Acta Astronautica 243 (2026) 237–250

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research paper

Autonomous optical flow-based initial orbit determination

Willem Oliveira

GNC, DLR Department of Guidance Navigation and Control, Institute of Space Systems, Robert-Hooke-Str. 7, 28359, Bremen, Germany

ARTICLE INFO

Keywords:
Optical navigation
Orbit estimation
Orbit determination
Optical flow
Pinpoint landing
Heading measurements

ABSTRACT

The navigation algorithms proposed in the literature to achieve pinpoint landing for space exploration missions require initialization with state estimates from ground. This limits the missions that can be accomplished and significantly increases operation costs. This paper introduces an orbit determination (OD) algorithm, which can be used to autonomously initialize existing terrain absolute navigation (TAN) algorithms with pinpoint landing accuracy. Visual data is collected over the course of two orbit periods. This data is used to estimate the Keplerian orbit parameters of the trajectory. The estimated orbit parameters are used to initialize a crater-based TAN system independently from ground telemetry, thus providing a fully autonomous navigation system with pinpoint landing capability.

An extensive test campaign was performed using both simulated measurements and rendered images. The algorithm was shown to successfully enable TAN initialization.

1. Introduction

In the context of space exploration, achieving pinpoint landing is highly desirable. This capability requires accurate absolute navigation. In addition, autonomous methods eliminate the dependency on ground station availability, reduce costs and widen the range of possible missions. The literature suggests that optical navigation is the most suitable method to achieve both the necessary accuracy and the autonomy [1–3].

Optical navigation methods can be classified as Terrain-relative Absolute Navigation (TAN) or Terrain-relative Relative Navigation (TRN). TAN methods rely on a map or catalog of landmarks, typically prepared on ground. By matching the landmarks detected in the images captured by the on-board camera to the prepared map, an absolute state is derived. However, these methods require an initial state estimate due to the size of the maps involved. On the other hand, TRN methods function by comparing consecutive images. The apparent movement from one captured image to the next is used to derive the movement of the camera with respect to the surface (egomotion). As such, TRN methods are fully autonomous, but only provide a relative state.

The current paper introduces a novel optical navigation method with the purpose of autonomously initializing existing TAN methods. This method only requires visual camera images and orientation measurements. The camera images are used to obtain Optic Flow (OF) maps. Each OF map, associated to a pair of images, is used to estimate the translation direction between the camera states in which those images were captured.

This process is repeated for two orbit periods, after which the estimated translation direction vectors and the associated timestamps are batch processed to estimate the Kepler orbit parameters of the trajectory. The estimated orbit parameters are then used to predict the camera states for the next orbit period. Finally, these predicted states are used to initialize TAN method chosen as an example, Crater Navigation (CNav).

1.1. State-of-the-art

At the start of the space age, Earth-bound operations were essential. Since then, spacecraft autonomy has been gradually increasing. In 1965, Gates and Gordon [2] state that extra-terrestrial precise landing cannot be achieved with Earth-based radio guidance. Although the method they propose requires on ground processing, it is based on optical measurements captured on-board.

The first of these sensors to fly in a mission was the Far Encounter Planet Sensor (FEPS) on board of the Mariner VI and VII missions [4]. By finding the brightness center of Mars, the FEPS was used as a position sensor [5], with a navigation error of 700 km. Additional on ground processing of the images reduced the navigation error to 50 km [6], via manual limb detection.

Limb detection methods use the observed illuminated limb of a target body to estimate the spacecraft's state. These methods are most appropriate at large distances from the target body, from which a larger portion of the limb can be seen and the variation in terrain

---

E-mail address: willem.magalhaesoliveira@dlr.de.

https://doi.org/10.1016/j.actaastro.2026.01.054
Received 14 May 2025; Received in revised form 8 January 2026; Accepted 24 January 2026

is not significant. Since Mariner VII, the ground-processing of limb detection algorithms was automated. It was used for Mariner IX [7]; during the Uranus and Neptune encounters of Voyager II [8,9]; the New Horizons Pluto flyby [10]; and the Kuiper-Belt Extended Mission [11]. A very simple limb detection algorithm was flown during the Deep Space 1 mission, the first mission to achieve fully autonomous navigation [12]. The target-relative delivery accuracy at encounters was found to be around 3 km 1-σ [13]. Other optical navigation methods include star-horizon, star occultation, optical flow and landmark tracking. Star-horizon methods measure the angles of specific stars with respect to the horizon of the target planet. This method was used by the crew of Gemini XII [14], some Apollo missions, including Apollo 13 [15], and the Skylab Program [16], in the form of hand-held sextants.

Despite being proven effective, automating this method is challenging due to the different exposure levels required to properly detect stars and the planet's limb. Additionally, as was observed during the Apollo 13 mission, in case debris are created from a structural failure, stars can become hard to detect, making the method unusable for the most crucial scenarios [15].

Star occultation methods are based on the measurement of the time at which stars rise above or set behind the target body or, alternatively, the apparent height of a star with respect to the planet's horizon [17]. These methods have various problems, including atmospheric effects and the inability to detect faint stars due to the necessary exposure.

The method proposed in this paper is based on OF measurements. Methods based on OF determine navigation information from the changes observed between two images. In [18], Dyblenko introduces an Orbit Determination (OD) method using a series of OF measurements. The translation vectors between sections of consecutive images are derived from an Optical Fourier Processor. From multiple optical flow measurements, the orbit parameters, apart from the right ascension of ascending node, can be determined through a filter. This method, however, has significant limitations, namely a requirement of initial position with an error up to 300 km; a limited working range of altitude, orbit eccentricity, and of a few degrees around the nadir-pointing for the camera attitude; as well as the inability to work on near equatorial orbits. The method proposed in this paper, successfully eliminates these limitations. This is possible by using camera images, instead of the Optical Fourier Processor; by accounting for the perspective effect when calculating the translation direction vector from the OF map; and by using a Least Squares (LSQ) batch processing method to estimate the orbit parameters, instead of a filter.

In [19], Christian introduces an OD method using only heading measurements obtained from OF maps. The heading measurements are simulated, assuming a point mass gravity model and with added noise of up to 1°. In practice, these heading measurements would be obtained from images captured on board and Inertial Measurement Unit (IMU) measurements by a method such as the one described in [20]. First, features are detected in consecutive images and matched. The matched features provide an OF map, which can be used to obtain the relative motion of the spacecraft via a LSQ formulation and a Random Sample Consensus (RANSAC) scheme to eliminate outliers.

In the current paper, the heading measurements are not simulated, but obtained from either simulated OF maps or rendered images. As with the method provided in [20], a LSQ formulation relating the relative motion with the epipole is used, as well as a RANSAC scheme to detect outliers. A different method is used to account for the effect of attitude change in the OF maps: in [20], attitude measurements are included in the LSQ formulation used to determine the heading direction; whereas the heading measurements used in the current paper were obtained by first predicting the rotation component of the OF maps, and then applying LSQ to the isolated translation component. That said, the OD method described in [19], as well as the one proposed in the current paper, are applicable regardless of the method used to measure the heading direction. As such, and since the scope of this paper is the OD method, rather than the measurement of the heading direction, further details on this topic are not provided.

The OD method described in [19] is based on the orbital hodograph, which is the locus of points defined by the tip of the spacecraft's velocity vectors. Although resulting in a successful OD for orbits simulated using the point mass gravity model, this method was only tested for heading errors of up to 1°, which is below the range observed when using measurement methods such as the one provided in [20].

Instead of the orbital hodograph, the algorithm proposed in this paper estimates the orbit inclination and the longitude of the ascending node via the angular momentum vector, which is estimated using linear LSQ. The remaining orbit parameters are estimated via a non-linear LSQ formulation, which relates them to the angle between the velocity vector and the longitude of the ascending node. RANSAC is used to eliminate outliers at each estimation step. Although the estimation assumes a point mass gravity model, it was tested for orbits simulated using the GRAIL (Gravity Recovery And Interior Laboratory) gravity model of degree and order 660.

Finally, landmark based methods work by matching landmarks detected in captured images with a map or catalog, which contains their 3D georeferenced coordinates. These landmarks can be either surface features such as craters and rocks, or image features such as the ones identified by feature detectors (*e.g.* SIFT and AKAZE). Thanks to the available map or catalog, the terrain-referenced 3D state of the spacecraft can be estimated, which is essential for pinpoint landing, here defined as landing within 100 m of the planned landing point. Some examples are: the crater-based algorithm developed by Leroy et al. [21]; a more recent crater-based method developed by Maass et al. [22,23]; and the terrain matching method proposed by Johnson et al. [24].

The first landing to autonomously use optical data in the Guidance, Navigation and Control (GNC) loop was NASA's Mars Exploration Rovers. However, precision landing was not achieved, since the Descent Image Motion Estimation System (DIMES) only provided velocity estimation to account for wind during landing [25].

On October 2020, the OSIRIS-Rex mission performed a sample collection and return from the surface of the asteroid Bennu. Preliminary testing prompted the development of an optical navigation system named Natural Feature Tracking (NFT) [26]. This system uses a textured shape model of the asteroid, the estimated state of the spacecraft and the sun position to render a set of small images. These images are matched to the image captured by the camera to determine a new state estimate.

On February 2021, NASA's Mars 2020 mission successfully landed, using an improved version of the Entry Descent and Landing sequence of the 2012 Mars Science Laboratory (MSL) mission. One of the improvements was the addition of the Lander Vision System (LVS), an optical landmark based TAN system [27], which reduced the position error to less than 40 m at 2 km altitude [28,29]. As with MSL, the initial state estimate is obtained from ground stations.

At the time of writing, the missions which have been confirmed to achieve precision landings in the context of space exploration were Apollo, Hayabusa, Smart Lander for Investigating Moon (SLIM). Apollo's and Hayabusa's precision landing was only possible thanks to the human detection of craters and features of the terrain [30,31]. SLIM was the first mission to achieve the 100 m landing accuracy autonomously [32]. The navigation method used was based on the detection of craters in images, which were then matched to a reference crater map. However, the initial value of the spacecraft's position was obtained by propagating OD results obtained from ground stations on Earth. Intuitive Machine, Inc. claims its lunar lander, Athena, landed within 250 m of its planned landing site [33]. To the author's knowledge, this claim has not yet been independently corroborated. Although precision landing was not achieved, the Chandrayaan lunar landing missions have used optical navigation [34]. Finally, the Chang'e landing missions have used optical sensors as part of their

Hazard Detections and Avoidance (HDA) system [35]. Currently, there is no information on the planned landing sites for the Chang'e missions.

To date, no precision landing on a large extraterrestrial body has been achieved without navigation from ground. The algorithm introduced in this paper aims at eliminating this requirement.

1.2. Paper structure

In this paper a new method for on-board autonomous OD based on OF is introduced and validated. This algorithm is envisioned as part of the navigation system for a landing mission on an extra-terrestrial body. It shall provide the state estimate required to initialize the already available and more precise TAN methods, which currently require this initial knowledge to be sent from ground stations. The current research focuses on the lunar case.

In Section 2, the relationships between translation direction vectors and orbit parameters are derived. In Section 3, the LSQ methods are obtained from these relationships. Section 4 describes the tests made and results obtained. Finally, the conclusions are drawn in Section 5.

2. Theoretical framework

In the absence of noise and for Keplerian orbits, four measurements are sufficient to obtain a measurement. Only two measurements are necessary to determine the orbit's inclination and longitude of ascending node. Using these orbit parameters, the measurements are transformed into the perifocal frame. Since there is no magnitude information, the measurements can be described as the angle with the periapsis direction. Finally, to determine the remaining four orbit parameters, four angle measurements are necessary, including from the two measurements used to determine the inclination and longitude of the ascending node.

In practice, due to noise and the presence of measurement outliers, a larger sample size is required. Additionally, for eccentric orbits, due to the inability to obtain OF maps while flying over the dark side of the Moon, the accurate estimation of the mean motion, requires measurements to be obtained for more than one orbit's period. In general, no more than two orbit periods are required. As such, perturbations are neglected and the Keplerian elements will be used for the OD. These parameters are: the inclination, i, the longitude of ascending node, Ω, the semi-major axis, a, the eccentricity, e, and the initial mean anomaly, M_0. To simplify the estimation process, the orbit parameters are divided into two groups: first, the orbit plane orientation parameters, i and Ω (Section 2.1); second, the ellipse shape parameters, a, e, ω and M_0 (Section 2.2).

2.1. Orbit plane orientation parameters from translation direction

For a Kepler orbit, any translation vector, $\Delta\boldsymbol{r}_{t_i,t_j}$, between two points in the trajectory will belong to the orbit plane, as will its normalized counterpart, $\Delta\hat{\boldsymbol{r}}_{t_i,t_j}$, which will be referred as the translation direction vector. As a consequence, the normal vector $\hat{\mathbf{h}}$ can be calculated as

$$\hat{\mathbf{h}} = \Delta\hat{\boldsymbol{r}}_{t_0,t_1} \times \Delta\hat{\boldsymbol{r}}_{t_2,t_3}, \qquad t_2 > t_1, t_3 < t_0 + \frac{T}{2} \tag{1}$$

where T is the orbit period. The conditions on the time intervals ensure that the direction of the normal vector is determined correctly.

Once the normal vector is known, the inclination i and the longitude of the ascending node Ω are calculated with

$$i = \arccos\left(\hat{\mathbf{h}} \cdot \left[0, 0, 1\right]^T\right) \tag{2}$$

$$\Omega = \begin{cases} \arccos \dfrac{\hat{N}_1}{|\hat{\mathbf{N}}|}, & \hat{N}_2 \geq 0 \\ 2\pi - \arccos \dfrac{\hat{N}_1}{|\hat{\mathbf{N}}|}, & \hat{N}_2 < 0 \end{cases} \tag{3}$$

$$\hat{\mathbf{N}} = \left[0, 0, 1\right]^T \times \hat{\mathbf{h}}, \tag{4}$$

where $\hat{N}_j$ is the jth coordinate of $\hat{\mathbf{N}}$.

2.2. Ellipse shape parameters from translation direction

To simplify the problem of determining the ellipse shape parameters, a new reference frame is defined, such that $\mathbf{X}_o$ coincides with $\hat{\mathbf{N}}$ and $\mathbf{Z}_o$ with $\hat{\mathbf{h}}$ (see Fig. 1). In other words, the XY_o-plane coincides with the orbit plane and $\mathbf{X}_o$ points to the ascending node. The XY_o-plane also contains all translation direction vectors, meaning the z_o-coordinate of said vectors equals 0, and the problem can now be treated in 2D.

The rotation of vectors from the Planet Centered Inertial (PCI)-frame to the O-frame is given by

$$\Delta\hat{\mathbf{r}}^{O} = R^{O}_{PCI}\,\Delta\hat{\mathbf{r}}^{PCI} \tag{5}$$

$$\mathbf{R}^{O}_{PCI} = \left[\hat{\mathbf{N}}, \quad \hat{\mathbf{h}} \times \hat{\mathbf{N}}, \quad \hat{\mathbf{h}}\right]^T \tag{6}$$

Since the vectors' magnitude contain no information, the measurements can be described as the angle α between $\mathbf{X}_o$ and the translation vectors $\Delta\hat{\boldsymbol{r}}$. This angle is given as $\alpha = \operatorname{atan2}(\Delta\hat{r}^{O}_{y}, \Delta\hat{r}^{O}_{x})$, where $\Delta\hat{r}^{O}_{x}$ and $\Delta\hat{r}^{O}_{y}$ are the x- and y-coordinates of $\Delta\hat{\boldsymbol{r}}^{O}$, respectively; and atan2 is the arc-tangent function that accounts for the quadrant when calculating the angle.

For further simplification, this frame is rotated around the $\mathbf{Z}_o$-axis by ω, such that the $\mathbf{X}'_o$-axis now points to the orbit's periapsis (see Fig. 1). The direction angle with the $\mathbf{X}'_o$-axis is given as

$$\alpha' = \alpha - \omega \tag{7}$$

Finally, the O''-frame is defined with the origin at the center of the ellipse, instead of the ellipse's focus, and axis parallel to their O'-frame counterpart.

Using the geometric representation shown in Fig. 2, an expression relating the angle α' to the orbit parameters is derived. The orbit is represented as a dashed red line; the blue line is a circle of radius a; and the green circle is a circle of radius equal to the orbit's semi-minor axis b. P is a point in the orbit; A is the point belonging to the blue circle for which the x''_o-coordinate is the same as for P; and B is the point belonging to the green circle for which the y''_o-coordinate is the same as for P. The eccentric anomaly E is the angle between $\mathbf{X}''_o$ and the position vector of point A.

As can be seen from Fig. 2, the position of a point P is given as

$$\mathbf{r}''_P = \begin{bmatrix} a\cos E_P \\ b\sin E_P \\ 0 \end{bmatrix} \tag{8}$$

$$b = a\sqrt{1 - e^2} \tag{9}$$

With this, $\Delta\hat{\boldsymbol{r}}'_{t_i,t_j}$ can be calculated as

$$\Delta\hat{\boldsymbol{r}}'_{t_i,t_j} = \Delta\hat{\boldsymbol{r}}''_{t_i,t_j} = \begin{bmatrix} a\left(\cos E_{t_j} - \cos E_{t_i}\right) \\ b\left(\sin E_{t_j} - \sin E_{t_i}\right) \\ 0 \end{bmatrix} \tag{10}$$

Note that $\Delta\hat{\boldsymbol{r}}'_{t_i,t_j}$ and $\Delta\hat{\boldsymbol{r}}''_{t_i,t_j}$ are the same, since the only transformation between the O'- and the O"-frames is translation.

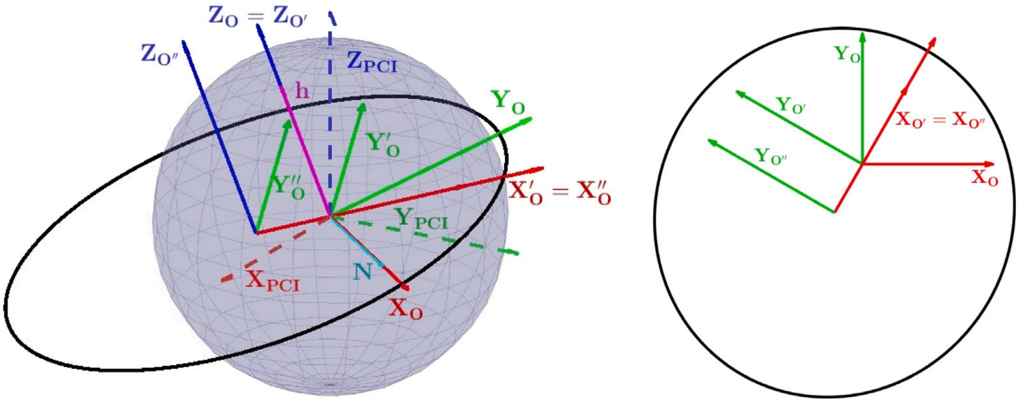
The angle of this direction vector can now be calculated as

$$\alpha' = \operatorname{atan2}\left(\sqrt{1 - e^2}\left(\sin E_{t_2} - \sin E_{t_1}\right), \cos E_{t_2} - \cos E_{t_1}\right) \tag{11}$$

Given its non-linearity, an estimation model based on Eq. (11) requires an initial estimate of the parameters. Obtaining such estimate requires a linear approximation of the expression. First, the translation direction vectors are assumed to be approximately equal to the velocity direction vectors: $\Delta\hat{\boldsymbol{r}}'_{t_i,t_j} \approx \hat{\mathbf{v}}'_{t_{ij}}$, where $\hat{\mathbf{v}}'$ is the normalized velocity vector; and t_{ij} is the time instance between t_i and t_j, given by $t_{ij} = \frac{t_i + t_j}{2}$.

This approximation is valid for small time intervals, since

$$\lim_{t_2 \to t_1} \Delta\hat{\boldsymbol{r}}'_{t_i,t_j} = \hat{\mathbf{v}}'_{t_{ij}} \tag{12}$$



(a) 3D

(b) Orbit plane

Fig. 1. Representation of orbit frames.

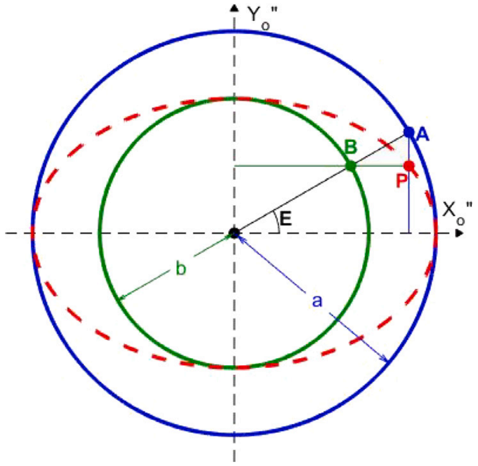


Fig. 2. Diagram of point P as function of orbit shape parameters and E.

and for close to circular orbits, since

$$\hat{\boldsymbol{\Delta r}}'_{t_i,t_j}\Big|_{e=0} = \hat{\mathbf{v}}'_{t_{ij}} \tag{13}$$

The expression for the velocity is obtained by derivating the position vector (Eq. (8)) with respect to time

$$\mathbf{v}'_P = \frac{d\mathbf{r}'_P}{dt} = \begin{bmatrix} -a\sin E\frac{dE}{dt} \\ b\cos E\frac{dE}{dt} \\ 0 \end{bmatrix} \tag{14}$$

which can be used to obtain the angle between the velocity vector and the x'_o-axis with

$$\tilde{\alpha}' = \text{atan2}\left(\sqrt{1-e^2}\cos E, -\sin E\right) \tag{15}$$

where $\tilde{\alpha}'$ is the angle of the velocity vector measured from the $\mathbf{X}'_o$-axis. An approximation of the measured translation direction angle, α, is obtained by combining Eq. (15) with Eq. (7).

$$\alpha \approx \tilde{\alpha} = \tilde{\alpha}' + \omega = \text{atan2}\left(\sqrt{1-e^2}\cos E, -\sin E\right) + \omega \tag{16}$$

The eccentric anomaly E can be calculated through the transcendental equation, $M = E - e\sin E$, with an iterative method. Finally, M is given as a function of time as $M = M_0 + nt$, where n is the mean angular motion.

Since Eq. (16) is still non-linear, it is simplified further in two steps. First, E is approximated as $E \approx M + e\sin M$. Substituting this approximation into Eq. (16) yields

$$\tilde{\alpha} \approx \tilde{\alpha}_{E1} = \text{atan2}\left(\sqrt{1-e^2}\cos(M + e\sin M), -\sin(M + e\sin M)\right) + \omega \tag{17}$$

Then the first order Taylor approximation of Eq. (16) around $e = 0$ is calculated

$$\tilde{\alpha} \approx \tilde{\alpha}|_{E_1,e=0} + \frac{d\tilde{\alpha}_{E_1}}{de}\bigg|_{e=0} e \tag{18}$$

The approximated velocity angle $\tilde{\alpha}_{E1}$ evaluated at $e = 0$ is given by Eq. (19). This expression is obtained by substituting $e = 0$ in Eq. (17) and simplifying it through trignometric relationships.

$$\tilde{\alpha}_{E1}|_{e=0} = M + \omega + \frac{\pi}{2} \tag{19}$$

The first derivative of $\tilde{\alpha}_{E1}$ with respect to e is given by Eq. (20).

$$\frac{d\tilde{\alpha}_{E1}}{de} = \frac{e\cos(E_1)\sin(E_1) + (1-e^2)\sin(M)}{\sqrt{1-e^2}\left(1 - e^2\cos^2(E_1)\right)} \tag{20}$$

When evaluated at $e = 0$, this derivative is equal to

$$\frac{d\tilde{\alpha}_{E1}}{de}\bigg|_{e=0} = \sin(M) \tag{21}$$

Substituting Eqs. (19) and (21) into Eq. (18) yields the final approximation for $\tilde{\alpha}$

$$\tilde{\alpha} \approx M + \omega + \frac{\pi}{2} + e\sin(M) \tag{22}$$

$$\approx nt + M_0 + \omega + \frac{\pi}{2} + e\sin(nt + M_0) \tag{23}$$

$$\approx nt + \underbrace{M_0 + \omega + \frac{\pi}{2}}_{c_{line}} + e\cos M_0\sin(nt) + e\sin M_0\cos(nt) \tag{24}$$

Thus, $\tilde{\alpha}$, and by extension α, can be approximated as a straight line of slope equal to n, added to a sinusoidal function with the same value for its frequency and an amplitude equal to e.

Although this approximation is still non-linear with respect to n, it is possible to obtain an appropriate initial estimate with linear LSQ via iteration, as is shown in Section 3.2.

3. Method

In this section, estimation models are derived from the theoretical framework described in Section 2.

In Section 3.1, a linear LSQ model is derived to estimate the inclination, i, and the longitude of the ascending node, Ω, from the measured translation direction vectors, $\boldsymbol{\Delta}\hat{\mathbf{r}}$, according to the relation shown in Section 2.1.

The estimated orbit plane orientation parameters are used to transform the measurements from the PCI-frame into the O-frame, which is defined such that the XY_O-plane coincides with the estimated orbit plane. By ignoring the z_O-coordinate of the measured translation direction vectors, the problem of determining the ellipse shape parameters is simplified into a 2D problem. The measurements are then described as an angle α over time.

Finally in Section 3.2, the ellipse shape parameters are estimated according to the relations derived in Section 2.2. This begins with an iterative linear LSQ model derived from approximations of the α function. The resulting estimate is used to initialize a non-linear LSQ model derived from a more accurate expression of α.

3.1. Estimation of orbit plane orientation parameters

To account for the measurement noise and potential outliers, a linear LSQ estimation within a RANSAC scheme [36] is used to estimate the orbit's inclination, i, and longitude of ascending node, Ω. The estimation is based on Eq. (1). First, multiple pairs of translation direction vectors are used to calculate the associated normal vector. These normal vectors are then used as measurements for the LSQ model described in Eqs. (25) through (27), where f is the model function, r_i are the residuals, and y_i are the calculated normal vectors.

$$f(\mathbf{x}, \hat{\mathbf{h}}) = \hat{\mathbf{h}} \tag{25}$$

$$\mathbf{r}_i = \mathbf{y}_i - \hat{\mathbf{h}} \tag{26}$$

$$\mathbf{y}_i = \hat{\boldsymbol{\Delta r}}_j \times \hat{\boldsymbol{\Delta r}}_k \tag{27}$$

The LSQ solution is given by minimizing the sum of the squared residuals $S = \sum_i \mathbf{r}_i^2$, which is done by finding the root of the partial derivatives $\frac{\partial S}{\partial \hat{h}_j}$, where $j = 1, 2, 3$.

$$\frac{\partial S}{\partial \hat{h}_j} = 2\sum_i^N \left(\hat{\mathbf{h}} - \mathbf{y}_i\right) \frac{\partial \hat{\mathbf{h}}}{\partial \hat{h}_j} \tag{28}$$

$$\frac{\partial \hat{\mathbf{h}}}{\partial \hat{h}_j} = \begin{cases} 1, \text{if } i = j \\ 0, \text{if } i \neq j \end{cases} \tag{29}$$

Eq. (33) shows the solution for $j = 1$. Similar results are obtained for $j = 2$ and $j = 3$.

$$\frac{\partial S}{\partial \hat{h}_1} = 2\sum_i^N \left(\hat{\mathbf{h}} - \mathbf{y}_i\right) \frac{\partial \hat{\mathbf{h}}}{\partial \hat{h}_1} = 0 \tag{30}$$

$$\sum_i^N \left(\hat{\mathbf{h}} - \mathbf{y}_i\right) [1 \quad 0 \quad 0]^T = 0 \tag{31}$$

$$\sum_i^N \left(\hat{h}_1 - y_{i,1}\right) = N\hat{h}_1 - \sum_i^N y_{i,1} = 0 \tag{32}$$

$$\hat{h}_1 = \frac{\sum_i^N y_{i,1}}{N} \tag{33}$$

Substituting Eq. (27) into Eq. (33) and the equivalent solutions for the remaining coordinates, the estimate of $\hat{\mathbf{h}}$ is given by

$$\tilde{\mathbf{h}}_e = \sum_{j,k=j+1} \frac{\hat{\boldsymbol{\Delta r}}_j \times \hat{\boldsymbol{\Delta r}}_k}{\left\| \hat{\boldsymbol{\Delta r}}_j \times \hat{\boldsymbol{\Delta r}}_k \right\|} \tag{34}$$

$$\hat{\mathbf{h}}_e = \frac{\tilde{\mathbf{h}}_e}{\|\tilde{\mathbf{h}}_e\|} \tag{35}$$

The pairs of translation direction vectors are selected such that $k > j$; the angle between the vectors is higher than 10° and lower than 170°; and $\hat{\boldsymbol{\Delta r}}_k$ is obtained within one orbit period from $\hat{\boldsymbol{\Delta r}}_j$. The minimum angle between vectors is meant to avoid crossing vectors that are close to parallel. The remaining conditions are meant to ensure the correct order of operation and, as a result, the correct direction of the resulting normal vectors.

Outlier rejection is done via RANSAC. From the available measurements, 100 different pairs of vectors are randomly selected. Each pair is used to calculate the corresponding normal vector. For each of these models, measured vectors that are within 2° of perpendicular with the calculated normal vector are considered inliers. In each case, the inliers are used to obtain a new estimate of $\hat{\mathbf{h}}$. This process is repeated until convergence. When all 100 estimations have converged, the one which results in the most inliers is chosen.

Once the estimated normal vector $\hat{\mathbf{h}}_e$ is calculated, the estimated inclination i_e and the longitude of ascending node Ω_e are determined according to Eqs. (2) and (3), respectively.

3.2. Estimation of ellipse shape parameters

As was shown in Section 2.2, the expression that relates the angle of the translation direction, α, to the ellipse shape parameters is non-linear. As a consequence, the non-linear LSQ estimation requires a good initial estimate. Approximations were made to this expression to allow for an initial estimate via linear LSQ, resulting in Eq. (24).

Note that this function is continuous, unlike the measured α which ranges from 0° to 360°. As such, the estimation process begins by increasing or decreasing the measured α by 360° until continuity is obtained.

The approximation, given by Eq. (24) is still non-linear with respect to the mean motion n. To account for this, n is estimated as the slope of the straight line that best fits the measured α over time function. This fit is less accurate for higher eccentricity values, and depending on which portion of the orbit is in shadow. Thus, a second initial estimate of n is obtained through an orbit period estimate, consisting in the average time difference between α_i, measured during the first orbit, and α_j, the angle measured during the second orbit which has the closest value to α_i. The remaining estimation steps are performed for each of these models, after which the model resulting in the lowest residuals is selected.

A series of linear LSQ estimations iterating over the estimated values of n is performed. For each iteration, the measurements are fitted to the curve given by

$$\alpha \approx \beta_{1,k} \sin(n_{k-1}t) + \beta_{2,k} \cos(n_{k-1}t) + n_k t + \beta_{4,k} \tag{36}$$

$$\beta = \left[e\cos M_0, \quad e\sin M_0, \quad n, \quad M_0 + \omega + \frac{\pi}{2}\right]^T \tag{37}$$

where $\beta_{X,k}$ is the Xth element of the estimated state β for iteration k.

As is represented in Eq. (36), n is kept as a constant in the equation components that interact with other estimation parameters, thus making it possible to linearly estimate the remaining parameters and obtain a new estimate of n. The iterations are repeated until the difference between the last two estimations of n are within a tolerance of 1×10^{-10}. The LSQ model used in these iterations is given by Eqs. (38) and (39).

$$\mathbf{X}_k = \begin{bmatrix} \sin(n_{k-1}t_1) & \cos(n_{k-1}t_1) & t_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \sin(n_{k-1}t_N) & \cos(n_{k-1}t_N) & t_N & 1 \end{bmatrix} \tag{38}$$

$$\mathbf{y} = \left[\alpha_1 \dots \alpha_N\right]^T \tag{39}$$

Once the iteration converges, the orbit parameters are calculated from the estimation parameters as follows

$$\hat{a} = \sqrt[3]{\frac{\mu}{\hat{\beta}_3^2}} \tag{40}$$

$$\hat{e} = \sqrt{\hat{\beta}_1^2 + \hat{\beta}_2^2} \tag{41}$$

$$\hat{M}_0 = \operatorname{atan2}\left(\hat{\beta}_2, \hat{\beta}_1\right) \tag{42}$$

$$\hat{\omega} = \hat{\beta}_4 - \hat{M}_0 - \frac{\pi}{2} \tag{43}$$

For cases where $\hat{e}$ is below a tolerance of 0.001, a circular orbit is assumed and the following parameters are used instead

$$\hat{a} = \sqrt[3]{\frac{\mu}{\hat{n}_0^2}} \tag{44}$$

$$\hat{e} = 0 \tag{45}$$

$$\hat{M}_0 = c_{line} - \frac{\pi}{2} \tag{46}$$

$$\hat{\omega} = 0 \tag{47}$$

This is done since, for circular orbits, a straight line is a better model of the measurements than Eq. (24). Doing so also addresses the redundancy of the ω parameter for circular orbits.

With this initial estimate, the OD can now be refined using non-linear LSQ for Eq. (16). Here, the approximation $\alpha \approx \tilde{\alpha}$ is used, since the true equation for α (Eq. (11)) is not consistently initialized properly with the initial estimates obtained from Eq. (38).

Eq. (48) through (57) show the model used for this non-linear LSQ

$$\beta = \begin{bmatrix} a & e & \omega & M_0 \end{bmatrix}^T \tag{48}$$

$$\hat{\alpha}_k(t_i, \hat{\beta}_{k-1}) = \text{atan2}\left(\sqrt{1-\hat{e}_{k-1}^2}\cos(E(t_i, \hat{\beta}_{k-1})), -\sin(E(t_i, \hat{\beta}_{k-1}))\right) + \hat{\omega}_{k-1} \tag{49}$$

$$\delta\mathbf{Y}_k = \begin{bmatrix} \text{angleError}\left(\alpha_{meas,1} - \hat{\alpha}_k(t_1, \hat{\beta})\right) \\ \vdots \\ \text{angleError}\left(\alpha_{meas,N} - \hat{\alpha}_k(t_N, \hat{\beta})\right) \end{bmatrix} \tag{50}$$

$$\frac{d\alpha}{da} = -\frac{3}{1+(1-e^2)\cot^2(E)}\frac{\sqrt{1-e^2}}{2\sin^2(E)(1-e\cos(E))}\sqrt{\frac{\mu}{a^5}}t \tag{51}$$

$$\frac{d\alpha}{de} = \frac{1}{1+(1-e^2)\cot^2(E)}\left(\frac{e\cot(E)}{\sqrt{1-e^2}} + \frac{\sqrt{1-e^2}}{\sin(E)(1-e\cos(E))}\right) \tag{52}$$

$$\frac{d\alpha}{d\omega} = 1 \tag{53}$$

$$\frac{d\alpha}{dM_0} = \frac{1}{1+(1-e^2)\cot^2(E)}\frac{\sqrt{1-e^2}}{\sin^2(E)(1-e\cos(E))} \tag{54}$$

$$\mathbf{J}_k = \begin{bmatrix} \left.\frac{d\alpha}{da}\right|_{t_1,\beta_{k-1}} & \left.\frac{d\alpha}{de}\right|_{t_1,\beta_{k-1}} & \left.\frac{d\alpha}{d\omega}\right|_{t_1,\beta_{k-1}} & \left.\frac{d\alpha}{dM_0}\right|_{t_1,\beta_{k-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \left.\frac{d\alpha}{da}\right|_{t_N,\beta_{k-1}} & \left.\frac{d\alpha}{de}\right|_{t_N,\beta_{k-1}} & \left.\frac{d\alpha}{d\omega}\right|_{t_N,\beta_{k-1}} & \left.\frac{d\alpha}{dM_0}\right|_{t_N,\beta_{k-1}} \end{bmatrix} \tag{55}$$

$$\delta\beta_k = (\mathbf{J}_k^T\mathbf{J}_k)^{-1}\mathbf{J}_k^T\delta\mathbf{Y}_k \tag{56}$$

$$\beta_k = \beta_{k-1} + \delta\beta_k \tag{57}$$

where $angleError(X, Y)$ is the minimum difference between two angles, *e.g.*, $angleError(10°, 350°)$ is 20° and not 340°; and the eccentric anomaly E is calculated using the Newton–Raphson method given by

$$E_0 = M \tag{58}$$

$$E_k = E_{k-1} - \frac{M + e\sin E_{k-1} - E_{k-1}}{e\cos E_{k-1} - 1} \tag{59}$$

It was observed that performing a new non-linear LSQ estimation using the true expression of α, without the approximation $\Delta\hat{\mathbf{r}}_{t_i,t_j} \approx \hat{\mathbf{v}}_{t_{ij}}$, does not consistently improve the estimation results. Therefore, the orbit parameters estimated via Eq. (48) through (57) are used as the final estimate.

4. Results

To test the estimation method, 5000 Moon orbiting trajectories were simulated, using the spherical harmonics GRAIL model [37] with degree and order 660. Three different camera elevation angles were considered, 0° (nadir-pointing), 30° and 60°. The camera azimuth angle was uniformly distributed between 0° and 360°. Each trajectory was tested with each of the camera elevation angles. For each of these 15000 cases, three different target baselines, 50 px, 250 px and 450 px, were used for each combination of trajectory and camera orientation, for a total of 45000 estimations. The target baseline is the desired average magnitude of the OF vectors in a given image, and is used to adjust the time interval between the images used. Varying this interval along the orbit accounts for the change in velocity and altitude observed in non-circular orbits. By adjusting the time interval between images based on the OF baselines, the method can be applied to a wider range of orbit parameters, in comparison to previous methods.

OF was simulated based on the camera's state. For each image pair, random points were selected within the first image frame. The 3D coordinates of the surface points associated with these image features were calculated and reprojected onto the second image plane using the pinhole camera model and assuming a spherical lunar surface. Errors and outliers were added using a performance model of AKAZE features (see Appendix A). Simulated orientation measurements were used to remove the rotation component of the OF. The resulting purely translational OF was used to calculate the epipole by finding the intersection point of the OF vectors via linear LSQ and RANSAC. Finally, the estimated coordinates of the epipole were used to determine the translation direction vectors used to estimate the orbit parameters. Appendix B provides further details on the estimation of the translation direction vectors. Obtaining the translation direction vectors from the epipole removes the requirement of near nadir-pointing cameras present in other methods.

The estimation method was further tested on rendered images. From the simulated trajectories, 43 were selected to be rendered. The selection includes only trajectories for which latitudes above 60° or below −60° cannot be seen from any of the cameras' positions. This is due to the limitations of the 3D model used (see Section 4.1). Each trajectory was rendered for the 3 different camera orientations. An additional 9 trajectories were generated, spanning the higher value range of the semi-major axis, which was lacking in the 43 trajectories originally selected. These new trajectories were circular, equatorial, with an azimuth angle of 0°, to account for the limits of the 3D rendering model.

The workflow when using rendered images was the same as for simulated measurements, except the OF was obtained from rendered images using AKAZE features. Features are detected in two different images and matched. These features are used to define the corresponding OF vectors: origin at the feature found in the first image, and end at the feature found in the second image. The collection of these OF vectors is the OF map used to estimate the heading direction between the associated images.

For both the simulated OF maps and the rendered images, the camera used had a field of view of 60° and an image resolution of 1024 px. Orientation measurements were simulated using 30 arcsec error in the boresight direction and 10 arcsec error in the lateral directions.

Section 4.2 introduces the metrics used to quantify the performance of the estimation algorithm. In Section 4.3, the results obtained for simulated measurements are shown. Section 4.4 provides the results obtained using rendered images for the estimation. Section 4.5 provides the results of the CNav initialization tests performed.

4.1. Image rendering – sensorDTM

sensorDTM is a version of the more general rendering software *SENSOR++*, which has been optimized for the lunar environment. This software was designed to translate a planet's Digital Elevation Map (DEM), Sun position, camera states and camera parameters into rendered images. Apart from the accurate depiction of the shadow patterns observed in lunar surface images, it also adds acquisition noise to the rendered image [38]. Further justifying its selection, there is precedent for the usage of this renderer in space exploration research, namely during the testing phase of the ATON (Autonomous Terrain-based Optical Navigation) project [39].

Despite its relevancy, *sensorDTM* has a few limitations. The software currently supports DEM data provided in simple cylindric projection. The DEM maps generated from the Japanese Selene (Kaguya) mission were selected for its near global coverage at sufficient resolution and for the use of the simple cylindric projection [40]. These maps are restricted to the latitude range of −60° to 60°, which accounts for about 87% of the lunar surface. DEMs of the remaining surface area are only available in polar stereographic projections. Due to the complications associated with transforming these DEMs into the necessary simple cylindric projection and verifying the results, and since the available DEM accounts for the vast majority of the surface, it was decided to restrict the study to the latitude range of −60° to 60°.

Second, the model is currently restricted to a constant albedo value which is set at the beginning of the rendering process. In other words, albedo textures cannot be used to better reflect the lunar surface's material. Finally, it does not simulate stars in the image. It is expected that, due to the much higher brightness of the lunar surface, stars will not be visible in properly exposed images. If this is not the case, stars could lead to additional outliers in the OF maps. However, if this is later found to be an issue, the visible surface can be isolated via a region of interest mask, defined through limb detection.

The images were rendered using a Field-of-View (FOV) of 60° and a square resolution of 1024 px.

4.2. Performance metrics

The goal of the estimation method is to provide an initial estimate for a TAN optical navigation algorithm.

The TAN selected for the tests is CNav, which works by matching craters detected in the captured images to craters available in a catalog. The initial state estimate is used to project the catalog craters onto the image. If the projected craters are sufficiently close to detected craters of similar size, CNav operates in tracking mode. In tracking mode, the craters are matched within the image space with a success rate of 95%. When this type of matching is not possible, CNav operates in acquisition mode. In acquisition mode, the relative depth of the detected craters must be estimated to enable matching them with the catalog craters in 3D space [41].

The success rate in acquisition mode has yet to be quantified. A conservative estimate of 10% success rate in acquisition mode will be used in this paper [42]. However, many factors can impact the success rate of CNav in acquisition mode, including the crater density of the observed terrain, the illumination angle and the coverage of the crater catalog used. Whereas crater density and illumination conditions are inherent to the trajectory, the catalog coverage is dependent on the quality of the initial estimate provided by the method presented in this paper.

The ideal catalog should include all craters visible in a given image, and no additional craters. Not only do large crater catalogs increase the computation time, but they also reduce the rate of true solutions, due to an increase of false crater matches. It is hypothesized that the catalog coverage can be related to the likelihood of a successful CNav acquisition solution at each time instance. The more craters found in both the image and corresponding catalog, and the less craters found in only one of these sets, the higher the probability of obtaining an accurate solution. This hypothesis has yet to be tested.

Once an estimated camera state is obtained with the proposed OD algorithm, it is used to generate the corresponding crater catalog, by selecting the craters that would be visible from the estimated camera state. Therefore, the performance of the OD algorithm is measured according to the catalog coverage, which can be quantified as the percentage of common area between the true state image and the corresponding estimated state virtual image. Considering the goal of initializing CNav, this metric is preferable to others, such as position error, since it is more directly related to how CNav functions. For example, assuming a 60° FOV camera, a 10 km position error at 10 km altitude will result in close to 0% of common area, while at 20 km altitude the same position error will result in about 50% of common area. In other words, metrics based on absolute position error do not correlate with the likelihood of CNav initialization.

The percentage of common area is given by

$$p_{common} = 100\frac{A^r_{common}}{A^r_{total}} \tag{60}$$

$$A^r_{total} = A^r_{true} + A^r_{est} - A^r_{common} \tag{61}$$

where p_{common} is the percentage of common area; A^r_{common} is the common area, in square pixels, between the true state and estimated state images as seen in the true image frame; A^r_{total} is the combined area of the true state and estimated state images as seen in the true image frame; A^r_{true} is the image area of the true state image; and A^r_{est} is the image area of the estimated state virtual image as seen in the true state image. These areas only refer to pixels representing the surface of the planet. In other words, they do not include points above the horizon. Fig. 3 depicts the workflow used to obtain the value of these areas.

The border points are the vertices of a convex polygon defining the intersection between the image frame and the horizon line projected onto the image. The horizon line is defined by determining the intersection line between the spherical surface of the Moon and the tangent cone centered on the camera, and projecting it onto the image. Fig. 4 shows the geometric relationship used to define the intersection line.

The border points of the estimated state virtual image are projected onto the real image, through the pinhole camera model. The intersection between these two polygons defines the common area. The areas of the three polygons are calculated according to Eq. (62), where the vertices are ordered clockwise; N is the number of vertices in the polygon; u_i and v_i are the image coordinates of vertex i; and vertex 0 is equal to vertex N.

$$A_{polygon} = \frac{1}{2}\sum_{i=0}^{N}\left(u_i v_{i+1} - u_{i+1} v_i\right) \tag{62}$$

Once again, it is expected that the percentage of common area has an impact on CNav's acquisition success rate. Considering the extremes, a percentage of common area close to 0% will likely result in a very low success rate; and a percentage of common area close to 100% will likely result in a higher success rate. Since this effect has yet to be quantified, a flat success rate of 10% is assumed for catalogs above a given coverage threshold. Eq. (67) shows the probability of obtaining one true solution with CNav's acquisition mode, within a given number of images, N_{images}.

$$p_{acquisition,N_{Images}} = 1 - p_{no\ acquisition,N_{Images}} \tag{63}$$

$$= 1 - \prod_{i}^{N_{images}} p_{no\ acquisition/image} \tag{64}$$

$$= 1 - \prod_{i}^{N_{images}} \left(1 - p_{acquisition/image}\right) \tag{65}$$

$$= 1 - \prod_{i}^{N_{images}} (1 - 0.1) \tag{66}$$

$$p_{acquisition,N_{Images}} = 1 - (0.9)^{N}_{images} \tag{67}$$

Solving for N_{images} with $p_{acquisition,N_{Images}} = 0.99$, to reach 99% probability of at least one successful CNav acquisition solution, at least 44 images with sufficient catalog coverage must be run in this mode. Assuming a conservative processing time of 10 s per image, this translates to 440 s. After one successful acquisition solution, CNav runs in tracking mode, for which the estimated success rate is 95%. Therefore, within 440 s of sufficient catalog coverage, there is a 99% chance of successful CNav initialization, through the acquisition mode, and a 95% rate of accurate state estimation on subsequent images, through the tracking mode. In other words, an OD estimation is considered successful if it results in sufficient catalog coverage for at least 440 s of the estimated trajectory.

Catalog coverage is considered "sufficient" if it results in at least 10% CNav acquisition success rate. Since the minimum percentage of common area for which this occurs is currently unknown, the OD algorithm's performance was evaluated for different thresholds of minimum coverage.

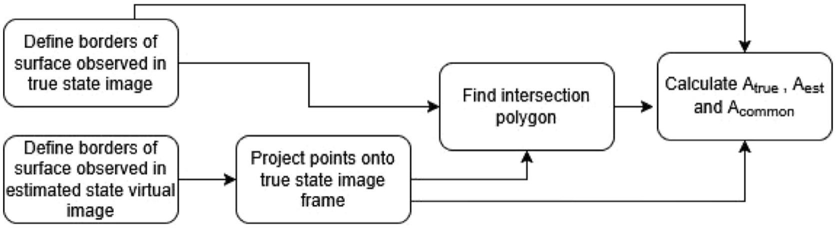


Fig. 3. Workflow used to calculate percentage of common area.

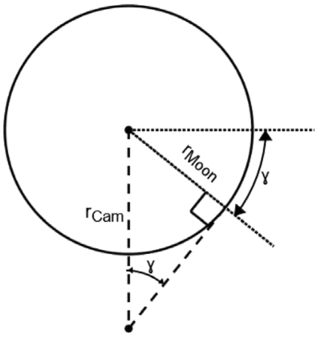


Fig. 4. Intersection between spherical surface and tangent cone.

Table 1
Estimation success rate for simulated feature measurements as a function of the camera elevation.

Threshold	Camera elevation		
common area [%]	0°	30°	60°
90%	99.4% (14908)	99.6% (14943)	56.7% (8511)
50%	99.8% (14972)	99.9% (14980)	64.0% (9593)
10%	99.8% (14972)	99.9% (14983)	64.6% (9691)

Table 2
Estimation success rate for simulated measurements as a function of the target baseline.

Threshold	Target baseline		
common area [%]	50 px	250 px	450 px
90%	83.0% (12445)	86.6% (12986)	86.2% (12931)
50%	88.2% (13232)	88.2% (13226)	87.2% (13087)
10%	88.7% (13311)	88.3% (13247)	87.3% (13088)

Table 3
Estimation success rate for rendered images as a function of the camera elevation.

Threshold	Camera elevation		
common area [%]	0°	30°	60°
90%	77.5% (100)	80.6% (104)	17.1% (22)
50%	90.7% (117)	88.4% (114)	61.2% (79)
10%	91.5% (118)	89.1% (115)	62.8% (81)

Table 4
Estimation success rate for rendered images as a function of the target baseline.

Threshold	Target baseline		
common area [%]	50 px	250 px	450 px
90%	54.3% (70)	62.0% (80)	58.9% (76)
50%	86.8% (112)	82.2% (106)	71.3% (92)
10%	88.4% (114)	83.7% (108)	71.3% (92)

4.3. *Results using simulated measurements*

Tables 1 and 2 shows the success rate for different common area thresholds, as a function of the camera elevation and target baseline, respectively, when using simulated measurements.

The target baseline has no significant impact on the estimation performance. On the other hand, there is a clear drop in the success rate for camera elevations of 60°. Unlike for lower camera elevations, when a 60° FOV camera is oriented 60° off-nadir, the horizon will be visible regardless of the camera's altitude. This is the case since the cone of vision includes a ray perpendicular to the nadir direction (60° from the camera elevation plus 30° from half the FOV), which has no intersection with the lunar surface.

In other words, for 60° of camera elevation, all images have a reduced area in which features can be detected, representing the lunar surface. This area is smaller, the higher the camera's altitude. The smaller the image area with visible surface, the worse the geometric dilution of precision becomes, when estimating the translation direction from OF measurements. This, in turn, leads to worse OD results.

Fig. 5 shows the reduction of the success rate with the increase of the radius of periapsis for a camera elevation of 60°. Fig. 5(a) shows the number of success cases. In Fig. 5(b), the same data is shown as a percentage of the cases within a given range of periapsis values. As can be seen, the altitude has no effect on the success rate for lower camera elevations.

4.4. *Results using rendered images*

Tables 3 and 4 shows the success rate for different common area thresholds, as a function of the camera elevation and target baseline, respectively, when using rendered images.

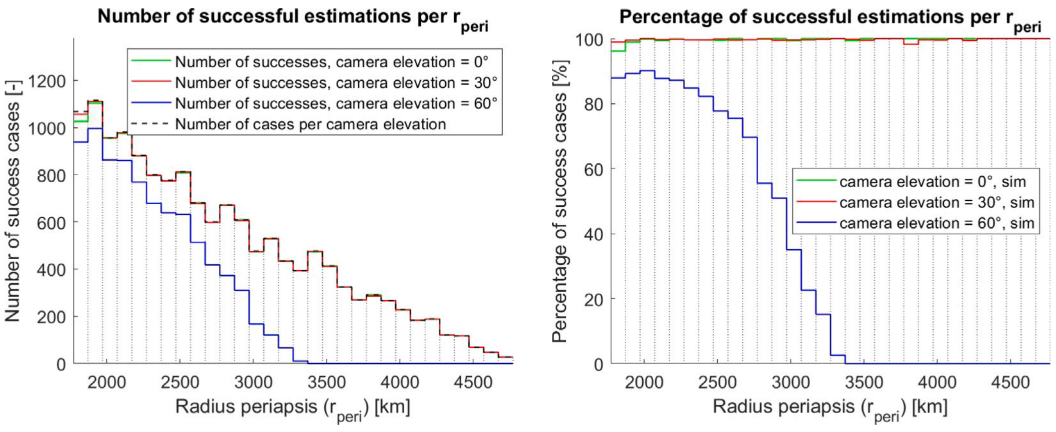
Comparing Tables 3 and 4 with Tables 1 and 2, the reduced success rate for a camera elevation of 60° is still the main effect when using rendered images. As with simulated measurements, the target baseline does not have a significant effect on the success rate.

Fig. 6 shows the effect of the radius of periapsis on the number of success cases for rendered images. Fig. 7 shows the effect of the radius of periapsis on the success rate for rendered images in comparison with the results using simulated measurements.

The small sample size available for the estimation results using rendered images does not allow for statistical significance to be achieved. The results obtained are, therefore, meant to identify possible discrepancies with the results from simulated measurements.

As expected, the general performance when using rendered images is lower than when using simulated measurements. The success rate is lower for camera elevations of 60°, and it drops to 0% for radius of periapsis of about 3500 km and higher. This effect is explained by the smaller visible surface area available in the images used to measure OF, as mentioned in Section 4.3. Fig. 8(b) shows the OD success rate as a function of the average percentage of visible surface area of the images used to measure OF. The decreased success rate for lower percentages of average visible area supports the hypothesis mentioned above.

Unlike for the simulated measurements, results for nadir-pointing cameras, 0° camera elevation, for the higher altitude circular orbits



(a) Number of successes (b) Success rate

Fig. 5. Success as a function of radius of periapsis for simulated measurements.

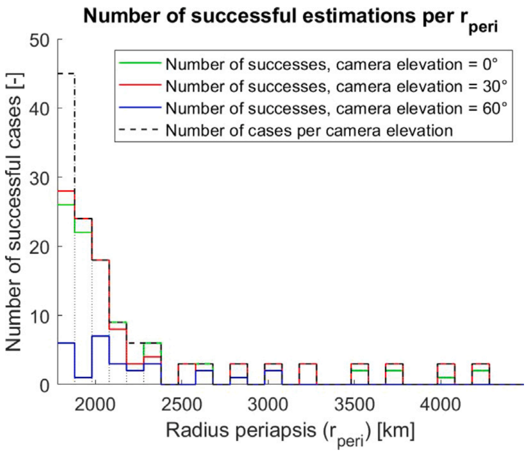


Fig. 6. Number of successes as a function of radius of periapsis for rendered images.

show a decrease in success rate, which is not observed for 30° camera elevation. This is likely due to the method used to estimate the translation direction from the OF maps. After removing the rotation component from the measured OF, the vectors of the resulting translation OF are used to find the epipole. This is done by finding the most likely intersection point of the vectors, via LSQ and RANSAC. Since for nadir-pointing cameras in a circular orbit, the noiseless translation OF is a map of parallel vectors, this method is sub-optimal for this combination of parameters. Although an intersection point can be obtained, due to measurement noise, the estimation error of each translation direction vector will be higher than for other parameter configurations. This in turn reduces the accuracy of OD. As such, for circular orbits, it would be advisable to angle the camera to point off-nadir.

4.5. CNav initialization tests

The TAN method used for these tests was CNav [23], a crater-based method. Image specific catalogs are created with the craters that would be visible from the estimated camera state at that time instance. Additionally, an aiding catalog combines all the craters included in at least one image specific catalog. More accurate estimations increase the number of catalog craters that are seen in the real image and reduce the number of catalog craters that are outside of image bounds, thus increasing the chance of true matches between observed and catalog craters, and reducing the chance of false matches. It is thus expected that the rate of CNav true solutions will increase for higher percentage of common area (see Section 4.2) associated with the OD.

As such, the selection of test cases was made based on the percentage of common area obtained from the OD. Of the 387 estimation results, 5 were selected. These were the cases for which the maximum percentage of common area threshold resulting in a successful estimation was 50%. In other words, for these cases there are least 44 images at 0.1 Hz with at least 50% common area between estimated and real images; but less than 44 images with at least 60% common area.

This choice was motivated by the expectation that, if initialization is possible for a 50% common area threshold, it will be more likely for higher thresholds. The vast majority of estimations were successful for a 50% common area threshold, about 90% of cases with camera elevations of 0° and 30°, and about 60% of cases with camera elevations of 60° (see Table 3). Therefore, if initialization is successful for the lower performance cases in this subset, it is likely to be achievable for the remaining cases. The initialization tests were done in open loop, meaning that the CNav results were not used to improve the state estimation.

Table 5 shows the number of true solutions obtained for each of the initialization tests. A CNav solution is considered a true solution if all crater matches used to estimate the camera pose are true matches. The crater matches are tested by projecting the matched catalog craters onto the image, using the true camera pose, and comparing the resulting image coordinates with the matched detected craters. These results can also be verified by images such as Fig. 9, where detected craters are marked in red, projected catalog craters are in blue, and matched craters are in cyan.

As can be seen, the minimum number of true solutions obtained was 29. Since after just one true solution is obtained by CNav the rate of true solutions in tracking mode is about 95%, these results satisfy the CNav initialization requirement.

5. Conclusion

In this paper, an algorithm was introduced to estimate Kepler orbit parameters from translation direction vectors, obtained via Optic Flow (OF). It was developed to provide an autonomous initial estimate for pre-existing Terrain-relative Absolute Navigation (TAN) methods.

The inclination and longitude of the ascending node are estimated via the normal vector between pairs of translation direction vectors. The measured translation direction vectors are projected onto the

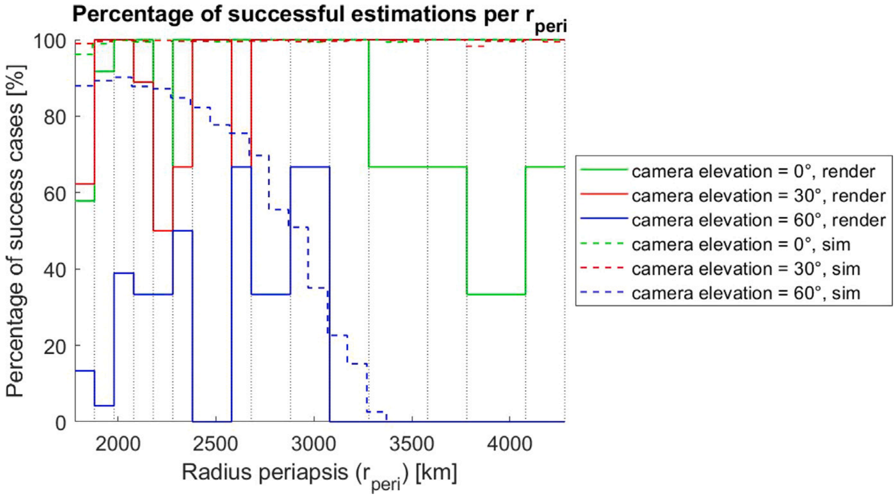
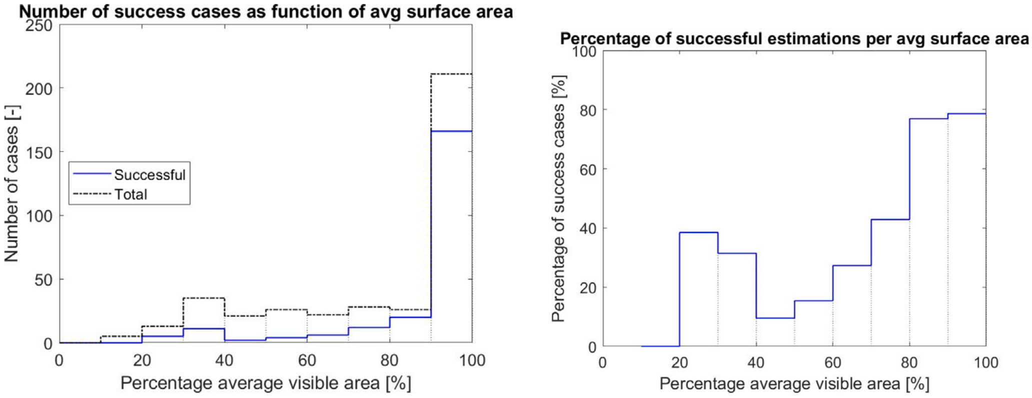


Fig. 7. Success rate as a function of radius of periapsis.



(a) Number of successes (b) Success rate

Fig. 8. Success as a function of average visible surface area.

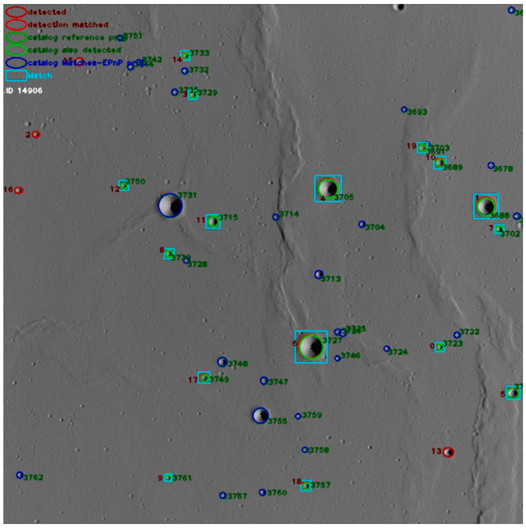


Fig. 9. Example of crater catalog projection with CNav solution.

Table 5
CNav initialization results.

Estimation case			Number of true solutions
Trajectory id	Camera elevation	Target baseline	
690	60°	50 px	58
1013	60°	50 px	92
1352	60°	50 px	29
3296	60°	250 px	47
3380	60°	450 px	91

estimated orbit plane; and the direction is described as the angle of the projected vector with respect to the ascending node.

The remaining 4 orbit parameters can then be estimated by LSQ estimation. Since the direction angle over time is highly non-linear, an initial estimate is obtained through iterative linear LSQ using an approximation around eccentricity equal to 0. This estimate is then used to initialize the final non-linear LSQ estimation. The estimated orbit parameters are then used to predict the camera's state, which can be used to generate the crater catalogs necessary to operate Crater Navigation (CNav).

The algorithm was first tested with simulated OF measurements on 5000 simulated trajectories, each with 3 different camera orientations and 3 different target baselines, for a total of 45000 test cases. From these, 43 trajectories were selected to be rendered. An additional 9 circular and equatorial trajectories were simulated at high altitudes. Each trajectory was rendered for 3 different camera orientations. The algorithm was then tested with these rendered images, where each image set was tested for 3 different target baselines, for a total of 468 test cases.

The performance was quantified via the percentage of common visible surface area between the real image and the image seen from the estimated camera state. For a given percentage threshold, an estimation is considered successful if at least 44 estimated camera states at 0.1 Hz are above said threshold.

For simulated measurements, at a threshold of 90% of common area, about 99% of cases with 0° and 30° of camera elevation were successful; and about 57% of cases with 60° of camera elevation were successful. For rendered images, at a threshold of 90% of common area, the success rate was about 77%, 81% and 17% for 0°, 30° and 60°, respectively. For a 50% threshold, the success rate increases to 91%, 88% and 61%, respectively.

For both simulated measurements and rendered images, a camera elevation of 60° results in a significantly lower success rate than 0° and 30°. This success rate decreases with increased radius of periapsis of the orbit. This effect is caused by the decrease in visible surface area available in the images used to determine OF.

Of the estimation results obtained with rendered images, 5 cases were chosen for CNav initialization tests. The maximum percentage of common area threshold resulting in a successful OD for these cases was 50%, making them some of the lowest performing testcases. The number of true solutions obtained with CNav ranged from 29 to 92. Given that CNav's rate of true solutions in tracking mode, after a true solution is obtained, is 95%, initialization is achieved. Since it is expected that for higher percentage of common area, the rate of CNav true solutions will increase, these tests indicate that the algorithm is sufficiently accurate to initialize CNav for the majority of orbits (about 90% of the test cases with camera elevation of 0° and 30°; and about 61% with camera elevation of 60°).

In conclusion, the method described was shown to allow for fully autonomous OD with sufficient precision to initialize existing TAN methods.

List of symbols

a	Semi-major axis
α	Angle between $\mathbf{X}_O$ and $\mathbf{\Delta r}$
A^r_{name}	Image surface area of “name” as seen from real camera state
b	Semi-minor axis
c_{line}	Y-intersect of $\alpha(t)$ for circular orbits
$\mathbf{\Delta r}_{t_i,t_j}$	Translation direction from position at t_i to position at t_j
E	Eccentric anomaly
e	Orbit eccentricity
$\mathbf{h}$	Specific relative angular momentum vector
i	Orbit inclination
M	Mean anomaly
μ	Gravitational parameter
$\mathbf{N}$	Vector in orbit plane from planet center to ascending node
n	Mean motion
ν	True anomaly
O	Orbit frame, $\hat{\mathbf{X}}_O = \hat{\mathbf{N}}$, $\hat{\mathbf{Z}}_O = \hat{\mathbf{h}}$
Ω	Longitude of ascending node
ω	Argument of periapsis
O'	Periapsis-aligned orbit frame, $\hat{\mathbf{Z}}_O = \hat{\mathbf{h}}$, $\hat{\mathbf{X}}'_O$ from planet center to periapsis
O''	Frame parallel to O’, centered in orbit ellipse center
$p_{acquisition,N_{images}}$	Probability of successful CNav acquisition for N_{images}
PCI	Planet Centered Inertial frame
p_{common}	Percentage of common area between 2 images
$\mathbf{R}^O_{PCI}$	Rotation from PCI frame to O frame
T	Orbit period
$\begin{bmatrix} u_i & v_i \end{bmatrix}^T$	Coordinates of image point
$\mathbf{v}$	Velocity vector

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: This research was funded through the Network/Partnering Initiative, contract 4000135311/19/NL/MH/ic, of the European Space Agency. The funding source had no involvement in the study design, the collection, analysis and interpretation of data, the writing of the report or in the decision to submit the article for publication.

Acknowledgments

This research was funded through the Network/Partnering Initiative, contract 4000135311/19/NL/MH/ic, of the European Space Agency, France.

Appendix A. Feature detector selection and model

To select a feature detector for translation direction estimation, an analysis was performed on 10000 image pairs rendered using the DLR-internal image rendering tool, *sensorDTM*. The three feature detectors tested were AKAZE [43], BRISK [44] and ORB [45], due to their availability in python's OpenCV library.

Each matched feature pair was compared to the corresponding predicted OF vector. The first image feature of each pair is used to find the associated 3D surface point, using its image coordinates and the associated depth generated by *sensorDTM*. Eqs. (A.1) and (A.2) respectively define the position of feature j and its homogeneous coordinates in the first image frame. Eq. (A.3) provides the direction vector from the principal point of the first camera to the surface point associated with feature j. With Eq. (A.4) this direction vector is transformed into the world frame. Finally, the direction vector is normalized through Eq. (A.5). The superfix I refers to the image frame, $C1$ refers to the first camera's frame, and PCF refers to the planet-centered planet-fixed frame.

$$\mathbf{r}^I_{j1} = \begin{bmatrix} u^I_{j1} & v^I_{j1} \end{bmatrix}^T \tag{A.1}$$

$$\bar{\mathbf{r}}^I_{j1} = \begin{bmatrix} u^I_{j1} & v^I_{j1} & 1 \end{bmatrix}^T \tag{A.2}$$

$$\mathbf{v}^{C1}_{C1\to j} = \mathbf{K}^{-1}\bar{\mathbf{r}}^{I1}_{j} \tag{A.3}$$

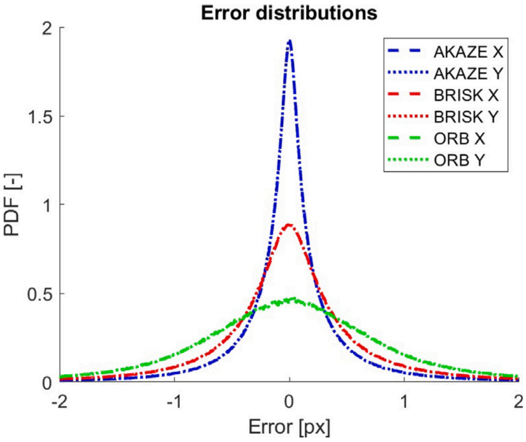


Fig. A.10. Comparison of match error distributions.

$$\mathbf{v}_{C1\to j}^{PCF} = \mathbf{R}_{C1}^{PCF}\mathbf{r}_{j1}^{C} \tag{A.4}$$

$$\hat{\mathbf{v}}_{C1\to j}^{PCF} = \frac{\mathbf{v}_{C1\to j}^{PCF}}{\left\|\mathbf{v}_{C1\to j}^{PCF}\right\|} \tag{A.5}$$

The position of the surface point in the world frame is obtained by adding this direction vector, scaled with the distance from the camera to the surface point, $d_{C1,j}$, to the position of the camera, as given by Eq. (A.6).

$$\mathbf{r}_{j}^{PCF} = \mathbf{r}_{C1}^{PCF} + d_{C1,j}\hat{\mathbf{v}}_{C1\to j}^{PCF} \tag{A.6}$$

This surface point is then reprojected onto the second image, using the pinhole camera model. The match error is defined as the difference between this prediction and the matched image feature found in the second image. Fig. A.10 shows the comparison between the error distributions obtained for the different detectors, for each image coordinate. Since AKAZE results in more precise matching than the other 2 detectors, it was selected for the application.

A.1. Performance model

In addition to the selection of the feature detector, the analysis performed also provided the performance model used to simulate OF maps. The match error magnitude was modeled as a function of the match distance (Fig. A.11(a)); the match distance was modeled as a function of the baseline (Fig. A.11(b)); and the feature density was modeled as a function of the Sun zenith angle (Fig. A.11(c)). These distributions were given as look-up tables. The angle of the match error was modeled as a uniform distribution.

A.2. Optical flow simulation

The first step in simulating the OF is to determine the number of features to be generated. The Sun zenith angle is calculated as the angle between the Sun direction vector and the camera's position vector. This angle is used to select the appropriate row in the look up table (see Table A.6). For example, if the Sun zenith angle is 34°, the following steps are applied to the row highlighted in Table A.6.

A random number between 0 and 1 is generated using a uniform distribution. This number represents the cumulative distribution associated with the number of features that will be generated, for the appropriate Sun angle range. The number of features to be generated is determined from this number. Continuing with the previous example, if the random number generated was 0.13, the corresponding number of features, n_{feat}, would be 250. Linear interpolation is used for random numbers that do not match the percentages in the look-up table. For example, for a random number equal to 0.12, the number of features would be 233.

Features are simulated by randomly generating image coordinates according to a uniform distribution. These image coordinates are used to find the associated surface point, which is then reprojected onto the second image. If a randomly generated image point does not have a corresponding surface point, which indicates a point above the lunar horizon, it is ignored and removed from the list of simulated features. Each surface point is also tested for its associated Sun angle: if the angle between the Sun direction vector and the surface point vector in PCF (Planet Centered And Fixed) is above 90°, the point is considered to be on the dark side of the Moon and removed from the list of simulated features.

The next step is to assign a match distance to each simulated feature match. The match distance is modeled as a function of each feature's baseline, using the same look-up table method as was used for the number of features.

Appendix B. Estimation of translation direction vector

Methods used to estimate egomotion from OF maps for complex movements, *i.e.* simultaneous translation and rotation, typically estimate both movement components from said OF maps. Since it is standard to include orientation sensors on board of orbiting and landing space missions, it is instead possible to use the orientation measurements to predict the rotation contribution to the OF, subtract it from the measured OF, and thus isolate the translation contribution. The resulting OF map can then be treated as the OF for purely translational movement when estimating the translation direction vector.

It can be shown that the OF for pure rotation depends only on the relative orientation of the two cameras. In other words, knowledge of the camera's position or even its absolute orientation is not required [46]. Thus, without loss of generality, when predicting the rotation component of the OF, the 3D coordinates associated with a given feature can be assumed to be $\begin{bmatrix} u_{i,1} & v_{i,1} & f \end{bmatrix}^T$, where $u_{i,1}$ and $v_{i,1}$ are the 2D coordinates of the features detected in the first image.

These points, $\mathbf{p}_{i,C1}$, are projected from the first camera plane onto the rotated camera plane, according to the pinhole camera model (Eq. (B.1)).

$$\mathbf{v}_i^{C1} = \left\|\mathbf{p}_{i,C1} - \mathbf{p}_0\right\| \tag{B.1}$$

$$\hat{\mathbf{v}}_i^{C1} = \frac{\mathbf{v}_i^{C1}}{\left\|\mathbf{v}_i^{C1}\right\|} \tag{B.2}$$

$$\hat{\mathbf{v}}_i^{C2} = \mathbf{R}_{C1}^{C2}\hat{\mathbf{v}}_i^{C1} \tag{B.3}$$

$$\mathbf{p}_i^{I} = \frac{f}{z_i^{C2}}\begin{bmatrix} x_i^{C2} \\ y_i^{C2} \end{bmatrix} \tag{B.4}$$

where $C1$ and $C2$ refer to the reference frames of camera 1 and 2, respectively; I refers to the image reference frame; $\hat{\mathbf{v}}_i$ is the unit projection vector for feature i; and $\mathbf{R}_{C1}^{C2}$ is given by

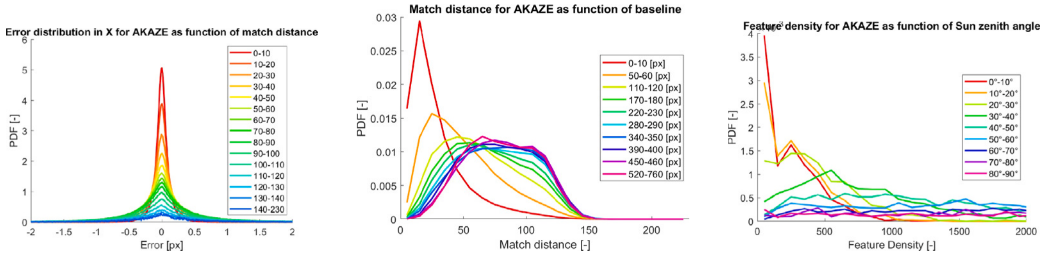
$$\mathbf{R}_{C1}^{C2} = \mathbf{R}_{W}^{C2}\mathbf{R}_{C1}^{W} \tag{B.5}$$

$$\mathbf{R}_{CX}^{W} = \begin{bmatrix} \mathbf{x}_{CX}^{W} & \mathbf{y}_{CX}^{W} & \mathbf{z}_{CX}^{W} \end{bmatrix} \tag{B.6}$$

$$\mathbf{R}_{W}^{C2} = (\mathbf{R}_{C2}^{W})^T \tag{B.7}$$

where $\mathbf{R}_{C1}^{W}$ and $\mathbf{R}_{C2}^{W}$ are the measured DCMs of the camera for the first and second images, respectively.

With the exception of translation parallel to the image plane, the OF vectors from pure translation intersect in a point, either within or outside the image frame. This point is the epipole and coincides with the projection of the translation vector onto the image plane. As such, the translation direction vector can be obtained by estimating the epipole's image coordinates and calculating the associated 3D vector through the pinhole camera model. The epipole estimation is done via



(a) Match error as a function of match distance

(b) Match distance as a function of baseline

(c) Feature density as function of Sun zenith angle

Fig. A.11. Distributions used for AKAZE performance model.

Table A.6
Section of cumulative probability look-up table for number of features as function of Sun zenith angle.

Sun zenith angle	Number of features per image						
	0	50	100	150	200	250	300
< 10°	0.00%	33.39%	39.62%	44.95%	51.44%	59.12%	67.69%
10°–20°	0.00%	16.98%	29.54%	36.59%	43.45%	50.95%	60.61%
20°–30°	0.00%	7.08%	12.90%	18.35%	25.16%	30.70%	39.60%
30°–40°	0.00%	1.57%	4.24%	6.73%	10.14%	13.00%	17.05%
40°–50°	0.00%	0.64%	1.36%	3.27%	4.45%	6.82%	9.64%
50°–60°	0.00%	0.17%	0.95%	1.90%	3.72%	5.36%	7.52%
60°–70°	0.00%	0.18%	0.45%	0.81%	1.79%	3.04%	3.94%
70°–80°	0.00%	0.95%	1.30%	1.99%	2.25%	2.60%	3.29%
≥ 80°	0.00%	2.15%	2.53%	3.09%	3.37%	4.12%	5.06%

a LSQ model within a RANSAC scheme, to account for both noise and outliers.

The most likely intersection point between 2 or more lines can be estimated by minimizing the squared perpendicular distance between this point and the intersecting lines. A line can be described as the set of point $\mathbf{p}$ such that

$$\mathbf{p} = \mathbf{a} + t\mathbf{n}, \tag{B.8}$$

where $\mathbf{a}$ is a known point in the line, $\mathbf{n}$ is the direction vector of the line, and $-\infty < t < \infty$.

The squared perpendicular distance of a point to the line, is given by

$$D(\mathbf{p};\mathbf{a},\mathbf{n}) = \left\|(\mathbf{a}-\mathbf{p}) - \left((\mathbf{a}-\mathbf{p})^T\mathbf{n}\right)\mathbf{n}\right\|^2, \tag{B.9}$$

which after some manipulation yields

$$D(\mathbf{p};\mathbf{a},\mathbf{n}) = (\mathbf{a}-\mathbf{p})^T\left(I - \mathbf{n}\mathbf{n}^T\right)(\mathbf{a}-\mathbf{p}), \tag{B.10}$$

The epipole, $\mathbf{p}_e$, of a translation OF, which coincides with the intersection point of the translation OF vectors is estimated by minimizing the sum of the squared differences between said point and each of the lines defined by the OF vectors, where $\mathbf{a}_i$ is the position of feature i, and $\mathbf{n}_i$ is the direction of the corresponding vector. Thus

$$\hat{\mathbf{p}}_e = \underset{p}{argmin}\left(\sum_{i=1}^{N_{feat}} D\left(\mathbf{p};\mathbf{a}_i,\mathbf{n}_i\right)\right) \tag{B.11}$$

Substituting Eq. (B.10) into Eq. (B.11) and reworking it, we obtain

$$\hat{\mathbf{p}}_e = \mathbf{R}^{+}\mathbf{q} \tag{B.12}$$

$$\mathbf{R} = \sum_{i=1}^{N_{feat}}\left(I - \mathbf{n}_i\mathbf{n}_i^T\right) \tag{B.13}$$

$$\mathbf{q} = \sum_{i=1}^{N_{feat}}\left(I - \mathbf{n}_i\mathbf{n}_i^T\right)\mathbf{a}_i \tag{B.14}$$

where $\mathbf{R}^{+}$ is the Moore–Penrose pseudo-inverse of $\mathbf{R}$.

The translation direction in the camera frame is obtained using the epipole's homogeneous coordinates $\mathbf{v}_H = \left[\hat{\mathbf{p}}_e : 1\right]^T$ and the camera's intrinsic matrix $\mathbf{K}$, as given by Eq. (B.15)

$$\mathbf{v}_C = \mathbf{K}^{-1}\mathbf{v}_H, \tag{B.15}$$

Finally, this vector can be rotated into the planet-centered planet-fixed frame with

$$\mathbf{v}^{PCF} = \mathbf{R}_C^{PCF}\Big|_{t2}\mathbf{v}^C, \tag{B.16}$$

where $\mathbf{R}_C^{PCF}\Big|_{t2}$ is the rotation matrix from the camera frame associated with the second image used for the OF to the world frame. If the OF vectors converge into the epipole, the translation direction is the symmetric: $-\mathbf{v}^{PCF}$.

Data availability

Data sets generated during the current study are available from the corresponding author on reasonable request.

References

[1] P. Cui, G. Xizhen, S. Zhu, W. Shao, Visual navigation using edge curve matching for pinpoint planetary landing, Acta Astronaut. 146 (2018) 171–180.
[2] C.R. Gates, H.J. Gordon, Planetary approach guidance, J. Spacecr. Rockets 2 (2) (1965) 160–166.
[3] D.A. Lorenz, R. Olds, Alexander May, A.J. May, C. Mario, M. Perry, E. Palmer, M. Daly, Lessons learned from OSIRIS-REx autonomous navigation using natural feature tracking, 2017, pp. 1–12, http://dx.doi.org/10.1109/aero.2017.7943684.
[4] D. Carpenter, The far encounter planet sensor, Tech. Rep., 1969.
[5] J.R. Scull, Mariner Mars 1969 navigation, guidance, and control, IFAC Proc. Vol. 3 (1) (1970) 163–174.
[6] W. Breckenridge, Mariner Mars 1969 optical approach navigation, in: 8th Aerospace Sciences Meeting, 1970, p. 70.
[7] N. Jerath, H. Ohtakay, Mariner IX optical navigation using Mars lit limb, J. Spacecr. Rockets 11 (7) (1974) 505–511.
[8] S. Synnott, A. Donegan, J. Riedel, J. Stuve, Interplanetary optical navigation-Voyager Uranus encounter, in: Astrodynamics Conference, 1986, p. 2113.
[9] J. Riedel, W. OWEN, J. Stuve, S. Synnott, R. Vaughan, Optical navigation during the Voyager Neptune encounter, in: Astrodynamics Conference, 1990, p. 2877.

[10] W.M. Owen Jr., P.J. Dumont, C.D. Jackman, Optical Navigation Preparations for New Horizons Pluto Flyby, Jet Propulsion Laboratory, National Aeronautics and Space …, Pasadena, CA, 2012.

[11] D.S. Nelson, E.J. Lessac-Chenen, J.Y. Pelgrift, C.D. Adam, F.J. Pelletier, J.R. Spencer, M.E. Holdridge, H.A. Weaver, C.B. Olkin, S.A. Stern, Optical navigation preparations for the new horizons kuiper-belt extended mission, J. Astronaut. Sci. 67 (2020) 1169–1188.

[12] J. Riedel, S. Bhaskaran, S. Synnott, W. Bollman, G. Null, An automomous optical navigation and control system for interplanetary exploration missions, 1996.

[13] M.D. Rayman, P. Varghese, D.H. Lehman, L.L. Livesay, Results from the deep space 1 technology validation mission, Acta Astronaut. 47 (2–9) (2000) 475–487.

[14] D.W. Smith, B.A. Lampkin, Sextant Sighting Measurements from on Board the Gemini XII Spacecraft, vol. 4952, National Aeronautics and Space Administration, 1968.

[15] J. Goodman, Apollo 13 guidance, navigation, and control challenges, in: AIAA Space 2009 Conference & Exposition, 2009, p. 6455.

[16] R. Walsh, J. Ferguson Jr., The United States air force manual space navigation experiment on skylab/DOD/NASA skylab experiment T-002, in: Astrodynamics Conference, 1976, p. 831.

[17] J.A. Christian, E.G. Lightsey, Review of options for autonomous cislunar navigation, J. Spacecr. Rockets 46 (5) (2009) 1023–1036.

[18] S.V. Dyblenko, Autonomous optoelectronic navigation for satellites on the basis of image motion analysis, in: Optical Technologies for Communications, vol. 5485, SPIE, 2004, pp. 145–152.

[19] J.A. Christian, Initial orbit determination from only heading measurements, J. Spacecr. Rockets 60 (4) (2023) 1169–1175.

[20] J.A. Christian, L. Hong, P. McKee, R. Christensen, T.P. Crain, Image-based lunar terrain relative navigation without a map: Measurements, J. Spacecr. Rockets 58 (1) (2021) 164–181.

[21] B. Leroy, G. Medioni, E. Johnson, L. Matthies, Crater detection for autonomous landing on asteroids, Image Vis. Comput. 19 (11) (2001) 787–792.

[22] G.F. Trigo, B. Maass, H. Krüger, S. Theil, Hybrid optical navigation by crater detection for lunar pin-point landing: Trajectories from helicopter flight tests, CEAS Space J. 10 (4) (2018) 567–581.

[23] B. Maass, S. Woicke, W.M. Oliveira, B. Razgus, H. Krüger, Crater navigation system for autonomous precision landing on the moon, J. Guid. Control Dyn. 43 (8) (2020) 1414–1431.

[24] A.E. Johnson, Y. Cheng, J.F. Montgomery, N. Trawny, B. Tweddle, J.X. Zheng, Real-time terrain relative navigation test results from a relevant environment for Mars landing, in: AIAA Guidance, Navigation, and Control Conference, 2015, p. 0851.

[25] Y. Cheng, J. Goguen, A. Johnson, C. Leger, L. Matthies, M.S. Martin, R. Willson, The Mars exploration rovers descent image motion estimation system, IEEE Intell. Syst. 19 (3) (2004) 13–21.

[26] D.A. Lorenz, R. Olds, A. May, C. Mario, M.E. Perry, E.E. Palmer, M. Daly, Lessons learned from OSIRIS-REx autonomous navigation using natural feature tracking, in: 2017 IEEE Aerospace Conference, IEEE, 2017, pp. 1–12.

[27] A. Johnson, S. Aaron, J. Chang, Y. Cheng, J. Montgomery, S. Mohan, S. Schroeder, B. Tweddle, N. Trawny, J. Zheng, The lander vision system for mars 2020 entry descent and landing, Guid. Navig. Control. 2017 159 (JPL-CL-CL# 17-0445) (2017).

[28] A. Johnson, N. Villaume, C. Umsted, A. Kourchians, D. Sterberg, N. Trawny, Y. Cheng, E. Geipel, J. Montgomery, The Mars 2020 Lander Vision System Field Test, Jet Propulsion Laboratory, National Aeronautics and Space …, Pasadena, CA, 2020.

[29] K.A. Farley, K.H. Williford, K.M. Stack, R. Bhartia, A. Chen, M. de la Torre, K. Hand, Y. Goreva, C.D. Herd, R. Hueso, et al., Mars 2020 mission overview, Space Sci. Rev. 216 (2020) 1–41.

[30] W. Wollenhaupt, Apollo orbit determination and navigation, in: 8th Aerospace Sciences Meeting, 1970, p. 27.

[31] H. MORITA, K. SHIRAKAWA, M. UO, T. HASHIMOTO, T. KUBOTA, J. KAWAGUCHI, Hayabusa descent navigation based on accurate landmark tracking scheme, J. Space Technol. Sci. 22 (1) (2006) 1_21–1_31.

[32] T. Ishida, S. Fukuda, K. Kariya, H. Kamata, K. Takadama, H. Kojima, S. Sawai, S. Sakai, Vision-based navigation and obstacle detection flight results in SLIM lunar landing, Acta Astronaut. 226 (2025) 772–781.

[33] IM-2 updates, 2025, https://www.intuitivemachines.com/im-2. (Accessed 14 April 2025).

[34] S.K. Biswas, H.B. Hablani, Ground based navigation of spacecraft in lunar transfer trajectory, with application to Chandrayaan-2, in: D. Choukroun, Y. Oshman, J. Thienel, M. Idan (Eds.), Advances in Estimation, Navigation, and Spacecraft Control, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 371–390.

[35] H. Zhang, J. Li, Z. Wang, Y. Guan, Guidance navigation and control for Chang'E-5 powered descent, Space: Sci. Technol. (2021).

[36] K.G. Derpanis, Overview of the RANSAC algorithm, Image Rochester NY 4 (1) (2010) 2–3.

[37] M.T. Zuber, D.E. Smith, M.M. Watkins, S.W. Asmar, A.S. Konopliv, F.G. Lemoine, H.J. Melosh, G.A. Neumann, R.J. Phillips, S.C. Solomon, et al., Gravity field of the moon from the gravity recovery and interior laboratory (GRAIL) mission, Science 339 (6120) (2013) 668–671.

[38] C. Paproth, E. Schlüßler, P. Scherbaum, A. Börner, SENSORpp: Simulation of remote sensing systems from visible to thermal infrared, in: ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, (1) 2012, pp. 257–260.

[39] S. Theil, N.A. Ammann, F. Andert, T. Franz, H. Krüger, H. Lehner, M. Lingenauber, D. Lüdtke, B. Maass, C. Paproth, J. Wohlfeil, ATON (autonomous terrain-based optical navigation) for exploration missions: Recent flight test results, CEAS Space J. (2018).

[40] J. Haruyama, T. Matsunaga, M. Ohtake, T. Morota, C. Honda, Y. Yokota, M. Torii, Y. Ogawa, LISM Working Group, Global lunar-surface mapping experiment using the lunar imager/spectrometer on SELENE, Earth, Planets Space 60 (4) (2008) 243–255.

[41] S. Woicke, H. Krüger, Hardware-in-the loop testing of crater navigation system for lunar landing, in: AIAA SciTech 2024 Forum, 2024, p. 0312.

[42] H. Krüger, B. Maass, B. Razgus, M. Solari, G. Trigo, S. Woicke, PTScientists ALINA: Guidance, navigation and control system, navigation analysis report, 2018.

[43] P.F. Alcantarilla, J. Nuevo, J. Nuevo, A. Bartoli, Fast explicit diffusion for accelerated features in nonlinear scale spaces, Br. Mach. Vis. Conf. (2013).

[44] S. Leutenegger, M. Chli, R. Siegwart, BRISK: Binary robust invariant scalable keypoints, Vision (2011) 2548–2555.

[45] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: 2011 International Conference on Computer Vision, 2011, pp. 2564–2571.

[46] H.C. Longuet-Higgins, K. Prazdny, The interpretation of a moving retinal image, in: Proceedings of the Royal Society of London. Series B. Biological Sciences, vol. 208, (1173) 1980, pp. 385–397.