VISION-BASED NAVIGATION FOR THE LUMIO CUBESAT MISSION

Paolo Panicucci[*], **Felice Piccolo**[†], **Antonio Rizza**[‡], **Gianmario Merisio**[§], **Francesco Topputo**[¶], **and Roger Walker**[‖]

The Earth-Moon system is constantly bombarded by meteoroids whose scientific interest relies in the possibility to infer information about Solar System evolution and formation. Moreover, the impact characterization is extremely crucial to understand possible threads for future human and robotic lunar missions. Nowadays impact observation is performed from the Earth, implying the impossibility of observing the lunar farside. To complement these observations, the LUMIO mission is currently under development. LUMIO is a 12U CubeSat mission to Earth-Moon L_2 studying the meteoroid impacts on the lunar farside. Besides its scientific objectives, LUMIO wants also to act a demostator for multiple critical technologies to be proven in cislunar environment. One of these technology is the autonomous vision-based navigation exploiting Moon's limbs. This paper describe the image processing and the navigation algorithm developed within the LUMIO mission. Moreover, the paper describes the image generation procedure which considers camera optics effects and detector errors. Finally, numerical results are presented for two navigation scenarios. Results show the feasibility of navigating around the Moon autonomously with less than 0.5 % of accuracy in position and 0.1 % accuracy in velocity with respect to the mean orbital motion.

INTRODUCTION

Meteoroids of different size constantly collide with the Moon surface. The study of their impact can provide insight in the Solar System formation and evolution. Although meteoroid impacts have been studied in the last 50 years to develop Solar System meteoroids models to represent their spatial spatial distribution, these models have been developed by using flashes observations of meteoroid impacts on the lunar surface as observed from the Earth. As these models are extremely important to understand the spatial distribution of the meteoroids around the Earth and to predict large and small impact on the Earth and Moon surfaces,[1] it is crucial to complement the observation taken from the Earth with some spaceborne ones. Indeed, Earth-based observations are limited by geometrical and illumination conditions which can limit the construction of meteoroids model. To overcome these limitation, a Moon-orbiting observatory could increase in quality and quantity the

[*]Assistant Professor, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milan

[†]PhD Student, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milan

[‡]PhD Student, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milan

[§]Postdoctoral Research Fellow, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milan

[¶]Full Professor, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milan

[‖]Head of the CubeSat Systems Unit, Directorate of Technology, Engineering and Quality, ESA, Keplerlaan 1, PO Box 299 NL-2200 AG Noordwijk, The Netherlands

observations used to fit meteoroid models. To answer these questions, the Lunar Meteoroid Impacts Observer (LUMIO) mission has been designed.[2] LUMIO is a CubeSat mission to a halo orbit at Earth-Moon L2 that shall observe, quantify, and characterize meteoroid impacts on the Lunar farside by detecting their flashes, complementing Earth-based observations on the Lunar nearside, to provide global information on the Lunar Meteoroid Environment and contribute to Lunar Situational Awareness.[3,4] LUMIO has successfully passed the Preliminary Design Review (PDR) at the end of Phase B under ESA funding with the support of national delegations of Italy (ASI) and Norway (NOSA). LUMIO wants not only to fulfill scientific objective, but also show the maturity multiple technologies in the cislunar environment. One of the critical technologies to be tested is the autonomous vision-based navigation experiment. This work aims at providing an insight in the LUMIO vision-based navigation experiment design and validation.

The rest of the paper is structured as follows. Section "The LUMIO mission" presents LUMIO by underlining the overall mission and the high-level spacecraft design after PDR. Section "Image Processing" describe the image processing (IP) algorithm. In Section "Navigation Fitler" the navigation filter processing IP measurement is outlined. Then, Section "Image Generation" illustrates the process used within the LUMIO mission to simulate images to assess the optical navigation experiment performance. Finally, validation simulations are presented in Section "Navigation Results" to assess the algorithm performance.

THE LUMIO MISSION

LUMIO is a CubeSat mission led by Politecnico di Milano to a halo orbit at Earth-Moon L_2. Politecnico di Milano's LUMIO mission envisages a 12U CubeSat integrated by Argotec equipped with a custom optical payload designed by Leonardo, a dedicated on-board payload data processing unit designed by S&T, a Solar Array Drive Assembly and X-band transponder developed by IMT, and the ground segment and operations in charge of Nautilus. The mission foresees the observation of the meteoroid impacts on the far side of the Moon so as to complement Earth-based observations which can only detect near side impact due to the tidally-locked motion of the Moon. The scientific payload, called LUMIO-Cam, is designed and built by Leonardo and it consists of a narrow-angle camera with two detectors providing the possibility of observing the flashes in the visible and the near-infrared spectra. This will improve the characterization of the meteoroid flux impacting the lunar surface advancing the understanding of how meteoroids evolve in cislunar space.

The LUMIO mission timeline, depicted in Figure 1, foresees a commissioning phase and a transfer phase lasting about 150 days where the spacecraft detumbles, performs the subsystems commissioning, and corrects the trajectory to target the Halo Injection Maneuver. This maneuver enables the spacecraft to insert into a halo orbit around the Earth-Moon L_2 point. The spacecraft will perform 1 year of nominal mission on the halo and it will end its life by impacting the Moon after having executed the End of Life Maneuver.

Thanks to the almost-periodic behavior of the halo orbits, the LUMIO concept of operation foresees two modes during its operative phase:[5] the Science Cycle and the Navigation & Engineering Cycle. The former lasts approximately 14 days and occurs when the Moon farside has the optimal illumination to perform flash detection (i.e., half of the Moon is not illuminated). The latter is defined as the complementary of the Science Cycle. During the Navigation & Engineering Cycle, the CubeSats cannot carry out scientific observations, therefore it can perform engineering activities, such as reaction wheel desaturation, communication with the Earth, station-keeping, and technological demonstration. A sketch of the LUMIO concept of operation is reported in Figure 2 for the sake of completeness.
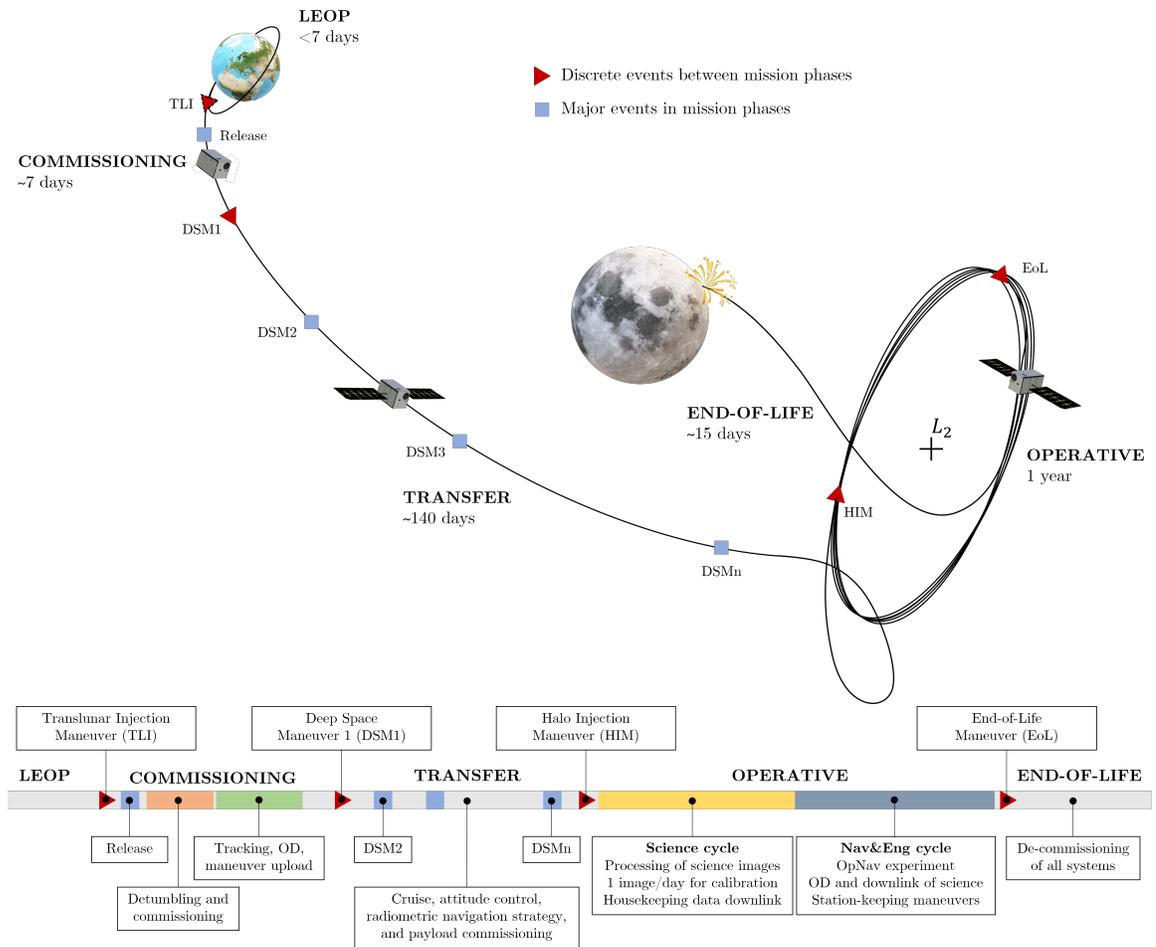
Figure 1: LUMIO mission timeline
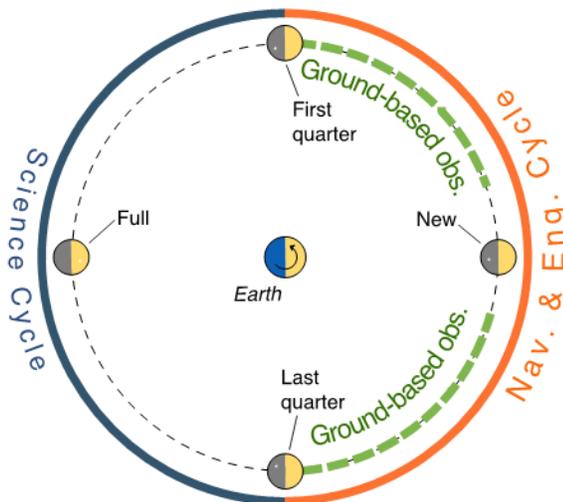
Figure 2: LUMIO concept of operation.

The LUMIO technological demonstration is the Optical Navigation Experiment (ONX).[6] The ONX aims at proving the feasibility of CubeSats to autonomously navigate in the cislunar environment without communication with the ground by exploiting images of the Moon. Indeed, the spacecraft-Moon range is computed by determining the appearance of the Moon in the image.[7–9] This piece of information is then provided to the navigation filter to determine the spacecraft state.[6] Before describing the vision-based navigation (VBN) chain, the LUMIO AOCS architecture is presented to provide information about the system design ensuring the correct spacecraft pointing and orbit.

IMAGE PROCESSING

The goal of the image processing is to determine the points belonging to the illuminated Moon limb in the image and use them to reconstruct the position vector of the Moon in the camera frame. Therefore, a dedicated image processing pipeline has been designed and implemented. As outlined by Reference,[10] the limb is a sharp feature that is convoluted with the camera Point Spread Function (PSF), leading to difficulties in reconstructing the exact limb location. Therefore, a sequential approach inspired by Reference[11] has been implemented for the LUMIO mission. The proposed approach allows the determination of the limb at subpixel precision and the rejection of possible outliers. The approach is divided into several steps.

Image Scanning Procedure

First, by knowing the Sun direction from sun sensors or ephemerides, the projection of the Sun rays can be determined. The main idea is to scan the image in this direction to reconstruct the image intensity evolution along this scanning direction to detect the first rapid increase in pixel intensity. This could be associated with the Moon lit limb. As noted in Reference,[11] it is important to determine only the directly illuminated limb even for low phase angles to avoid systematic errors in the vision-based navigation solution. Moreover, it is worth noting that the projection of the Sun direction could be deduced directly from the image as proposed in Reference,[7] but numerical investigations have shown that the algorithm proposed in Reference[7]'s work is not well-conditioned when the phase angle is close to zero, leading to unstable solution in determining the lit limb.

By analyzing the time profiling in detail, it is noted that the most time-consuming operation is to rotate pixel coordinates to compute the intensity line. This takes about the 90% of the computational time. Therefore, it is decided to develop a simplified scanning procedure to decrease the computational time of the overall IP and to enable higher IP measurement frequency, if deemed necessary in future phases. It is important to note that the developed simplified procedure was developed by ensuring that the overall performance of the image processing is not affected by this algorithmic change. The main reasoning behind the simplified scanning procedure developed is that it is not necessary to obtain scan the image in projection of the incoming Sun direction, as outliers in edge detection are discarded by the RANSAC algorithm. Therefore, it is decided to scan the image in the direction of image axes (i.e., image rows and columns) to avoid any rotation on the pixel coordinates. This removes completely the rotation calculation and simplifies the scanning procedure as intensity lines are simply allocated by extracting the line or column in the image. The developed procedure works as follows:

1. The direction of the incoming light is projected on the image plane to compute the projection of the incoming light.

2. A logical check is performed to understand if the intensity lines are the image lines, the image columns, or both. This step also computes the direction of the scanning in the image (i.e., top-to-bottom or bottom-to-top for the columns and left-to-right or right-to-left for the lines). The logical check is performed as shown in Figure 3. If the projection of the Sun incoming light (i.e., the yellow arrow in Figure 3) belongs to the yellow angular interval in Figure 3, the scanning direction are the blue arrows in in Figure 3. The number of scanned lines is an algorithm input parameter, and lines are equally spaced in the image to obtain the desired number of intensity lines.

Limb determination on the scanning line

It is necessary to determine where the limb is along the scanned line. The limb is theoretically identified as a sharp change in intensity. Moreover, after the sharp change, the intensity stabilizes at a quite high value compared to the dark background. It is important not to select bright pixels due to cosmic rays, stars, or hot pixels when performing the coarse estimation of a limb point. Therefore, the coarse edge location is determined as the first pixel of a given interval of pixels whose intensity values are above a user-defined threshold τ_{thr}. The threshold τ_{thr} strongly depends on the SNR obtained during flight and calibration. This enables removing localized intensity variations in the scanning line which are not due to the limb. In the following analyses it is set to 20 DN over 255 DN.

Then, a subpixel estimation of the limb location must be determined to obtain an accurate position estimation for the navigation filter. Indeed, the accuracy of the image processing position solution strongly relies on the accuracy in limb point determination, implying that a subpixel solution leads to a better measurement given to the filter. Therefore, a sequential refinement of the coarse solution is performed Reference.[11] A patch of N_{patch} pixels is extracted from the image around the selected coarse limb point and the Laplacian of Gaussian filter is used to determine a finer estimation of the limb location. The Laplacian of Gaussian is preferred over other edge-finding techniques as it searches for zero crossing of the second derivatives without any thresholding mechanism. The subpixel location of the limb is determined by using the Zernike-moment approach,[11] which is preferred for its simplicity in implementation over more complex algorithms like the partial-area-effect method.[12] The proposed and implemented method is tested on synthetic images of the Moon determining the limb point location with subpixel accuracy.

Finally, outlier rejection is performed by using the RANSAC (RANdom SAmple Consensus) algorithm[13] by fitting a circle on the subpixel limb points. The circle fitting is selected as the Moon flattening is 0.0012, leading to almost a perfect circle when projected in the image. Moreover, this simplifies the calculation and the error computation as the error between a point $\boldsymbol{P}$ to the circle limb is computed as:

$$\epsilon = ||\boldsymbol{P} - \boldsymbol{C}|| - r \tag{1}$$

where $\boldsymbol{C}$ is the fit circle center and r is the fit circle radius.

Position Determination

Once the inliers are identified, the limb points can be used to compute the optical observables. The optical observables depend on the implemented image processing algorithm (e.g., planet centroid and apparent diameter[8] or position vector[14]). For the LUMIO mission, it has been decided to
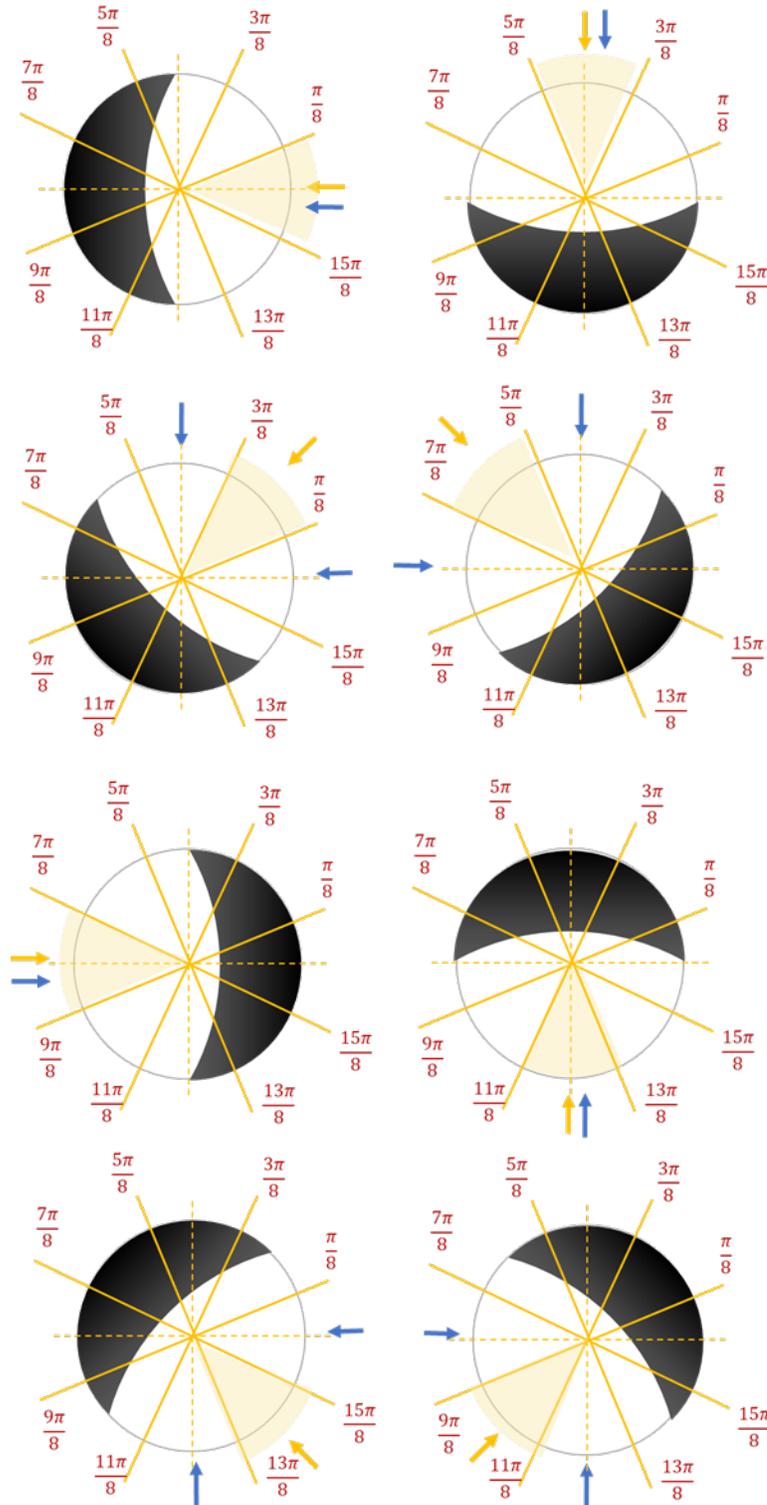
Figure 3: Simplified scanning procedure exploited in the LUMIO image processing. The yellow arrow is associated with the Sun incoming ray direction, the yellow area is the valid interval for the scanning direction associated with a scanning direction reported in blue.

directly estimate the position vector within the image processing pipeline to simplify the navigation filter implementation. The Moon projection is estimated by fitting a circle to the observed lit limb points, which are retrieved from 2D images as previously outlined. Let $[D]$ be the matrix containing on the diagonal the semimajor axes of the Moon ellipsoid, let $[PC]$ be the rotation matrix from the camera reference frame to the Moon principal axes reference frame, and let $[K]$ be the intrinsic camera matrix. The position estimation is estimated as follows:[11]

$$r = - \left([D]\,[PC]\right)^{-1} \left(n^T n - 1\right)^{-\frac{1}{2}} n \tag{2}$$

where r is the camera-to-object position vector in the camera reference frame and n the solution to the least-squares problem in the Cholesky decomposition space fitting the limb point location on the Moon ellipsoid (see Reference 14 for more details). Let $k = (f_x,\ f_y,\ c_x,\ c_y,\ s)$ be the vector containing the camera calibration matrix $[K]$ elements, where f_x and f_y are the focal lengths in pixel, c_x and c_y are the camera center coordinates in pixels, and s is the camera skew. The covariance of the estimated position $[P_{rr}]$ is then computed by considering the uncertainty on the limb point location, on the inertial attitude knowledge error, and on the camera calibration matrix:

$$[P_{rr}] = [N]\,[P_{nn}]\,[N]^T + [Q]\,[P_{q_v q_v}]\,[Q]^T \tag{3}$$

where $[P_{nn}]$ is the covariance on vector n arising from the least square problem,[14] $[P_{q_v q_v}]$ is the covariance on inertial attitude knowledge error, and $[N]$ and $[Q]$ are the derivatives of the estimated position vector with respect to the limb point locations and the inertial attitude knowledge error. In detail:

$$[N] = \left(n^T n - 1\right)^{-\frac{1}{2}} [PC]^T\,[D]^{-1} \left([I_{3\times3}] - \left(n^T n\right) / \left(\left(n^T n - 1\right)\right)\right) \tag{4}$$

$$[Q] = 2\left(n^T n - 1\right)^{-\frac{1}{2}} [PC]^T \left[\left([D]^{-1}\,n\right)^\times\right] \tag{5}$$

where $[I_{3\times3}]$ is the 3x3 identity matrix and $[(\cdot)^\times]$ is the skew-symmetric matrix associated with the cross product. The covariance $[P_{nn}]$ is computed as:[14]

$$[P_{nn}] = \left([Y]\,[P_{yy}]^{-1}\,[Y]^T\right) \tag{6}$$

where $[P_{yy}]^{-1}$ is the inverse of the covariance matrix associated with the post fit error residual of the total least square problem. By assuming that $[P_{yy}]$ is diagonal, it is possible to compute analytically $[P_{yy}]^{-1}$:

$$[P_{yy}]^{-1} = \left([J]\,[D]\,[PC]\left([K]^{-1}\,[P_{hP\,hP}]\,[K]^{-T} + [\Pi]\,[P_{kk}]\,[\Pi]^T\right)[PC]^T\,[D]^T\,[J]^T\right)^{-1} \tag{7}$$

where $_hP$ is the limb point P in homogeneous coordinates, $[P_{hP\,hP}]$ is the covariance on $_hP$, $[P_{yy}]_{ii}^{-1}$ is the ith diagonal element of $[P_{yy}]^{-1}$, $[P_{kk}]$ is the covariance on the camera calibration vector k. Note that $[J]$ and $[\Pi]$, reported here for the sake of brevity, are the matrices used to perform the mapping between the post-fit residuals and the homogeneous points errors and calibration errors. Moreover, the covariances $[P_{hP\,hP}]$, $[P_{kk}]$, and $[P_{q_v q_v}]$ are expressed for simplicity as:

$$[P_{hP\,hP}] = \mathrm{diag}\left(\sigma_{\mathrm{edge}}^2,\ \sigma_{\mathrm{edge}}^2,\ 0\right) \tag{8}$$

$$[P_{kk}] = \mathrm{diag}\left(\sigma_{f_x}^2,\ \sigma_{f_y}^2,\ \sigma_{c_x}^2,\ \sigma_{c_y}^2,\ 0\right) \tag{9}$$

$$[P_{q_v q_v}] = \sigma_{\mathrm{att}}^2\,[I_{3\times3}] \tag{10}$$

where the standard deviations are reported in Table 1.

Parameter Name	Parameter	Value
Edge location variance	σ_{edge}	0.2 pixels
Focal length x-axis variance	σ_{f_x}	0.5 pixels
Focal length y-axis variance	σ_{f_y}	0.5 pixels
Optical center x-axis variance	σ_{c_x}	0.5 pixels
Optical center y-axis variance	σ_{c_y}	0.5 pixels
Attitude knowledge variance	σ_{att}	10 arcsec

Table 1: Variances considered in the algorithm for the LUMIO mission

NAVIGATION FITLER

The output of the IP algorithm is the camera-to-Moon position estimation in the camera reference frame, yet no information on spacecraft velocity is available. An extended Kalman filter (EKF) is then used to estimate the spacecraft state (position and velocity), and to increase the navigation accuracy.

The on-board EKF estimates the state of the spacecraft in a J2000 reference frame centered in the Earth-Moon barycenter. The dynamic model used in the EKF accounts for the gravitational attraction of the Moon, the Earth and the Sun. The equations of motion are integrated using a Runge-Kutta 4 integration scheme. The state transition matrix (STM) is also computed integrating the variational equations along with the equations of motion.

To account for uncertainties in the dynamics, residual accelerations are included in the state vector as Gauss-Markov processes. Furthermore, three additional GM processes are used to model IP measurement biases. In total the state vector and the STM are expanded to include 6 additional elements (3 for residual accelerations and 3 for IP biases). The statistical properties of the modeled GM processes are reported in Table 2. IP measurements are currently processed once every 5

	Time constant	Standard deviation
Residual acceleration	$[1,\ 1,\ 1]^T$ days	$[5, 5, 5]^T \times 10^{-9}$ km/s^2
IP measurement bias	$[0.2,\ 0.2,\ 0.2]^T$ days	$[1,\ 1,\ 100]^T$ km

Table 2: Statistical properties of GM processes modeled in the EKF

minutes. The measurement model $\boldsymbol{h}(\boldsymbol{x})$ is given by:

$$\boldsymbol{h}(\boldsymbol{x}) = -\left[CN\right]\left(\boldsymbol{r} - \boldsymbol{r}_{Moon}\right) + \boldsymbol{b}_{IP} \tag{11}$$

$$[H] = \begin{bmatrix} -\left[CN\right] & \left[\mathbb{0}_{3\times3}\right] & \left[\mathbb{0}_{3\times3}\right] & \left[I_{3\times3}\right] \end{bmatrix} \tag{12}$$

where $\left[\mathbb{0}_{3\times3}\right]$ is a 3x3 matrix of zeros, $\boldsymbol{r}$ and $\boldsymbol{r}_{Moon}$ are the position vectors of LUMIO and the Moon, respectively, $\boldsymbol{b}_{IP}$ is the EKF-estimated IP bias, and $[CN]$ is the rotation matrix from the J2000 frame to the LUMIO camera frame and. The latter is computed as:

$$[CN] = [CB]\,[BN] \tag{13}$$

in which $[BN]$ is the rotation matrix from the J2000 frame $\mathcal{N}$ to the LUMIO body frame $\mathcal{B}$ and $[CB]$ is the rotation matrix from the LUMIO body frame $\mathcal{B}$ to the LUMIO camera frame $\mathcal{C}$.

The measurement covariance matrix is obtained by summing the contributions due to the measurement error of the IP and the attitude estimation error. Thus:

$$[R] = [P_{\boldsymbol{r}\boldsymbol{r}}] + [R]_{\text{att}} \tag{14}$$

Note that $[P_{rr}]$ is given by Equation 3 and $[R]_{\text{att}}$ is computed by taking the derivatives of Equation 11 with respect to the rotation from $\mathcal{N}$ to $\mathcal{C}$. By using the quaternions to ease the calculations, $[R]_{\text{att}}$ is expressed as:

$$[R]_{\text{att}} = \left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}_{\mathcal{C}/\mathcal{N}}}\right] [P_{\boldsymbol{q}_v \boldsymbol{q}_v}] \left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}_{\mathcal{C}/\mathcal{N}}}\right]^T \tag{15}$$

where $\boldsymbol{q}_{\mathcal{C}/\mathcal{N}}$ is the quaternion from $\mathcal{N}$ to $\mathcal{C}$, $[P_{\boldsymbol{q}_v \boldsymbol{q}_v}]$ is the quaternion covariance matrix, and

$$\left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}_{\mathcal{C}/\mathcal{N}}}\right] = 2\left[\left([CN]^{-1}\boldsymbol{r}\right)^{\times}\right] \tag{16}$$

To reject possible incorrect measurements provided by the IP algorithm a measurement editing procedure is introduced. Define the innovation, $\delta\boldsymbol{r}$, and its theoretical covariance (i.e., $[S]$) as:

$$\delta\boldsymbol{r} = \bar{\boldsymbol{r}} - \boldsymbol{h}(\boldsymbol{x}) \tag{17}$$

where $\bar{\boldsymbol{r}}$ is the IP measurement. According to the EKF hypotheses, the innovation should behave as a white noise with covariance $[S]$. Therefore, the quantity:

$$m^2 = \delta\boldsymbol{r}^T [S] \delta\boldsymbol{r} \tag{18}$$

which is the squared Mahalanobis distance of the innovation, should behave as a χ^2 distribution with degrees of freedom equal to the number of elements in $\delta\boldsymbol{r}$. Thus, measurements whose innovation is unexpectedly large can be rejected by comparing the value of m^2 to a threshold m_t that can be derived according to the χ^2 statistics. Currently, in the EKF the threshold is set to $m_t = 11.3449$, which means rejecting measurements whose probability is lower than 1%. If $m^2 < m_t$ the Mahalanobis distance test is passed, and the EKF update step is executed. Otherwise, the test is failed, and the EKF update is skipped.

IMAGE GENERATION

The rendering procedure is reported in Figure 4. The core of the rendering procedure is the image generation performed in CORTO,[15] an open-source rendering software developed at Politecnico di Milano's DART Lab. The output of rendering engine is an error-free image respecting the observational geometry and the light-celestial-body interaction as defined by the celestial object characteristics. As the image is error-free, a post-generation step is added to the rendering pipeline to account for the effect of the camera optics and detector on the images. This process is performed by two successive steps:

1. The image is convoluted with the Point Spread Function (PSF) to account for the non-perfect focusing of the optical system;

2. The image is corrupted by relevant noises induced by the detector.

In order to test the image processing algorithm outlined in Section "", it is necessary to generate images of the Moon on the LUMIO operational orbit. To do so, a simulator has been implemented in Blender (see https://www.blender.org/, last accessed 2nd February 2023). Blender can render images with path tracing by simulating the behavior of light, materials, and sensors. The user must define the position and orientation of the objects in the scene (e.g., the camera, the Sun, and the Moon) and the scattering law between the incident light on the object surfaces.
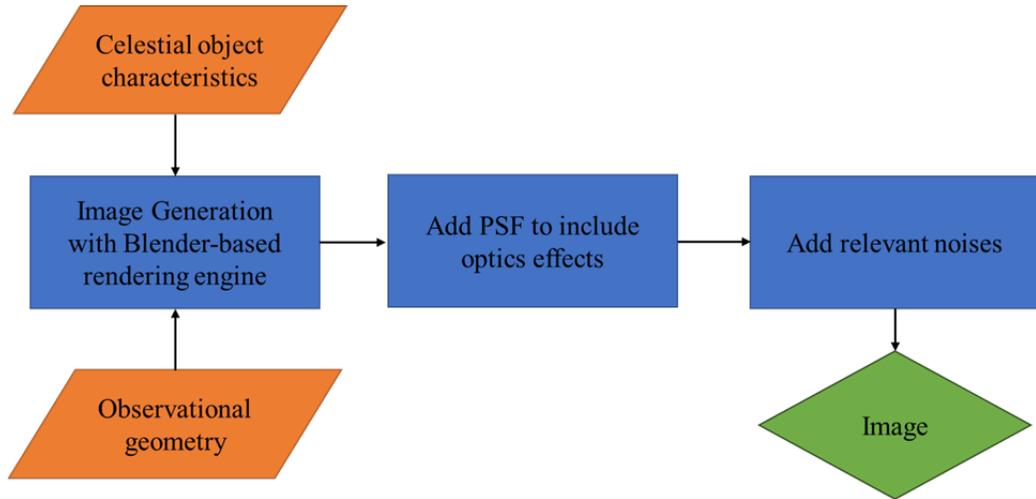
Figure 4: The rendering procedure overview
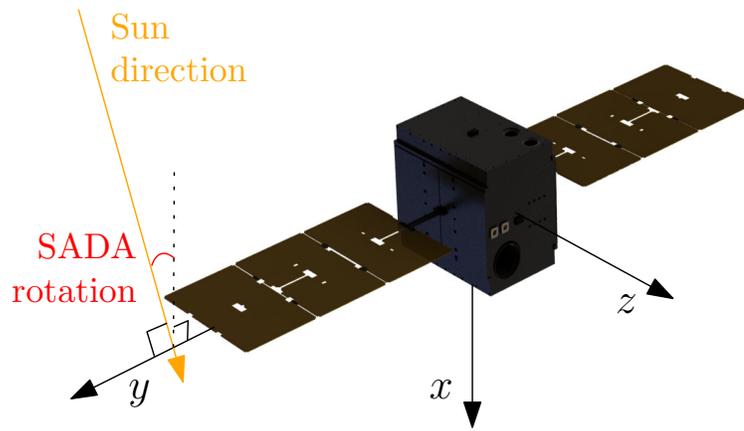
Figure 5: LUMIO pointing definition (Credits: Argotec)

Noiseless Image Generation

In order to test the image processing algorithm, it is necessary to generate images of the Moon on the LUMIO operational orbit. To do so, the CORTO simulation environment is exploited. The images are simulated by exploiting SPICE kernels of the LUMIO mission to gather positions of LUMIO, the Moon, and the Sun. The LUMIO pointing definition, shown in Figure 5, foresees the primary pointing towards the desired celestial body depending on the mission phase. Moreover, the secondary pointing must ensure that the Solar Array Drive Assembly axis, i.e., the axis around which the solar panels rotate, is perpendicular to the Sun direction. This reference pointing definition implies that the Sun moves on the x-z plane keeping its projection on the x-axis positive. Therefore, the Sun light as seen from the LUMIO-Cam arrives from a constant direction (i.e., the image x-axis).

10

Point Spread Function

The Point Spread Function (PSF) refers to the mathematical representation of how an optical system transforms a point source into an image on the detector. Indeed, in an ideal camera system light from a point source would be focused onto a single pixel on the sensor; however, in real-world cameras the light path is modified by several effects, such as diffraction and optical aberrations, causing the light to spread out over multiple pixels on the sensor. The current considered PSF is a 7 x 7 kernel with a defocus of 1 pixel.

Detector Noise Model

Detector noises are added to the image during the conversion from photon to electron and from electron to DN. As the detector errors are usually expressed in electrons, the image content is first converted from DN to electrons. Then, noises are computed thanks to performance model provided by the LUMIO-Cam manufacturer. Considered noises are: photon shot noise, photo response non-uniformity, dark signal non-uniformity, and readout noise. these are applied as reported in Reference 16.

Finally, it is considered to have a random number of hot pixels within the image. To account for this effect, a random integer n_{hot} is sampled between 0 and 200. This is the number of hot pixels in the image. Then n_{hot} random x and y image coordinates are generated to distribute the pixel in the image and their image content is saturated.

Figure 6a, Figure 6b, Figure 6c, and Figure 6d show the effect of the overall noise model applied to a test image from the dataset. Figure 7 shows a detail of the rendering image where hot pixels are highlighted.

IMAGE PROCESSING RESULTS

The proposed image processing is tested on synthetic images of the Moon spanning the first Navigation & Engineering (Nav&Eng) cycle of the LUMIO mission to assess the algorithm performance and its robustness to errors.

The performance of the algorithm after outlier rejection are reported in Figure 3. The limb points are determined with 0.2 pixels of standard deviation which is consistent with the setting provided to the RANSAC algorithm. The mean of the inliers is slightly biased, leading to a systematic error in the position solution provided to the filter (see also Section "Navigation Results" for more details). To understand the performance of the image processing algorithm, the image processing is tested over a dataset of images generated with the procedure in Section "Image Generation" covering the first Nav&Eng cycle (i.e., about 15 days).

The error of the proposed image processing pipeline is reported in Figure 9a and Figure 9b in terms of fitted circle radius error and center error on the image plane for the noisy image dataset. Note that the circle radius error can be directly linked with the error in the camera z-axis and the circle center error can be mapped to the error on the camera x- and y-axes. Moreover, Figure 10 shows the evolution of the camera position error along with the evolution of the phase angle and the Moon's apparent diameter. It is worth noting that the image processing is not affected by illumination conditions as an illuminated limb is always visible in the image, as also noted by Reference 10. Moreover, the performance of the image processing is not affected by errors in the Sun direction as outlined in numerical investigations not reported here for the sake of brevity. The image processing performance is consistent with results from the literature affected by the Moon's apparent diameter. This
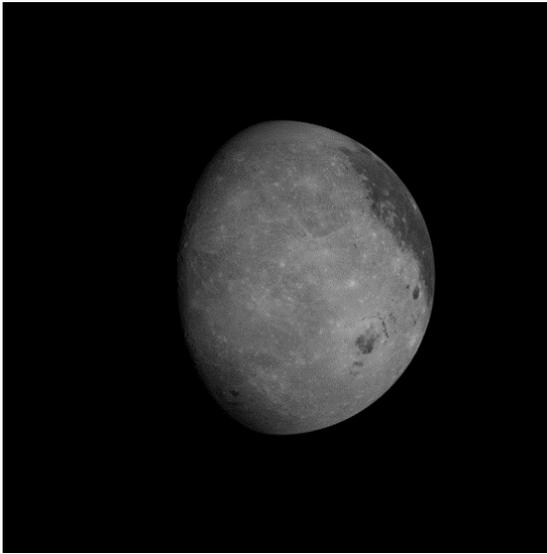
(a) Noiseless image
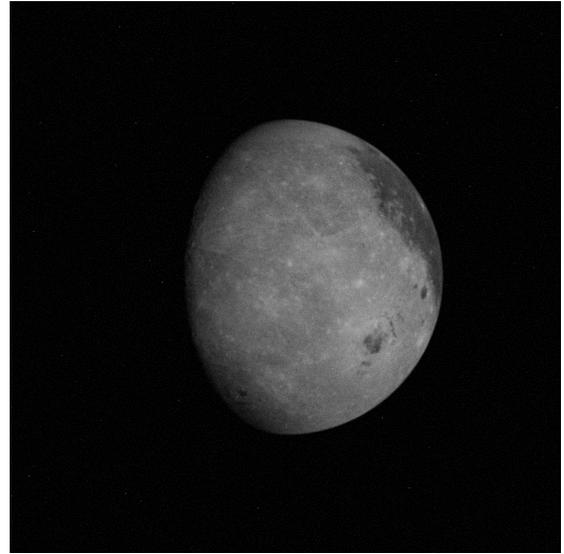
(b) Noisy image
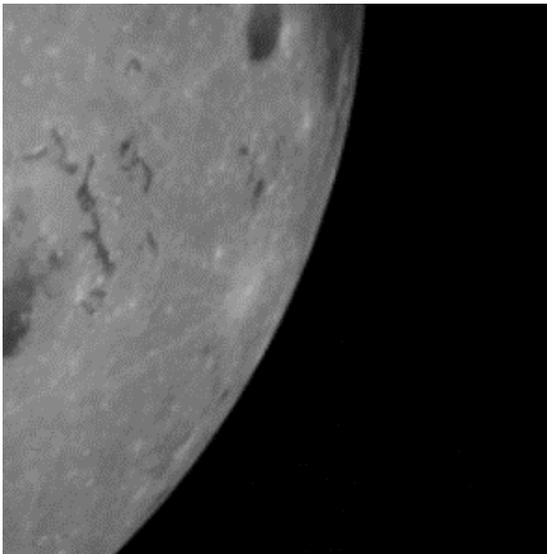
(c) Detail of the noiseless image
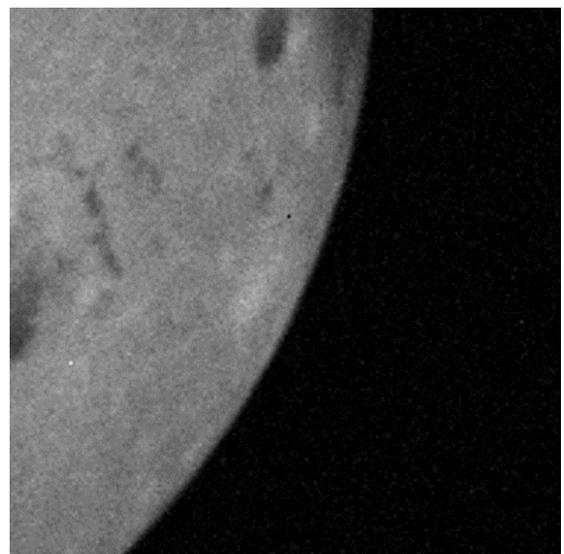
(d) Detail of the noisy image

Figure 6: Comparison between noiseless and noisy images.

Figure 7: A detail of the render image showing hot pixels

is because a smaller Moon implies that fewer scanning lines are impacting the illuminated limb. The position error is below 200 km for most of the cases in the camera boresight direction, whereas it is below 10 km for the in-plane directions. Note that the error on the x-axis is higher as the Sun rays are coming from the positive axis direction. This leads to constraining the position estimation with higher accuracy on the y-axis and having a high correlation between the x-axis and the z-axis as shown in Figure 10.

NAVIGATION RESULTS

The proposed numerical simulations investigate the performance of the VBN chain in two different scenarios. In both scenarios, the images are rendered with the reference attitude on the reference trajectory to obtain a reference dataset to be used in all the simulations (see Section "Image Generation"). Then the following parameters are perturbed to consider relevant errors of the VBN scenario:

1. Initial position and velocity are distributed as a Gaussian distribution with zero mean and a standard deviation of 30 km and 0.1 km/s respectively.

2. Inertial attitude knowledge is distributed as a Gaussian distribution with zero mean and zero mean and a standard deviation of 10 arcseconds.

3. Sun direction as estimated by the Sun Sensors to be used for the image processing pipeline is distributed as a Gaussian distribution with zero mean and a standard deviation of 5 degrees.
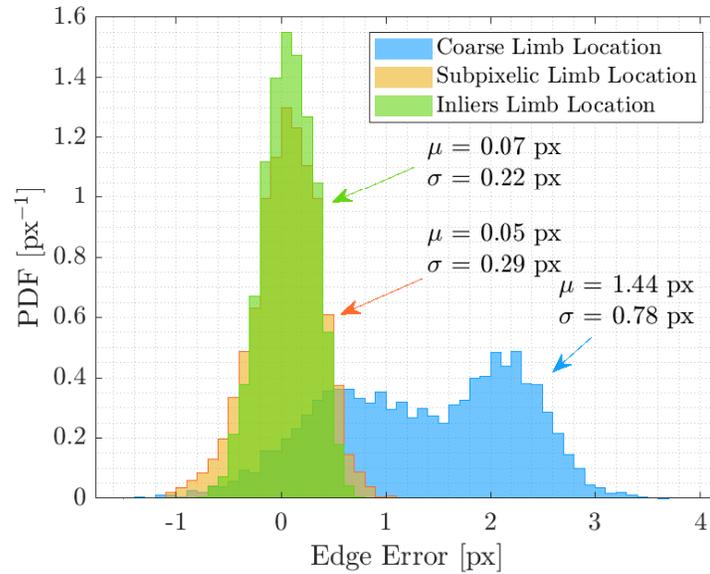
Figure 8: Error of the detected limb points at different stages of the IP

4. Camera calibration matrix to consider residual errors in in-flight calibration. It is distributed as a Gaussian distribution with zero mean and a standard deviation of 0.5 pixels on the focal lengths and the camera center. The skew is considered zero and perfectly known as per standard camera.[13]

In the following analyses images are processed sequentially every 300 seconds and the image processing algorithm provides the position estimate and its covariance to the navigation algorithm that determines the spacecraft states. To have a preliminary assessment of the statistical performance of the navigation filter, a Monte Carlo simulation with 1000 samples is performed.

As no detailed scheduling of the LUMIO mission is currently available, it is not determined yet where the optical navigation will take place on the orbit. To show the versatility and the performance of the VBN experiment, two scenarios are analyzed capturing different observational and dynamical characteristics on the operational orbit.

In the first scenario, the first three days of the Nav&Eng cycle are simulated to study the VBN chain results when the moon occupies the majority of the field of view. This implies that more points are detected when scanning the image searching for the Moon limbs, leading to a more accurate estimation of the spacecraft position with respect to the Moon. Moreover, the spacecraft orbits closer to the moon when the dynamics is less linear, leading to a worse performance of the navigation filter.

In the second scenario, the furthest positions in the first Nav&Eng cycle (i.e., from 6 to 9 days) are simulated to understand the performance of the algorithm when the Moon is smaller in the image. This leads to detecting fewer points, implying less accurate and precise image processing measurements.

Scenario 1: Closest range to the Moon

The results in terms of position and velocity in the camera frame for the Monte Carlo simulation of the first scenario are reported in Figure 11a and Figure 11b. The filter converges rapidly despite

(a) Error on the estimated circle center for the IP (b) Error on the estimated circle radius for the IP
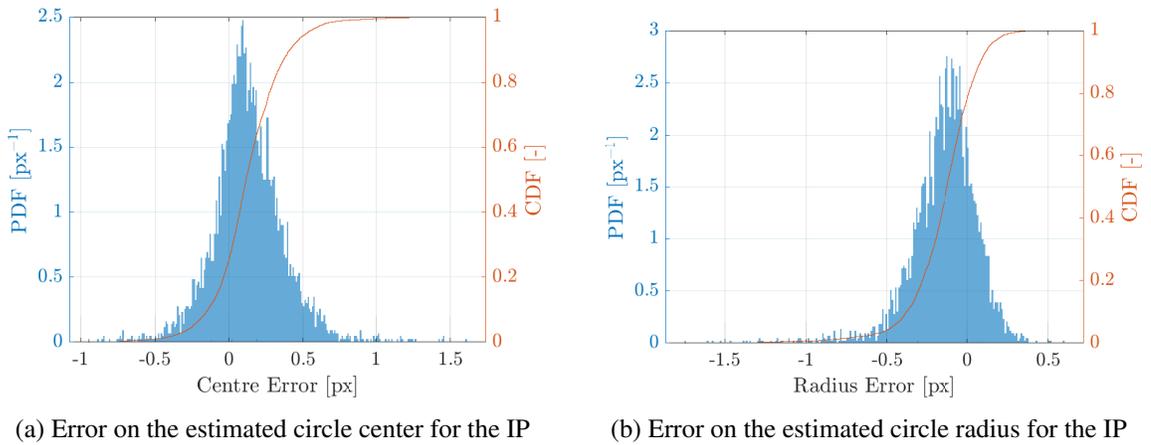
Figure 9: Error on the circle fitted by the RANSAC algorithm
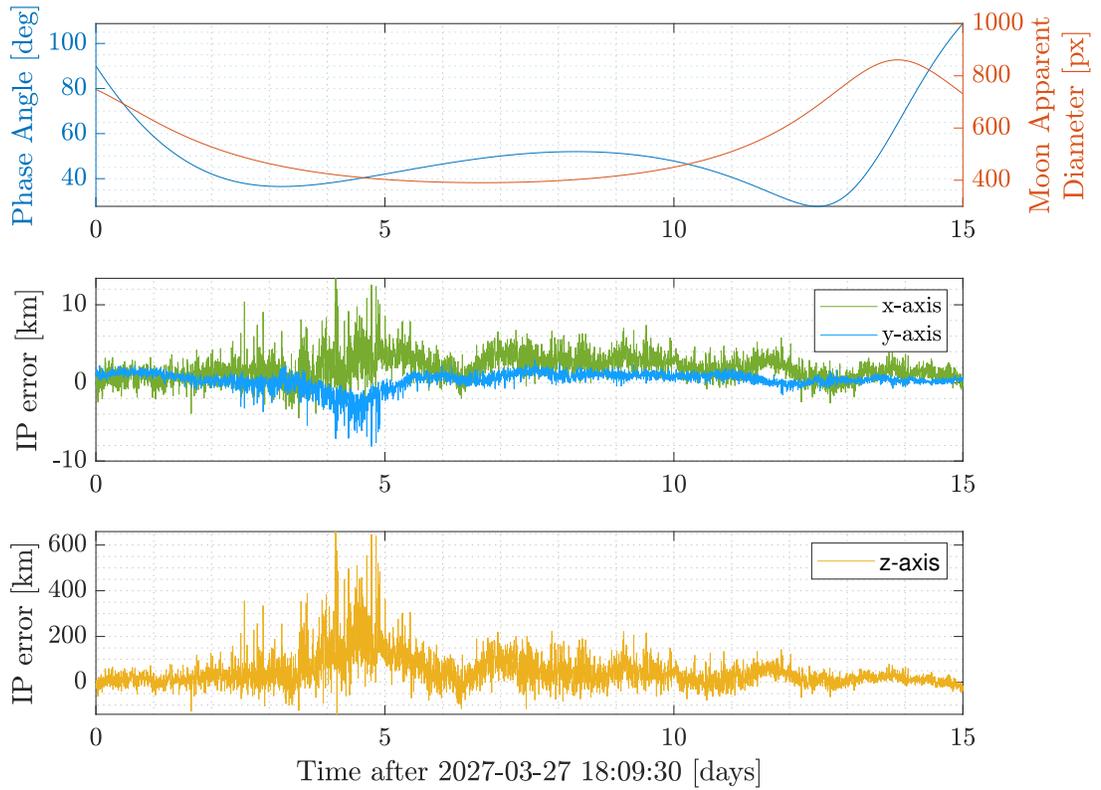
Figure 10: The IP with the Simplified Scanning Procedure error on the Cartesian axes for the first Nav&Eng cycle. In the first plot, the evolution of the phase angle and the Moon apparent diameter is reported as well.

the spacecraft is orbiting in the more non-linear part of the orbit. This is due to the accuracy and precision of the image processing measurement (see Figure 9a, Figure 9b, and Figure 10) which show low variability with respect to the true position. As mentioned before, this is because a high percentage of lines scanning the image impact the Moon limbs, leading to a higher number of points to determine the position. Moreover, as the most informative points to constrain the conic fitting are the ones with the highest curvature, the probability to detect these points is higher, leading to a more precise solution from the image processing. The position is estimated with an error below 100 km for the z-axis and 10 km for the x-axis and y-axis. Moreover, the covariance bounds are below 200 km for the z-axis and 20 km for the x-axis and y-axis. This is consistent with vision-based navigation techniques that can provide accurate measurement in the image plane, while less precise orbit determination on the camera boresight. The velocity is well determined as well with an error below 1 m/s for the z-axis and 0.5 m/s m for the x-axis and y-axis. the velocity covariance bound converges to about 4 m/s for the z-axis and 1.5 m/s m for the x-axis and y-axis.

Scenario 2: Furthest range to the Moon

The results for the second scenario are presented in Figure 12a and Figure 12b, showing the position and the velocity of the spacecraft with respect to the Moon in the camera frame. The filter results show worse performance than the previous case mainly for the lower accuracy and precision of the image processing measurements (see Figure 9a, Figure 9b, and Figure 10). The introduction of the measurement bias as a Gauss-Markov process tries to mitigate this effect by estimating the IP systematic error from the measurement time series, but the observability of this systematic error is limited by the errors in the dynamics modeling in the filter and the IP random errors. Nevertheless, the effect of the systematic error is counteracted by the bias estimation performed in the filter, leading to a lower systematic error in the filtered results. The position is estimated with an error below 200 km for the z-axis and 10 km for the x-axis and y-axis. Moreover, the covariance bounds are below 200 km for the z-axis and 20 km for the x-axis and y-axis. It is possible to notice that a systematic error is still present in the position estimate despite the slight correction obtained by the modeled bias. The systematic error is mainly present on the z- and x-axis because they are strongly correlated due to the constant light direction (see Figure 10 for correlation in the IP measurements). It is important to note that the bias in the position is mainly due to the subpixel determination algorithm which is strongly dependent on the camera's optical response. In future phases of the mission, a better characterization of the optical payload could lead to design a dedicated subpixel filter to limit this effect. The systematic error is also visible on the z-axis velocity even though the estimation converges to 1 m/s on each axis. The velocity covariance bounds stabilize to 2 m/s for the z- and x-axis due to the measurement correlation and 0.5 m/s m for the y-axis. It is worth noting the slight systematic error in the velocity estimation due to the inaccuracy of the image processing measurement.

CONCLUSION

This paper outlines the status of the design and development of vision based navigation technological demonstration for the LUMIO mission. This work focuses on the image processing and the navigation filter design, with particular attention to reduce the computational burden due to the image processing. The image processing is able to efficiently detect point belonging to the Moon limb with subpixel precision and then determine an estimate of the LUMIO-Moon relative position. The image processing results provide position determination at km-level precision which is
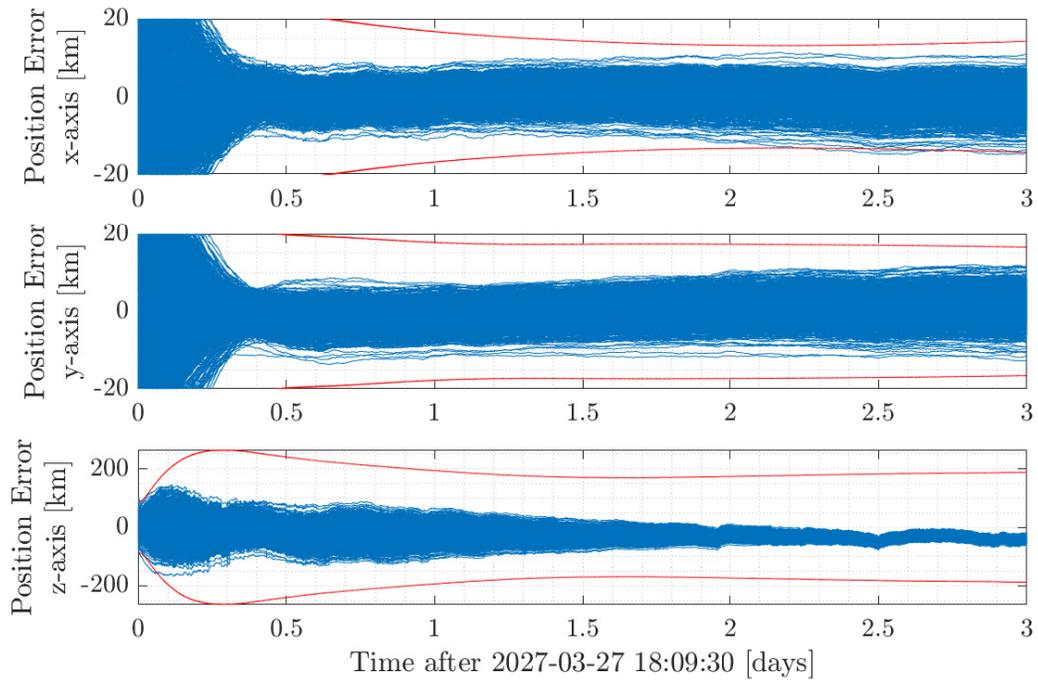
(a) Position error
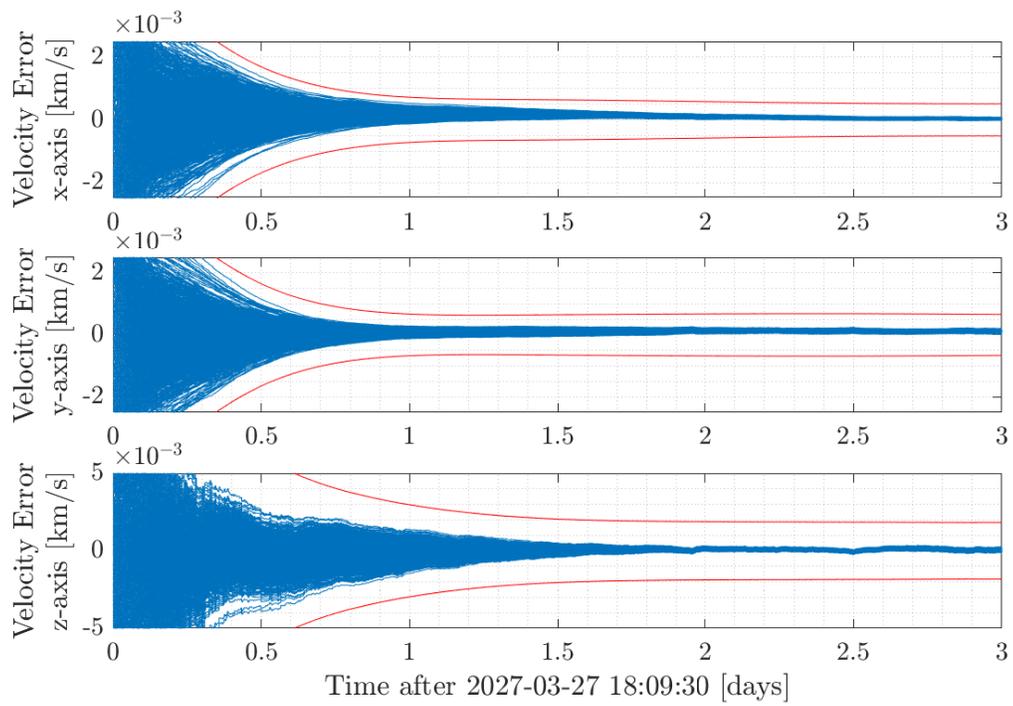
(b) Velocity error

Figure 11: The Monte Carlo results in the camera frame for the first three days after the beginning of the first Nav&Eng Cycle
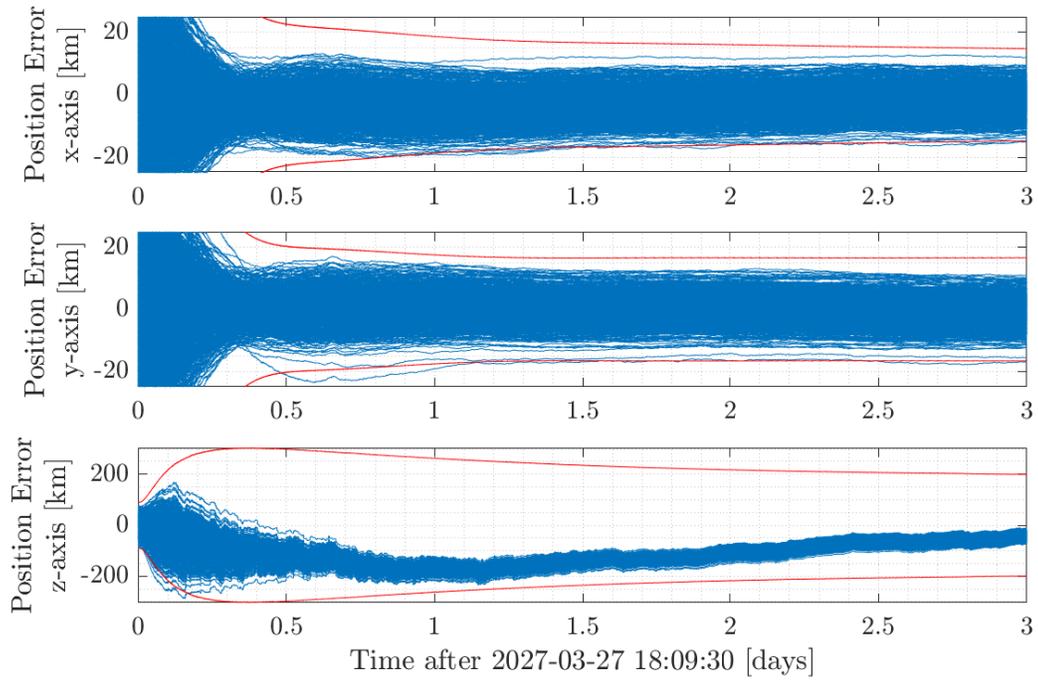
(a) Position error
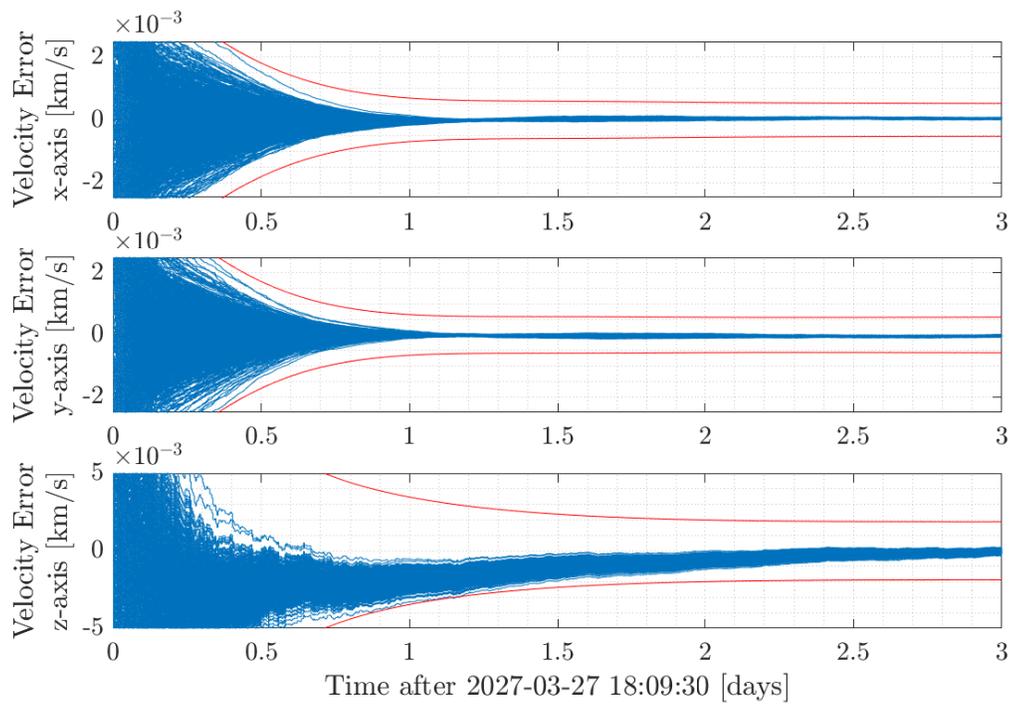
(b) Velocity error

Figure 12: The Monte Carlo results in the camera frame from the 6th day to the 9th day after the beginning of the first Nav&Eng Cycle

then fed to the navigation filter. The navigation filter is designed to smooth the solution to gather the full spacecraft state. The filter include the determination of dynamics and measurement biases to increase the robustness to unmodelled dynamics and to image processing systemac errors (e.g., camera calibration errors and limb point errors). Numerical investigation show that the filter can determine the LUMIO state with less than 0.5 % of accuracy in position and 0.1 % accuracy in velocity with respect to the mean orbital motion.

Future work will investigate the performance of the algorithm in processor- and camera-in-the-loop tests at DART Lab hardware-in-the-loop facilities[17–19] in order to understand the performance degradation with real camera noises and the algorithm execution time. Finally, further numerical investigation are foreseen to assess the algorithm performance in accordance with the final development of the LUMIO Cam and its characterization (e.g., distortion and calibration error characterization).

ACKNOWLEDGMENT

REFERENCES

[1] Z. Ceplecha, J. Borovička, W. G. Elford, D. O. ReVelle, R. L. Hawkes, V. Porubčan, and M. Šimek, "Meteor phenomena and bodies," *Space Science Reviews*, Vol. 84, No. 3, 1998, pp. 327–471.

[2] A. Cervone, F. Topputo, S. Speretta, A. Menicucci, E. Turan, P. Di Lizia, M. Massari, V. Franzese, C. Giordano, G. Merisio, D. Labate, G. Pilato, E. Costa, E. Bertels, A. Thorvaldsen, A. Kukharenka, J. Vennekens, and R. Walker, "LUMIO: A CubeSat for observing and characterizing micro-meteoroid impacts on the Lunar far side," *Acta Astronautica*, Vol. 195, 2022, pp. 309–317.

[3] F. Topputo, G. Merisio, V. Franzese, C. Giordano, M. Massari, G. Pilato, D. Labate, A. Cervone, S. Speretta, A. Menicucci, E. Turan, E. Bertels, J. Vennekens, R. Walker, and D. Koschny, "Meteoroids detection with the LUMIO lunar CubeSat," *Icarus*, Vol. 389, 2023, p. 115213.

[4] G. Merisio and F. Topputo, "Present-day model of lunar meteoroids and their impact flashes for LUMIO mission," *Icarus*, Vol. 389, 2023, p. 115180.

[5] A. M. Cipriano, D. A. Dei Tos, and F. Topputo, "Orbit design for LUMIO: The lunar meteoroid impacts observer," *Frontiers in Astronomy and Space Sciences*, Vol. 5, 2018, p. 29.

[6] V. Franzese, P. Di Lizia, and F. Topputo, "Autonomous optical navigation for the lunar meteoroid impacts observer," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 7, 2019, pp. 1579–1586.

[7] D. Mortari, C. N. D'Souza, and R. Zanetti, "Image processing of illuminated ellipsoid," *Journal of Spacecraft and Rockets*, Vol. 53, No. 3, 2016, pp. 448–456.

[8] J. A. Christian, "Optical navigation using planet's centroid and apparent diameter in image," *Journal of guidance, control, and dynamics*, Vol. 38, No. 2, 2015, pp. 192–204.

[9] J. A. Christian, "A tutorial on horizon-based optical navigation and attitude determination with space imaging systems," *IEEE Access*, Vol. 9, 2021, pp. 19819–19853.

[10] G. Jonniaux, K. Kanani, P. Regnier, and D. Gherardi, "Autonomous vision-based navigation for JUICE," *67th International Astronautical Congress, IAC-16 A*, Vol. 3, 2016, p. 5.

[11] J. A. Christian, "Accurate planetary limb localization for image-based spacecraft navigation," *Journal of Spacecraft and Rockets*, Vol. 54, No. 3, 2017, pp. 708–730.

[12] A. Trujillo-Pino, K. Krissian, M. Alemán-Flores, and D. Santana-Cedrés, "Accurate subpixel edge location based on partial area effect," *Image and Vision Computing*, Vol. 31, No. 1, 2013, pp. 72–90.

[13] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[14] J. A. Christian and S. B. Robinson, "Noniterative Horizon-Based optical navigation by cholesky factorization," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 12, 2016, pp. 2757–2765.

[15] M. Pugliatti, C. Buonagura, and F. Topputo, "CORTO: The Celestial Object Rendering TOol at DART Lab," *Sensors*, Vol. 23, No. 23, 2023, p. 9595.

[16] R. D. Gow, D. Renshaw, K. Findlater, L. Grant, S. J. McLeod, J. Hart, and R. L. Nicol, "A comprehensive tool for modeling CMOS image-sensor-noise performance," *IEEE Transactions on Electron Devices*, Vol. 54, No. 6, 2007, pp. 1321–1329.

[17] F. Ornati, P. Panicucci, E. Andreis, and F. Topputo, "RETINA: a highly-versatile optical facility for camera-in-the-loop testing of spaceborne Vision-Based Sensors," *46th Annual AAS Guidance, Navigation and Control Conference*, 2024.

[18] P. Panicucci and F. Topputo, "The TinyV3RSE Hardware-in-the-Loop Vision-Based Navigation Facility," *Sensors*, Vol. 22, No. 23, 2022, p. 9333.

[19] M. Pugliatti, V. Franzese, P. Panicucci, and F. Topputo, "TINYV3RSE: The DART Vision-Based Navigation Test-bench," *AIAA Scitech 2022 Forum*, San Diego, CA & Virtual, January 2022, p. 1193.