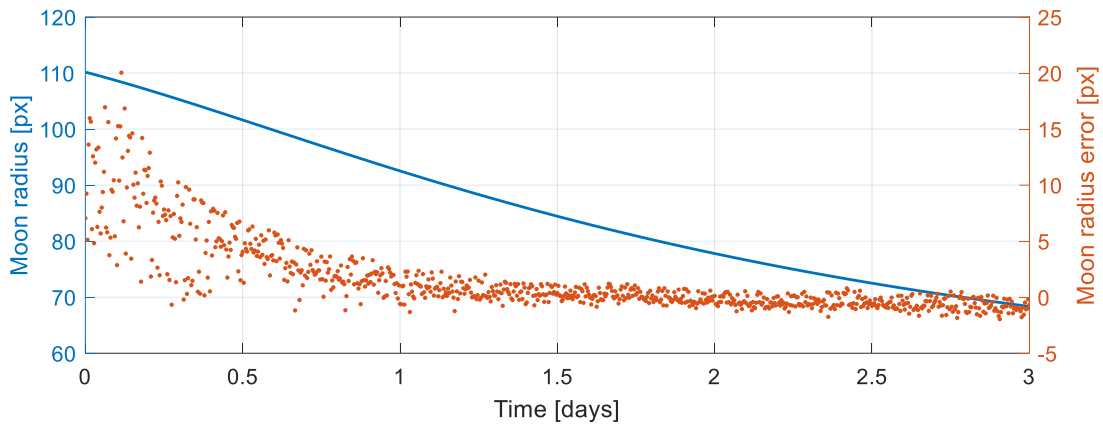


Figure 18: Comparison between estimation error of Moon's radius and its dimension

The estimation of the position of the center of the Moon in the image suffers as well from the pixel intensity spreading, but it is bounded between ± 2 px after 1 day. Figure 19 shows the resulting errors with respect to the real location of the Moon center for both the X and Y component in camera frame, expressed in PDF and CDF.

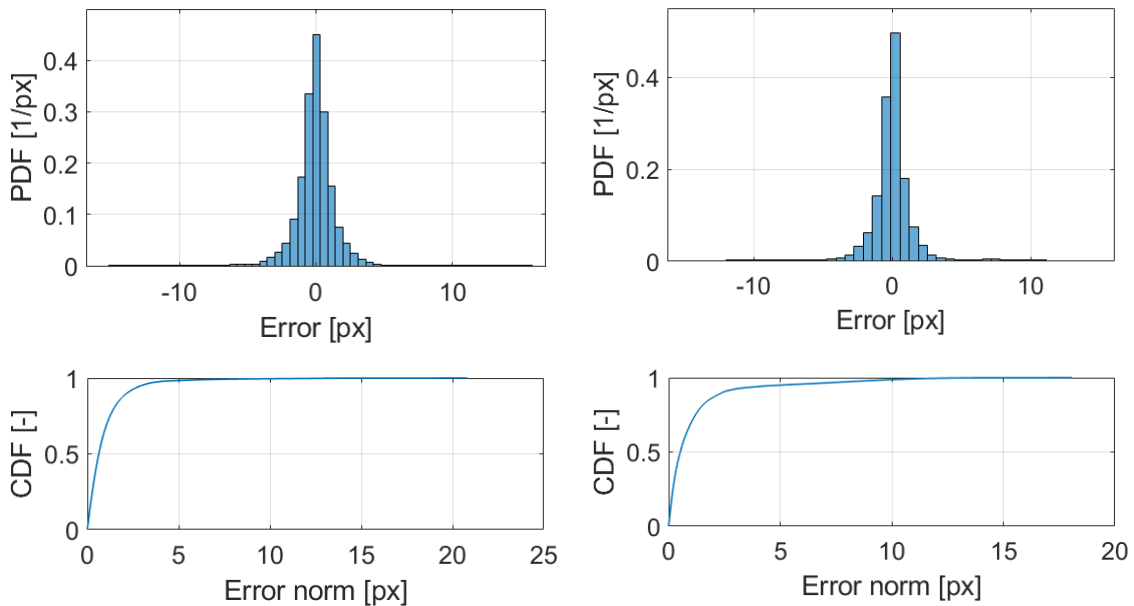


Figure 19: Estimation of circle center – overexposed dataset

The output from the image processing chain provides the position vector from the satellite to the center of the Moon, expressed in the camera reference frame and visible in Figure 20.

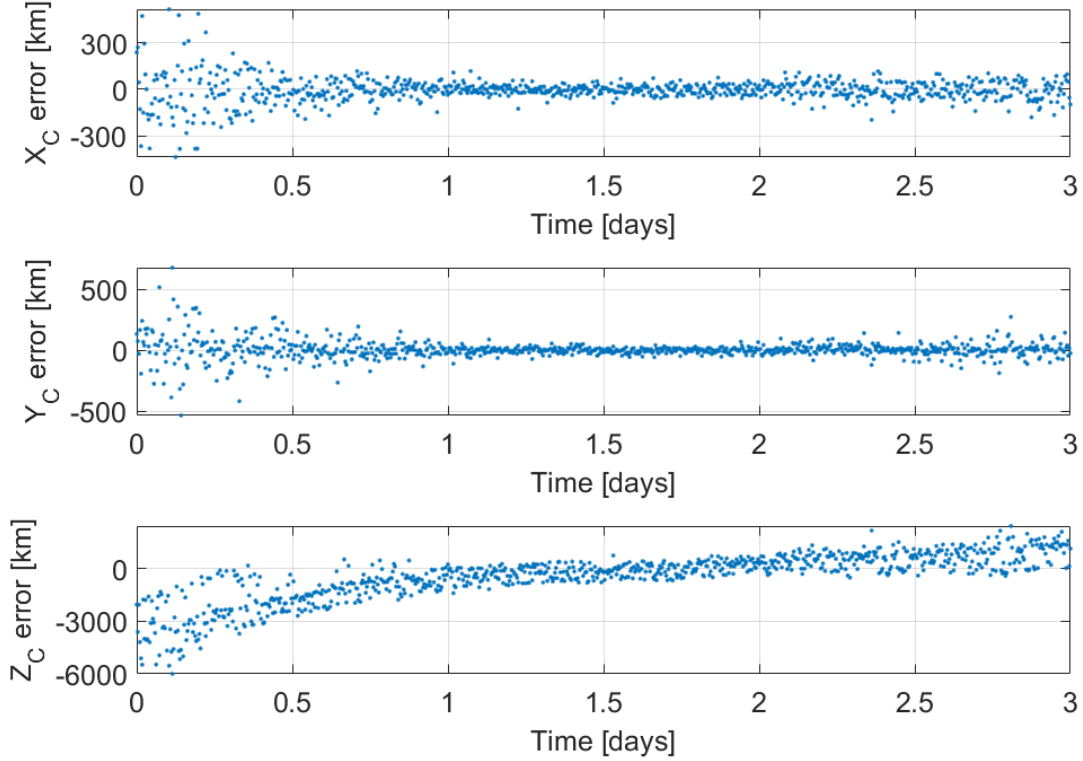


Figure 20: Estimated position vector error, overexposed dataset

The severe overestimation of the radius caused by the camera saturation effects is reflected on the underestimation of the position vector component along the boresight direction. It can also be seen that the performance improves with the increasing distance from the Moon. The two in-plane directions are retrieved with higher accuracy, as they are related to the placement of the center of the Moon in the image. However, also these two components suffer the most from the high distortions caused by the reduced range.

The position vector thus estimated is fed to the navigation filter. As for the simulations with the correctly exposed dataset, also in the case of over-exposed images the measurement covariance matrix is initialized considering a covariance value extracted from the IP results. The VBN analysis here presented considered a covariance on the edge detection of 50 px. As for the previous VBN results, the statistical performance of the EKF is assessed through a 500-samples Monte Carlo analysis with the same uncertainties values. Figure 21 presents the resulting position error extracted from the EKF, in the camera reference frame. The figure presents the evolution of the error on the three-reference frame axes for the 500 samples, as well as the related 3σ covariance bounds. Finally, Figure 22 shows the evolution of the estimated velocity error, bounded by its 3σ covariance. The 1σ errors on the position estimated from the overall OpNav strategy are now in the order of 35 km for the two in plane directions and 220 km along the boresight. The 1σ errors estimated on the velocity are instead 1.2 m/s, 0.5 m/s and 0.6 m/s. As expected, accounting for image over-exposure lowers the performance significantly.

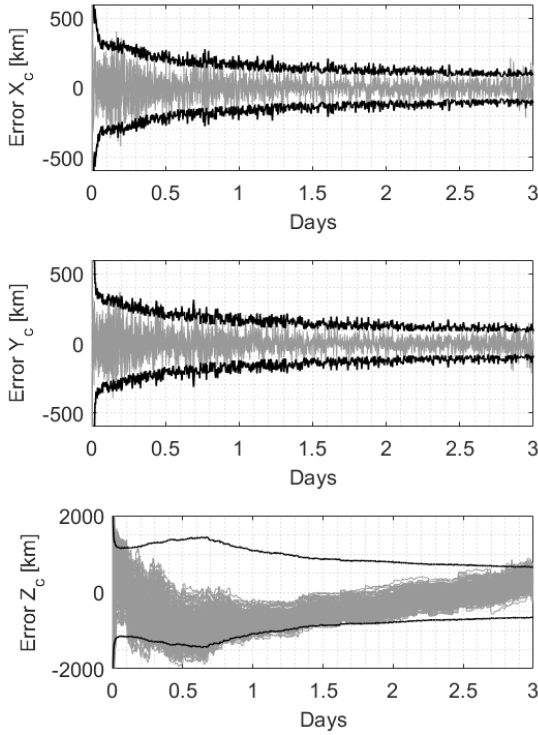


Figure 21: Position error and 3σ for 500 samples of the Monte Carlo simulation

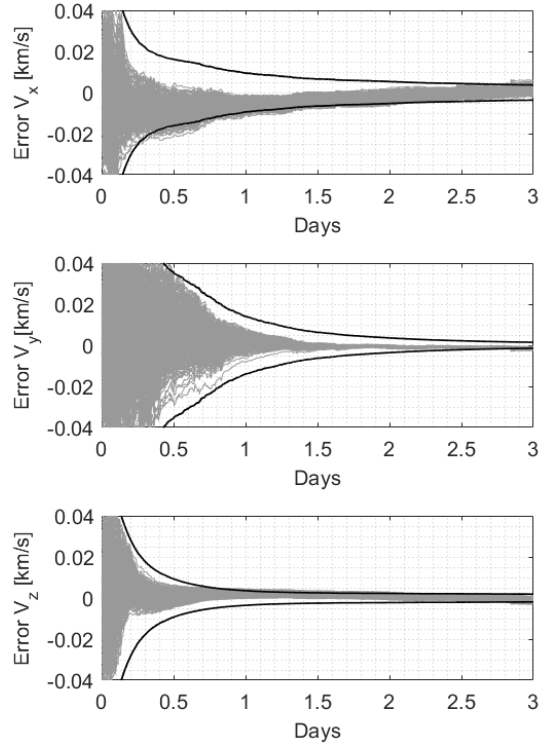


Figure 22: Velocity error and 3σ for 500 samples of the Monte Carlo simulation

CONCLUSIONS

This work presents a lunar horizon-based optical navigation algorithm that exploits star trackers in place of navigation cameras. The study is conducted within the project STAR Nav, and aims at assessing if and to what extent can STR be employed as optical navigation sensors. Hardware limitations as well as overexposure issues are considered in the design of the VBN pipeline. The image processing is constructed to extract useful information of the Moon limb from either illuminated segments or small portions of the image. The relative position from the camera to the body is retrieved and fed as a measurement to an Extended Kalman Filter. The validation of the strategy is carried out on synthetically generated renderings of the Moon, coupled with information on the nominal orbital trajectory of the satellite. In addition, the VBN chain is tested on a second dataset composed of overexposed images, showing the impact of saturation and light effects on the achievable navigation performance.

It has been highlighted that the reproduction of saturation and distortion effects are dependent on parameters and functions that are not fully known at the present moment, being the instrument involved in the process not characterized at such off-nominal conditions. Future works will focus on the analysis of the best way to mimic the star tracker behavior for different exposure times. The effects of image saturation on the Moon appearance will be studied thoroughly, as well as their consequences on the estimation of the position vector. In conclusion, possible solutions that could enhance the accuracy of the state vector estimation will be explored in depth, and results will be analyzed to better evaluate the impact of the use of STR within an optical navigation chain.

ACKNOWLEDGMENTS

This work has been conducted under ESA Contract No. 4000139932/22/NL/CRS within the General Support Technology Programme (GSTP) through the support of the national delegations of Italy (ASI). The authors would like to acknowledge the support received by Simone Becucci, Marcella Belcari, Damiano Macchi, Luca Cappabianca, Ranieri Marchi and Andrea Sica in the context of the STAR Nav study.

REFERENCES

- ¹ Jonniaux, Gregory, et al. "Autonomous vision-based navigation for JUICE." 67th International Astronautical Congress, IAC-16 A. Vol. 3. 2016
- ² P. Panicucci, F. Piccolo, S. Borgia, A. Rizza, V. Franzese, and F. Topputo, "Current Status of the LUMIO Autonomous Optical Navigation Experiment", *12th International Conference on Guidance, Navigation & Control Systems*, Sopot, Jun. 12-16 2023.
- ³ R. Inman, G. Holt, J. Christian, K. W. Smith and C. D'Souza, "Artemis I Optical Navigation System Performance", AIAA 2024-0514. AIAA SCITECH 2024 Forum. January 2024.
- ⁴ C. Giordano, C. Buonagura, A. Martinelli, G. Merisio, V. Franzese and F. Topputo. "Trajectory design and analysis of the LUMIO CubeSat", AIAA 2024-1271. AIAA SCITECH 2024 Forum. January 2024
- ⁵ D. C. Brown, "Decentering distortion of lenses". *Photogrammetric Engineering*. 32 (3): 444–462
- ⁶ J. A. Christian, "A tutorial on horizon-based optical navigation and attitude determination with space imaging systems" *IEEE Access*, 9:19819–19853, 2021.
- ⁷ H. Radim, and J. Flusser. "Numerically stable direct least squares fitting of ellipses." *Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization*. WSCG. Vol. 98. Citeseer, 1998.
- ⁸ H. Richard, and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- ⁹ D. Renshaw and J. Christian. "Subpixel localization of isolated edges and streaks in digital images." *Journal of Imaging*, 6:33, 05 2020.