Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

When a Plane Is Not Enough: Multi-Beam Laser Ranging for Optical-Flow Based Velocity Estimation in Lunar Landing

Jacqueline Ulken
European Space Research and Technology Centre (ESTEC)
Pietro Fanti
Politecnico di Milano
Dario Izzo
dario.izzo@esa.int

European Space Research and Technology Centre (ESTEC)

Article

Keywords:

Posted Date: November 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11679004/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: Competing interest reported. Author DI serves as Associate Editor of this journal and was not involved in the journal's review of, or decisions related to, this manuscript. The authors declare no competing financial or non-financial interests.

When a Plane Is Not Enough

Multi-Beam Laser Ranging for Optical-Flow Based Velocity Estimation in Lunar Landing

Jacqueline Ulken[1], Pietro Fanti[2], Dario Izzo[1*]

[1] Advanced Concepts Team, European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands.
[2] Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, Milano, Italy.

*Corresponding author(s). E-mail(s): dario.izzo@esa.int;

Abstract

Small lunar landers can estimate velocity from optical flow scaled by one laser rangefinder, assuming planar terrain. Dispersed descents can reach relief that violates this assumption. Here, a few range beams across the image are joined by Delaunay triangulation into a piecewise-planar depth model, from which linear least squares gives the velocity. Rangefinders with one to 49 beams are compared in 20 simulated descents over five south-polar sites, at camera pitches up to 45°. With nine beams and 0° camera pitch, the five sites reach similar velocity errors, and their mean is 80 % lower than with one beam. Triangulation keeps improving, while a plane fitted to the same beams levels off from about nine: over relief, the single-beam failure is likely a modeling error. A few beams may thus offer a lightweight middle ground between one rangefinder and dense depth sensing. Their best placement remains open.

Keywords: Lunar landing, Optical flow, Velocity estimation, Laser rangefinder, Multi-beam ranging, Visual odometry

1 Introduction

The Moon has once again become a destination for both government and commercial missions. Agency programs such as *Chandrayaan* [1], *Chang'e* [2], *Artemis* [3] and the Lunar Gateway [4] now share the landscape with privately developed landers from ispace [5], Firefly Aerospace [6] and Intuitive Machines [7], a trend accelerated by the Google Lunar X Prize [8]. Small landers have also demonstrated onboard image-based navigation, notably JAXA's Smart Lander for Investigating Moon (SLIM), which used vision-based navigation during its precision lunar landing [9]. Commercial vehicles, however, operate under tight mass, power, and cost margins, and several recent landing attempts have exposed vulnerabilities in descent sensing and the interpretation of its measurements. In three cases, difficulties involved altimetry: ispace's first lander discarded valid rangefinder readings after crossing a crater rim [5], its second suffered a rangefinder hardware anomaly [10], and Intuitive Machines' IM-1 descended without its primary rangefinders [7].

Autonomous descent requires the lander to estimate its velocity and altitude onboard, without relying on continuous ground contact or a dedicated lunar satellite-navigation constellation. Inertial propagation alone accumulates errors, motivating complementary navigation measurements [11, 12]. Lunar landing architectures combine inertial sensing with measurements from radar altimeters, scanning and Doppler LiDAR, and image-based terrain-relative navigation [13–20]. The mass, power, and computational requirements of these architectures motivate resource-conscious alternatives for small landers. One such approach, demonstrated on aerial robots, combines a monocular camera, an IMU, and a compact rangefinder [21–24]. The camera provides optical-flow measurements, while inertial angular-rate measurements allow the rotational contribution to be accounted for. Recovering translational velocity also requires

the depths of the tracked features, since optical flow depends on the ratio of translational velocity to scene depth. Range measurements provide these depth samples, and a surface model extends them to the tracked feature locations [25–28].

The FlowToTheMoon framework [29] follows this approach, combining sparse Lucas–Kanade optical flow, inertial measurements and a single laser rangefinder to estimate lander egomotion. The range measurement parameterizes a simplified planar or spherical model of the visible surface, from which feature depths are inferred, and the resulting estimator achieves sub-m s^{-1} velocity error along a representative descent trajectory provided by ispace. That result, however, was obtained over the kind of flat terrain that mission designers select for landing sites. Dispersions or partial failures can bring a vehicle over terrain it was never meant to land on, as when IM-1, navigating without its laser altimeter [7], touched down on a slope of about 12° beside the rim of a 1.2 km crater and came to rest tilted by roughly 30° [30]. Over such terrain, craters, slope changes, and other local structures can violate the assumed surface geometry, making feature depths inferred from a model anchored to a single range sample unreliable.

Rather than replacing this lightweight sensor setup with a dense three-dimensional sensor, this study distributes a limited number of laser range measurements across the camera field of view. The beam footprints are connected into a piecewise-planar surface by Delaunay triangulation, so that the depth model follows the terrain beneath the lander instead of relying on a single sample. Multi-beam ranging has flight heritage on larger landers, including the multi-beam fault-tolerant navigation of Chang'E-3 [31, 32] and the multi-beam Terminal Descent Sensor of the Mars Science Laboratory [33]; the question addressed here is whether a lightweight laser-based equivalent delivers similar benefits within the constraints of a small lunar lander. To answer it, single- and multi-beam configurations are compared along the full descent, over different types of terrain. We assess how the benefit of spatially distributed range measurements depends on

terrain relief, camera orientation, and beam count, and compare a single-plane model with a piecewise-planar reconstruction using the same multiple range measurements.

2 Results

We evaluate whether spatially distributed laser range measurements improve optical-flow-based velocity estimation over non-planar lunar terrain. The estimator combines sparse optical flow with known angular rates and a range-based model of scene depth to recover the lander's translational velocity. The velocity estimates are evaluated along the same simulated landing descent profile, flown over five lunar sites ranging from flat terrain to steep relief, with camera mounting pitches between 0° and 45° (Section 4.1). We compare rangefinders with one to 49 beams against two references that read depth directly from the dense LiDAR snapshot: *Full* and ground-truth depth, which also reprojects the tracked features to the temporal midpoint of each frame pair (Section 4.3.1). Unless stated otherwise, the beams follow the circle-packing layout and are joined by Delaunay triangulation to reconstruct a continuous piecewise-planar model of the surface, while the one- and two-beam cases use a slope-estimation planar model (Section 4.2.3). The error is the norm e_v of the velocity error (Section 4.3.3).

2.1 Effect of beam count and surface representation

As Table 1 shows, the error of the single-beam baseline varies with the landing site: with the camera at 0° mounting pitch, its mean rises from $0.24\,\mathrm{m\,s^{-1}}$ at *Flat* to $2.23\,\mathrm{m\,s^{-1}}$ at *Cliff*. With a 9-beam rangefinder (circle-packing layout, Delaunay triangulation), the errors at the five sites become similar in size, and the average over all sites drops by 80 %.

Nine beams thus bring all five sites to a similar low error, raising the question of how many are actually needed. We next pool the frame-wise errors over all sites and camera pitches and report their median and interquartile range. As Figure 1a shows,

Table 1: Velocity estimation errors per landing site for the single-beam baseline and a 9-beam rangefinder (circle-packing layout, Delaunay triangulation), with the camera at 0° mounting pitch. Errors are averaged over the 259 frame pairs of each descent, and the last row over the five descents.

Landing site	Mean e_v ($\mathrm{m\,s^{-1}}$)		Reduction (%)
	1 beam	9 beams	
Flat	0.24	0.15	34.8
Crater	0.62	0.18	70.6
Incline	0.42	0.18	56.4
Peak	1.62	0.19	88.3
Cliff	2.23	0.30	86.5
All sites	1.03	0.20	80.3

the median error generally decreases as beams are added. The error falls fastest over the first five beams and then keeps decreasing slowly, with no clear elbow point. This decrease is also not monotonic: three beams do worse than two.

The number of beams alone, however, does not explain the gain. A single plane fitted to the same beams levels off after about nine beams, and fitting it to the dense depth at all tracked features (*Full*) does not remove the remaining velocity error. Its depth error also changes little with the number of beams (Figure 1b). With triangulation, instead, both errors keep falling. Additional range samples therefore yield different gains depending on the surface model used to infer feature depths.

The association between depth and velocity errors also changes with beam count. The Spearman rank correlation between the frame-wise depth and velocity errors, pooled over all 20 scenarios, decreases from 0.76 with one beam to 0.50 with nine triangulated beams and 0.12 with 49 beams.

2.2 Dependence on terrain and camera orientation

The gain varies across sites, as shown by the per-site curves in Figure 2. At *Cliff*, the median falls from $2.59\,\mathrm{m\,s^{-1}}$ with one beam to $0.34\,\mathrm{m\,s^{-1}}$ with nine, whereas at *Flat* the curve changes little.

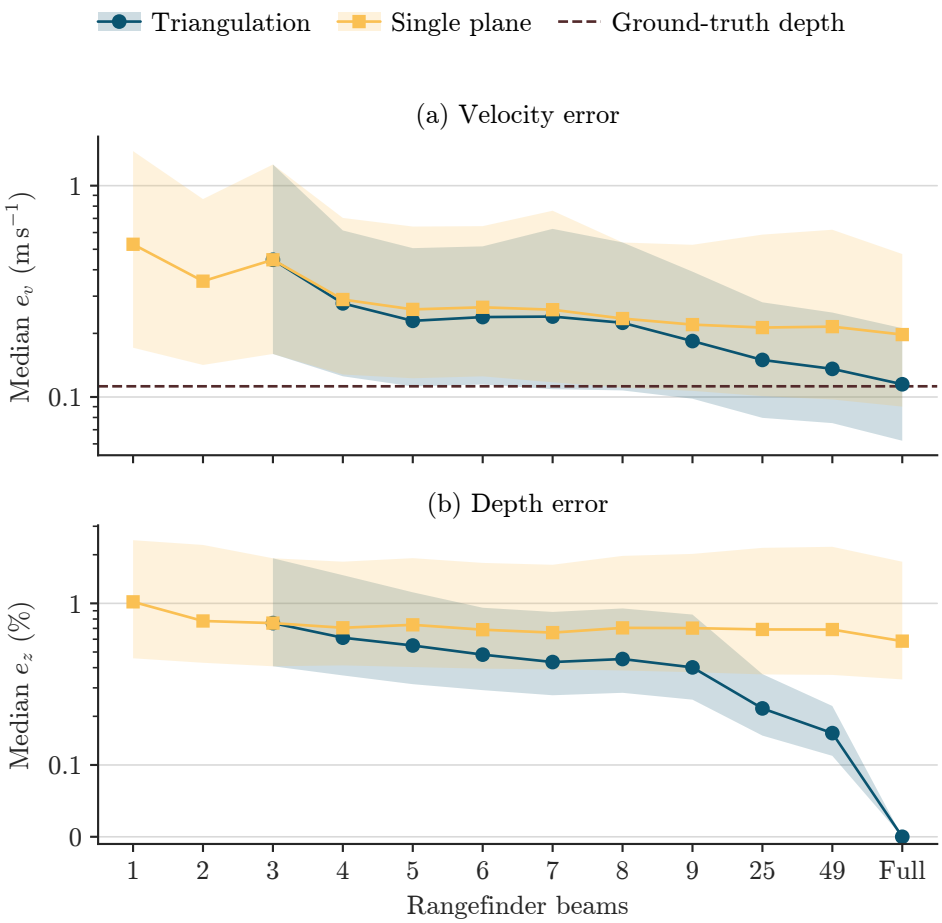


Fig. 1: Velocity and depth errors as a function of the number of beams, over all landing scenarios and camera mounting pitches with Delaunay triangulation (teal circles) and with a single plane fitted to the same beams (gold squares). The shaded areas span the 25th–75th percentiles of each. *Full* denotes the reference configuration using dense LiDAR depth directly at the tracked features, or a single plane fitted to those depths. With one and two beams, the single plane is the slope-estimation planar model.
(a) Median of the velocity error norm, $e_v = \|\boldsymbol{v}_{\text{est}} - \boldsymbol{v}_{\text{gt}}\|$. The dashed line is the median error with ground-truth depth.
(b) Median of the relative inverse-depth error e_z at the tracked features; with *Full*, directly assigned feature depths have no error by construction.

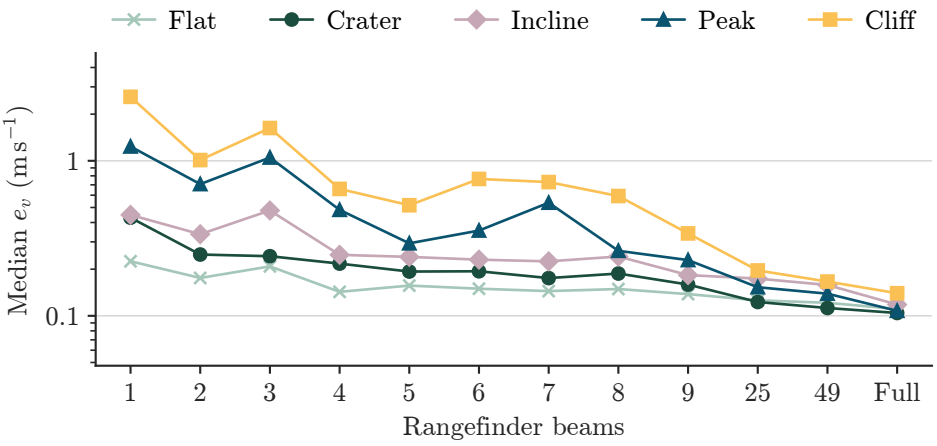


Fig. 2: Median velocity estimation error norm e_v as a function of rangefinder beam count for each landing site, over all camera mounting pitches. Sites with sharper relief, such as *Cliff*, benefit more from more beams.

These curves pool all camera mounting pitches. At the two extremes of relief, *Flat* and *Cliff*, the curves at 0° and 45° pitch show how relief and camera orientation combine (Figure 3). At *Flat*, they depend little on the number of beams or pitch, except for the single beam at 45°. At *Cliff*, they rise from 0° to 45°, so more beams are needed for the same median error.

2.3 Error evolution during descent

Pooled statistics also hide when errors arise during a descent and in which component. Even in the easiest scenario, *Flat* with the camera at 0° pitch, where nine beams lower the mean error by only about a third (Table 1), the errors with one and nine beams stay close for most of the descent (Figure 4). In a short burst late in the descent, however, the single-beam vertical error reaches 0.68 m s^{-1} in single frames, versus at most 0.17 m s^{-1} with nine beams. These maxima are calculated from the unsmoothed

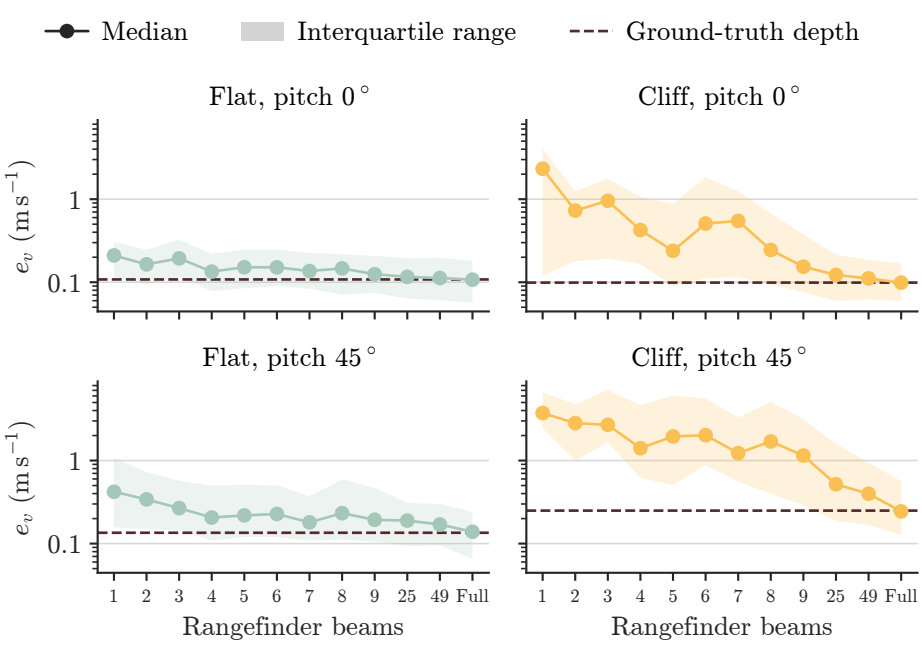


Fig. 3: Velocity estimation error as a function of rangefinder beam count at the *Flat* and *Cliff* sites, for camera mounting pitches of 0° and 45°. Median of the velocity error norm e_v; the shaded area spans the 25th–75th percentiles, and the dashed line is the median error with ground-truth depth. *Full* denotes the reference configuration using the full LiDAR depth information.

frame-wise errors, while the figure shows running medians over nine frames. The additional beams thus make the estimate more robust by suppressing its largest vertical errors, in a way that the mean error alone does not reveal.

2.4 Sensitivity to beam placement

The preceding results use the circle-packing layout. With the ragged grid, nine triangulated beams do barely better than one beam, with a median error of 0.48 m s^{-1} against 0.53 m s^{-1} over all scenarios. Circle packing gives a median error of 0.18 m s^{-1} with the same beam count. At *Cliff*, pooled over camera mounting pitches, the nine-beam median errors are 2.30 m s^{-1} with the ragged grid and 0.34 m s^{-1} with circle packing.

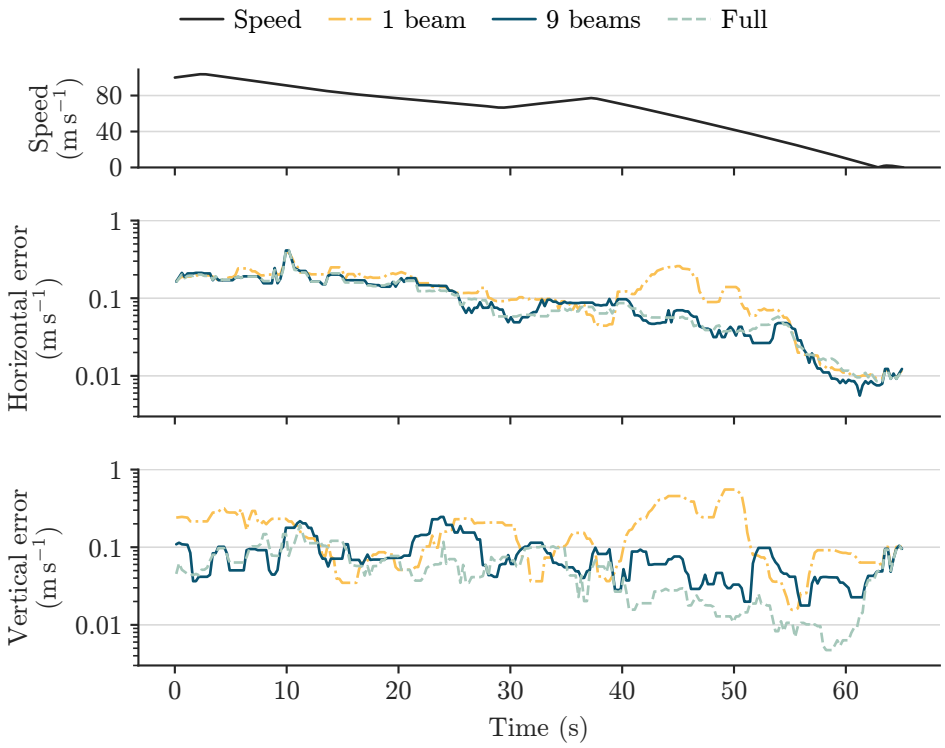


Fig. 4: Velocity estimation error during landing at *Flat*, with the camera at 0° mounting pitch. Even in this easiest scenario, the additional beams increase robustness in a way that aggregated results hide.

The ragged grid puts its outer beams on the image border, so its convex hull covers the whole view and no feature depth is extrapolated; still, it gives higher velocity errors than circle packing. In an additional check at *Cliff*, with nine beams and camera mounting pitches of 15° and 30°, the same grid shrunk to span 70 % of the image width and height came close to circle packing, and shrinking it to 50 % raised the error again. The benefit of additional beams therefore depends on their placement, not only on their number.

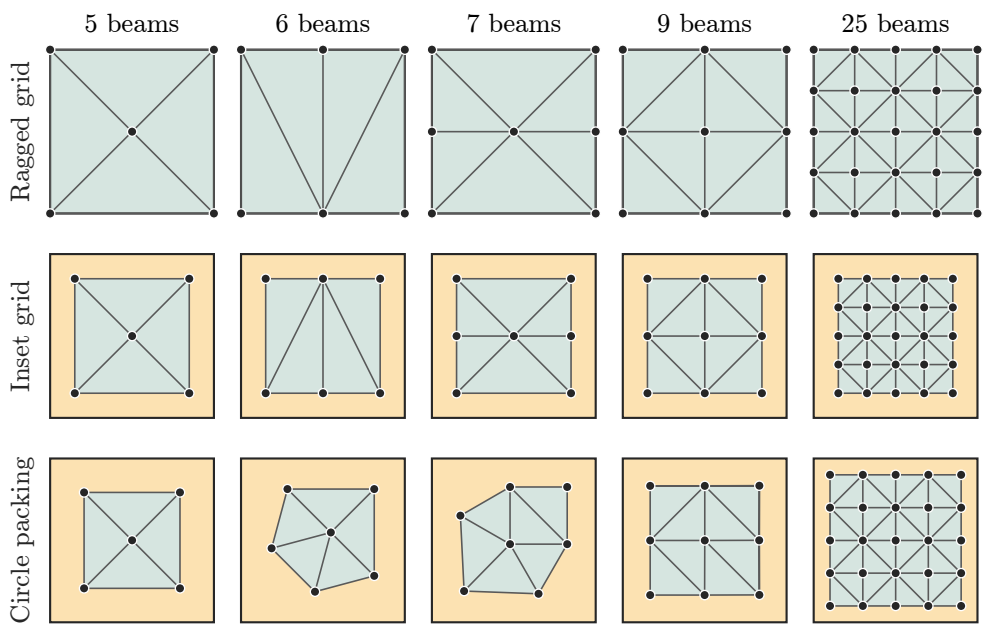


Fig. 5: Beam positions of the layouts. The rows show the ragged grid (top), the same grid shrunk to span 70 % of the image width and height (middle), and the circle-packing layout (bottom). Inside the convex hull of the beams (mint), inverse depth is interpolated within the triangles; outside it (gold), it is extrapolated.

3 Discussion

In the simulated descents, a lightweight multi-beam rangefinder makes the velocity estimate largely insensitive to the terrain under the lander. A few beams, spread across the field of view and joined by Delaunay triangulation, bring the five landing sites to a similar error: with nine beams and the camera at 0° pitch, the mean error over all sites is 80 % lower than with one beam (Table 1). As beams are added, the triangulated estimate keeps improving toward the error obtained with the dense LiDAR depth (*Full*), while fitting a single plane levels off from about nine beams, even when it is fitted to the dense depth at every tracked feature (Figure 1). Over rough terrain, the single-beam failure is therefore likely a modeling error more than a shortage of

depth samples: a plane *is not enough*, as it cannot follow the relief. In contrast, the triangulated surface follows it between the beams.

The change in how the surface is determined may also explain why three beams can perform worse than two (Figure 1a). With two beams, one degree of freedom in plane orientation remains to fit optical flow jointly with velocity. Three non-collinear range measurements instead fix the plane, even when it does not represent the terrain around the tracked features. This interpretation remains tentative, however, because the transition also changes the velocity estimation from nonlinear optimization with a robust loss to ordinary linear least squares (Section 4.2.4).

In principle, because the translational part of the optical flow scales with inverse depth (Equation (1)), errors in the inferred feature depths perturb the motion-field inversion and can bias the velocity estimate. While the depth model is coarse, this error grows with relief and camera tilt: with one beam, it is several times larger at *Cliff* than at *Flat*, and it rises as the camera is pitched toward 45°. The decreasing rank correlation between depth and velocity errors is consistent with depth-model errors becoming less dominant as more beams are added, although this correlation alone does not separate the contributions of depth errors, optical-flow errors, and feature geometry. Over *Flat*, the largest errors that the additional beams remove are vertical (Figure 4), in the component that the descent must bring close to zero for a soft touchdown.

Compared to the baseline single-beam approach [29], the multi-beam ranging simplifies the estimation. The baseline either assumes level terrain, which fails on slopes, or estimates slopes from velocity, which makes the inversion nonlinear and can be unstable. With three beams or more, the measured ranges fix the depth model, and the velocity follows from linear least squares, with no plane parameters, bounds, or initial guess (Sections 4.2.3 and 4.2.4). This comes at little computational cost: on one core of a workstation processor (Section 4.3.2), a frame takes about 21 ms with nine

beams, no more than with one, most of it spent on the optical flow, and with nine beams almost all frames finish within a tenth of the 250 ms between frames. This margin suggests that the method can run in real time on the slower processors of small landers, although it was not measured on flight-representative hardware.

Since extra beams add little computation, what remains is how many to fit and where to point them. The error has no clear elbow, but most of the gain is reached within five to nine beams (Figure 1a), and, over steep relief, a camera tilted toward 45° needs more of them (Figure 3). Beam-pattern design was not a focus of this study: only two layouts were compared (Section 4.3.1), neither optimized, and the circle-packing layout used in the Results was selected on the same scenarios. The layout comparisons show that covering the entire image with the convex hull of the beams does not necessarily minimize velocity error (Figure 5). These results suggest two competing factors: spreading the beams toward the border brings more tracked features inside the hull, where depth is interpolated rather than extrapolated, while keeping them inside brings the samples closer to the features, with smaller triangles. As the two pull in opposite directions, a layout between the extremes is likely best; circle packing appears to be one such compromise, but nothing here shows that it is optimal. Which layout is best, and how this depends on the terrain, the camera tilt, and the distribution of the tracked features, would need a dedicated study.

How far these findings carry is bounded by what the simulation leaves out. On the sensor side, the images rendered with the Planet and Asteroid Natural Scene Generation Utility (PANGU) [34] come from an ideal pinhole camera without noise or distortion, and the robustness of the single-beam baseline to added pixel noise was shown at *Flat* only [29], not with several beams or over rough terrain. The rangefinder returns the exact range of each beam, with no range noise, beam divergence or footprint, missed returns, misalignment, or timing error, and the attitude and angular rate come from the simulation rather than noisy inertial measurements. On the scenario

side, all descents follow one profile under one illumination, over five sites near the lunar south pole and at fixed camera pitches, and the profile ends in a hover above the site, before touchdown. Each velocity, finally, is estimated from a single frame pair, without temporal filtering. The results are therefore comparative evidence, obtained in simulation, of what a few beams change in the velocity estimate, and they do not establish readiness for flight.

Turning this comparison into a flight design requires, first, realistic models of the camera and the rangefinder, and then hardware-in-the-loop tests or real descent imagery. Temporal filtering or fusion with inertial data would let the estimator draw on the history of the descent. The beam pattern, fixed here for a whole descent, could also change with the camera pitch or the site. The triangulated surface, computed at every frame anyway, might feed hazard detection, a use not assessed here. Within these limits, a few range beams sit between the single rangefinder of small landers and the scanning LiDAR of large ones. Over a nominal site, they tighten an estimate that already works. Over the slopes and rims that a dispersed descent can reach, they keep it from degrading as the single-beam estimate does.

4 Methods

We evaluate whether spatially distributed range measurements improve optical-flow-based translational velocity estimation using simulated lunar descents. The movement of a camera relative to its environment can be estimated from a sequence of images by tracking the apparent motion of scene features over time. For lunar descent, the observed terrain can generally be treated as stationary between consecutive images [20, 29, 35]. Under this assumption, the apparent motion of tracked terrain features is induced by the camera's translational and rotational motion, or egomotion. The motion-field equations relate the camera's translational and angular velocities, together with the depth of each observed scene point, to its image-plane velocity.

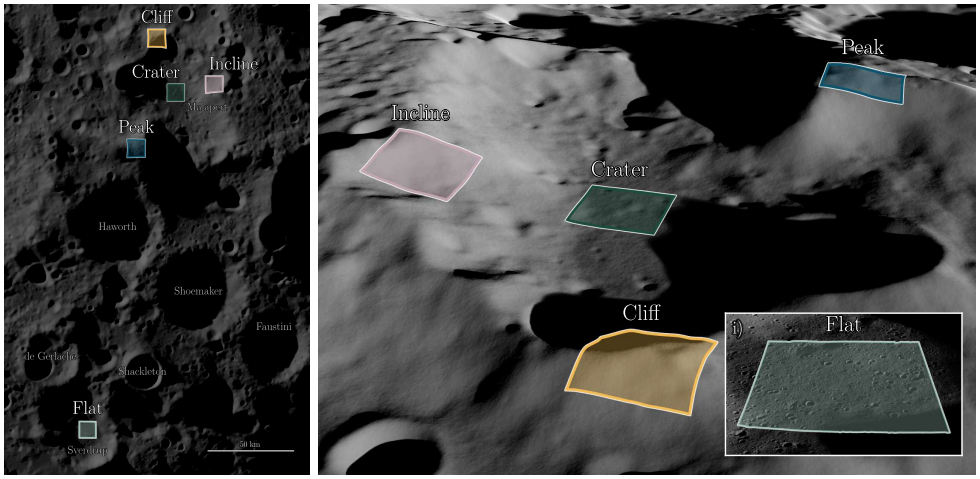
The simulated landing scenarios are described first (Section 4.1). The processing pipeline (Section 4.2) combines the motion-field model, sparse optical-flow estimation, range-based depth modeling, and inversion of the motion-field equations; a previously proposed single-beam planar depth model [29] is the baseline, and the multi-beam approaches use the same optical-flow measurements and motion-field equations, with depth reconstruction and the corresponding inversion procedure depending on the rangefinder configuration. The rangefinder configurations, evaluation protocol, and metrics follow (Section 4.3).

4.1 Simulated landing scenarios

We generated synthetic descent scenarios under controlled lunar landing conditions using the simulation framework developed by Cowan et al. [29]. Spacecraft trajectories were obtained from a six-degree-of-freedom lander model incorporating lunar gravity, a primary descent engine, and attitude-control thrusters. A minimum-fuel optimal control problem was formulated and discretized using the Hermite–Simpson direct transcription method, implemented in AMPL and solved with SNOPT [36, 37]. The reference trajectory begins at a nominal altitude of 4 km, with a downward velocity of $100\,\mathrm{m\,s^{-1}}$, and terminates at an altitude of 100 m with zero velocity, following a horizontal displacement of 1 km. We interpolated the optimized trajectory to a nominal sampling frequency of 4 Hz and transformed it into the rendering coordinate system.

To investigate the effects of terrain geometry and camera orientation, we selected the same four lunar landing sites used by Cowan et al. [29]: *Flat*, *Crater*, *Incline*, and *Peak*, plus a new fifth landing site that we called *Cliff*. Figure 6 shows their locations and surrounding terrain, while Table 2 quantifies their terrain characteristics. The five sites range from level ground to pronounced elevation changes. *Flat* is the flattest terrain identified by Cowan et al. [29] in the rendered digital elevation model (DEM), derived from data of the Lunar Orbiter Laser Altimeter (LOLA) [38]. *Flat* lies on the

floor of Sverdrup crater; the other four are in the Malapert region: *Crater* on the floor of Malapert crater, *Incline* on a slope of its rim, and *Peak* and *Cliff* on top of the rim.



(a) Ventral view (b) Local view

Fig. 6: Landing sites near the lunar south pole selected to evaluate robustness: *Flat* (■ mint), *Crater* (■ green), *Incline* (■ pink), *Peak* (■ teal), and *Cliff* (■ gold). Each site is shown from a ventral (a) and a local (b) viewpoint under realistic illumination.

Table 2: Landing sites and the relief seen by the camera at 0° pitch: slope of the plane fitted to the dense LiDAR depth of each frame, and deviation of the terrain from that plane, both as medians over the descent.

Site	Terrain	Slope (°)	Deviation from a plane (%)
■ Flat	Floor of Sverdrup crater	1	0.21
■ Crater	Floor of Malapert crater	1	0.45
■ Incline	Slope of the Malapert rim	18	0.60
■ Peak	Top of the Malapert rim	6	1.08
■ Cliff	Top of the Malapert rim	8	1.85

For each landing site, the camera was pitched at four fixed angles of 0°, 15°, 30°, and 45°, with camera roll and yaw set to zero. The camera is fixed to the lander, so its axis follows the lander's attitude: at 0° pitch it points to the local nadir at the

start and the end of the descent, and departs from it by 17° at the median, and by up to 32° in between. This yielded 20 scenarios based on the same nominal descent profile. We positioned the reference trajectory above each landing site and applied the corresponding camera orientation. We rendered synthetic images using the Planet and Asteroid Natural Scene Generation Utility (PANGU) [34] at 1024 px × 1024 px resolution and a camera field of view of 60°. Surface appearance was modeled using the Hapke bidirectional reflectance model [39], with fixed Sun azimuth and elevation of 300° and 4°, respectively, and precomputed static shadows.

For each camera pose, PANGU also generated a dense LiDAR snapshot containing the distance to the lunar surface along each camera ray. These measurements provide geometric ground truth corresponding to the rendered images. They also measure the relief of each site (Table 2): a plane is fitted to the depth of each frame at 0° pitch, the angle between its normal and the local vertical gives the slope of the terrain, and the median relative difference between the measured depth and that of the plane, over the image, gives its deviation from a plane. The simulated camera follows an ideal pinhole model, without explicitly accounting for sensor effects such as lens distortion, image noise, or rolling shutter. However, previous work showed that the underlying optical-flow navigation framework is robust to moderate simulated image noise under comparable lunar landing conditions [29]. The present study therefore focuses on isolating the effects of terrain geometry and rangefinder configuration on velocity estimation.

4.2 Velocity estimation

The proposed velocity estimation pipeline combines optical-flow measurements from consecutive camera frames, scene depth estimated from laser rangefinder measurements, and angular velocity measurements from an inertial measurement unit (IMU). These inputs are related through the motion-field equations, which are inverted using

least-squares optimization to estimate the lander's translational velocity. Fig. 7 shows an overview of the pipeline.

4.2.1 Motion-field equations

Assume a pinhole camera observing a static scene. Let $\mathcal{C}$ denote the camera frame and $\mathcal{W}$ a static world frame. For compactness, all vectors are expressed in the camera frame $\mathcal{C}$, unless otherwise stated.

A scene point P has camera-frame coordinates $\boldsymbol{p} = [X, Y, Z]^\top$, with $Z > 0$. Let $\boldsymbol{v} = [v_x, v_y, v_z]^\top \in \mathbb{R}^3$ and $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top \in \mathbb{R}^3$ denote the translational and angular velocities of the camera relative to the world frame $\mathcal{W}$, respectively.

With this convention, the camera-frame velocity of a static scene point is

$$\dot{\boldsymbol{p}} = -\boldsymbol{v} - \boldsymbol{\omega} \times \boldsymbol{p}.$$

Let f_x and f_y be the horizontal and vertical focal lengths of the camera, expressed in pixels. For the scene point P, the centered pixel coordinates are

$$x = f_x \frac{X}{Z}, \qquad y = f_y \frac{Y}{Z}.$$

Then, for optical-axis depth $z = Z$, the motion field equations [40] are

$$\begin{aligned} \dot{x} &= \frac{1}{z}(x v_z - f_x v_x) + \frac{xy}{f_y}\omega_x - \left(f_x + \frac{x^2}{f_x}\right)\omega_y + \frac{f_x}{f_y} y \omega_z, \\ \dot{y} &= \frac{1}{z}(y v_z - f_y v_y) + \left(f_y + \frac{y^2}{f_y}\right)\omega_x - \frac{xy}{f_x}\omega_y - \frac{f_y}{f_x} x \omega_z. \end{aligned} \tag{1}$$

Equivalently, they can be expressed in matrix form as

$$\boldsymbol{u} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{z} \underbrace{\begin{bmatrix} -f_x & 0 & x \\ 0 & -f_y & y \end{bmatrix}}_{\boldsymbol{L}_v(x,y)} \boldsymbol{v} + \underbrace{\begin{bmatrix} \frac{xy}{f_y} & -\left(f_x + \frac{x^2}{f_x}\right) & \frac{f_x}{f_y} y \\ f_y + \frac{y^2}{f_y} & -\frac{xy}{f_x} & -\frac{f_y}{f_x} x \end{bmatrix}}_{\boldsymbol{L}_\omega(x,y)} \boldsymbol{\omega}, \tag{2}$$

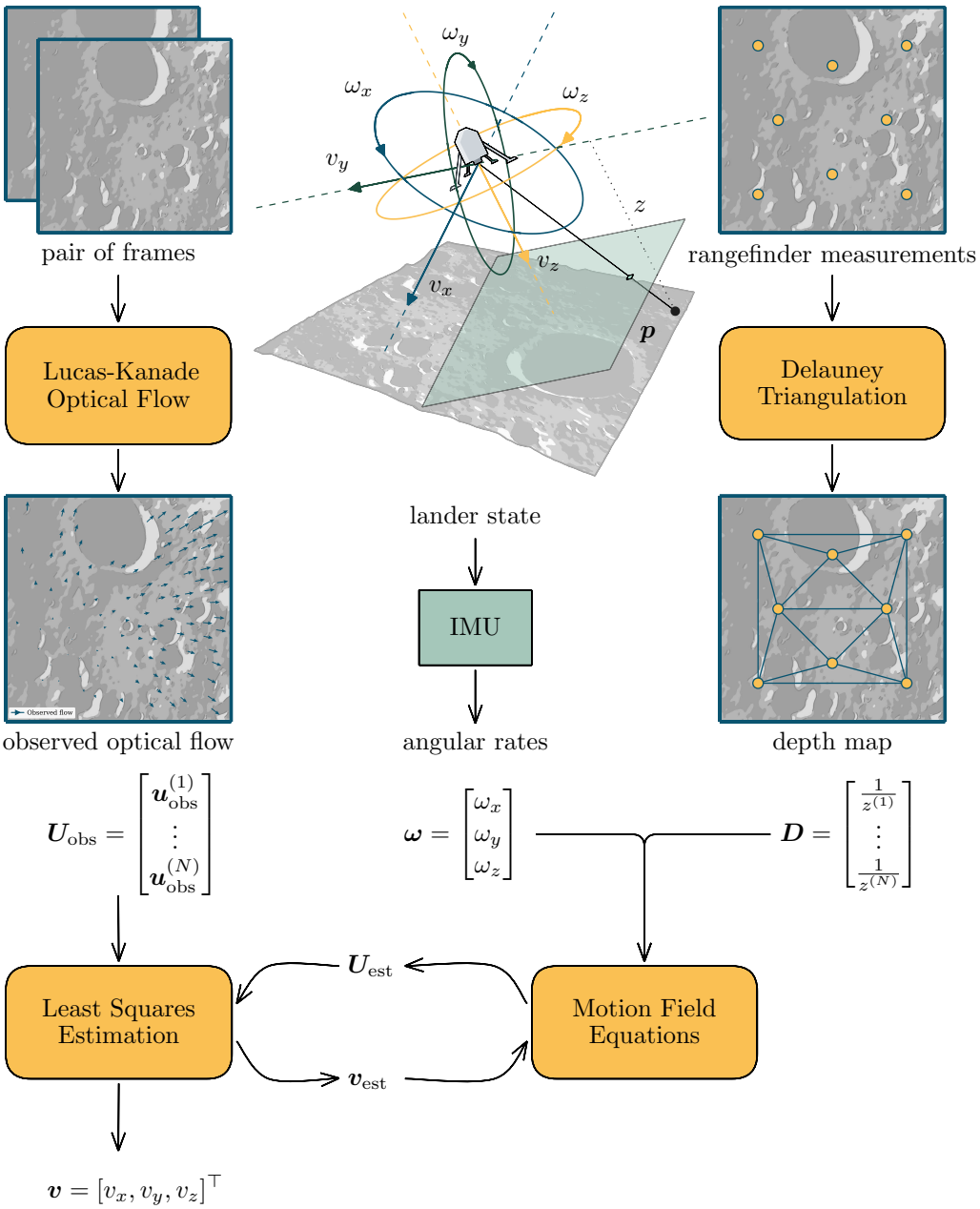


Fig. 7: Overview of the velocity estimation pipeline.

where $\boldsymbol{u}$ is the optical flow of the scene point $\boldsymbol{p}$, and $\boldsymbol{L}_v$ and $\boldsymbol{L}_\omega$ are the translational and rotational interaction matrices, respectively, relating camera motion to image-plane velocity.

The translational component depends on inverse depth $1/z$, whereas the rotational component is independent of scene depth. Given optical-flow measurements, an estimate of the depth at each tracked image feature, and angular velocity measurements from an inertial measurement unit, the translational velocity $\boldsymbol{v}$ can therefore be estimated by inverting the motion field equations.

4.2.2 Optical-flow estimation

The motion field in Equation (1) is approximated by tracking image features between consecutive camera frames.

Following the FlowToTheMoon framework [29], distinctive image features are identified with the Shi–Tomasi corner detection method [41], and their positions are tracked between consecutive frames with the pyramidal Lucas–Kanade algorithm [42, 43]. This produces a sparse optical-flow field, consisting of displacement vectors at the tracked feature locations rather than a dense velocity estimate for every image pixel.

For a feature tracked from pixel coordinates (x_k, y_k) to (x_{k+1}, y_{k+1}) across two consecutive frames, the velocity on the image plane is approximated by

$$\boldsymbol{u} \approx \frac{1}{\Delta t} \begin{bmatrix} x_{k+1} - x_k \\ y_{k+1} - y_k \end{bmatrix}, \tag{3}$$

where Δt is the mean time interval between consecutive frames, calculated from the trajectory timestamps.

The tracked feature positions and their optical flow are the observations from which the camera's translational velocity is estimated (Section 4.2.4).

4.2.3 Depth map generation

The motion-field equations (1) require the inverse depth $1/z$ at each tracked image feature. A depth map specifies the optical-axis depth z of the observed surface at each image location, corresponding to the component of the scene point's position along the camera's optical axis. Although LiDAR can produce a dense depth map, these systems impose additional hardware and computational requirements. Instead, we use sparse range measurements from one or more laser rangefinder beams to approximate the surrounding terrain with a simplified geometric model.

Each rangefinder beam provides the distance to the surface along a known direction. The rangefinders are aligned and synchronized with the camera, providing one measurement per beam for each image frame. Given the beam direction and measured range, the corresponding three-dimensional surface point in the camera frame can be reconstructed. A depth model then uses these measurements to estimate the surface depth at image positions where no direct measurement is available.

Single-beam planar model

As a baseline, we adopt the planar depth model introduced by Cowan et al. [29]. This model uses a single laser rangefinder beam aligned with the camera's optical axis to measure the distance ρ to the surface at the image center.

The model assumes that the observed terrain can be approximated by a plane. Let $\hat{\boldsymbol{n}} = [n_x, n_y, n_z]^\top$ denote the unit normal of this plane in the camera frame, and let the height H be its perpendicular distance from the camera. The plane is then described by

$$\hat{\boldsymbol{n}}^\top \boldsymbol{p} = H.$$

Substituting the pinhole camera projection equations and using $H = \rho n_z$, the inverse depth at any centered pixel coordinate (x, y) becomes

$$\frac{1}{z(x,y)} = \frac{n_x}{Hf_x}x + \frac{n_y}{Hf_y}y + \frac{1}{\rho}. \tag{4}$$

The plane orientation determines how depth varies across the image, while the central range measurement ρ provides the absolute scale.

We adopt the slope-estimation variant of the planar model, in which the plane orientation is estimated jointly with the camera's translational velocity during motion-field inversion. This allows the model to accommodate inclined terrain without requiring its orientation to be known beforehand, at the cost of two additional unknowns in the optimization: two components of $\hat{\boldsymbol{n}}$, the third following from its unit length and the chosen orientation of the normal.

Multi-beam depth models

We extend the single-beam approach by using multiple rangefinder beams distributed across the camera's field of view. Each beam provides a range measurement at a known image position, giving a set of three-dimensional surface points in the camera frame.

For two beams, we retain the planar approximation of the single-beam model, but constrain the plane to pass through both measured surface points. The additional measurement constrains the surface inclination, although two points alone do not uniquely determine a plane. The remaining degree of freedom must therefore be estimated during motion-field inversion.

For three or more non-collinear measurements, a plane can be determined directly from the measured surface geometry, without requiring an attitude-based terrain orientation or additional plane parameters in the velocity optimization. We investigate two approaches for reconstructing the depth map from these measurements.

Planar fitting. In the first approach, a single plane is fitted to all measured surface points using least squares. With more than three points, the fitted plane minimizes the squared distances to the measured points, providing a single planar approximation

of the observed terrain. The resulting plane can then be used to calculate the depth at any image position.

Delaunay triangulation. In the second approach, illustrated in Figure 8, the measured points are connected using Delaunay triangulation in the image plane. This divides the measured region into non-overlapping triangles whose vertices correspond to rangefinder measurements. Each triangle defines a local planar surface in three dimensions, allowing the reconstructed terrain to follow changes in elevation and slope that cannot be represented by a single plane.

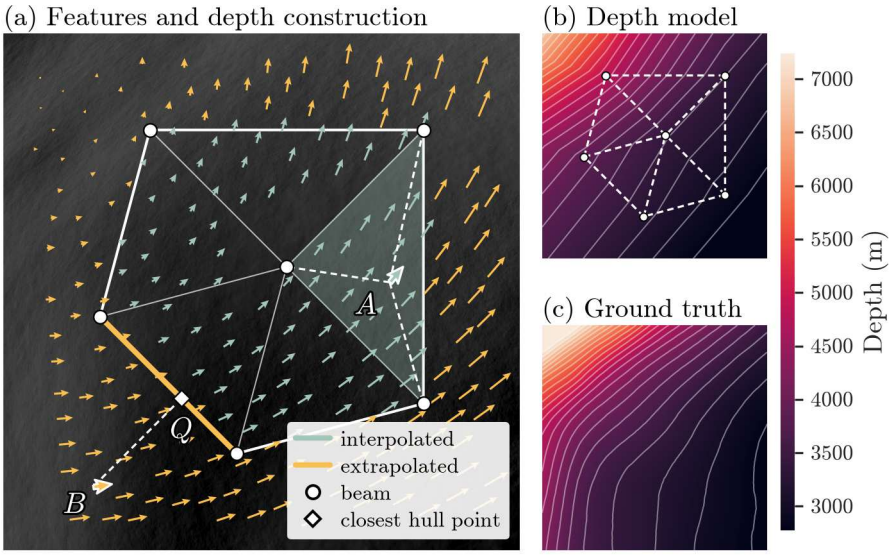


Fig. 8: Example of depth assignment by the *Delaunay triangulation* approach. (a) Features inside the convex hull of the beams (mint), such as A, interpolate inverse depth linearly within their triangle. Features outside it (gold), such as B, take the inverse depth at the closest boundary point Q and extend it with the gradient of the plane fitted to all beams. (b) The resulting depth model, with the triangulation dashed. (c) Ground-truth depth. Contours every 200 m.

For an image position inside a triangle, depth is obtained by intersecting the corresponding camera ray with the plane defined by that triangle's three measured surface points. Outside the convex hull of the measurements, inverse depth is extrapolated

from the nearest point on the convex-hull boundary, determined in normalized image coordinates. The inverse depth at this boundary point is obtained by linear interpolation along the corresponding edge and extended to the queried position using the gradient of a global planar inverse-depth model fitted to all range measurements by least squares. This approach preserves the interpolated inverse depth at the boundary, ensuring a continuous transition between interpolation and extrapolation.

With exactly three non-collinear measurements, both approaches give the unique plane through the three points. With additional measurements, however, they differ: planar fitting represents the entire observed region with one plane, whereas triangulation constructs a piecewise-planar surface that can represent more complex terrain.

4.2.4 Motion-field inversion

The optical-flow measurements and estimated depths can now be combined to recover the camera's translational velocity by inverting the motion-field equations. Each tracked feature provides two constraints, corresponding to its horizontal and vertical image-plane velocities. Since the angular velocity is known, only the three translational velocity components remain unknown when the depth model is fixed. In principle, two tracked features are sufficient to determine these components, provided their constraints are independent. In practice, we use more features to reduce the influence of measurement errors through least-squares estimation, following Cowan et al. [29].

For N tracked features, the individual motion-field equations (2) are combined into a single system,

$$\boldsymbol{U} = \boldsymbol{A}\boldsymbol{v} + \boldsymbol{B}\boldsymbol{\omega}, \tag{5}$$

where

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_{\text{obs}}^{(1)} \\ \vdots \\ \boldsymbol{u}_{\text{obs}}^{(N)} \end{bmatrix}, \qquad \boldsymbol{A} = \begin{bmatrix} \frac{1}{z^{(1)}} \boldsymbol{L}_v^{(1)} \\ \vdots \\ \frac{1}{z^{(N)}} \boldsymbol{L}_v^{(N)} \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{L}_\omega^{(1)} \\ \vdots \\ \boldsymbol{L}_\omega^{(N)} \end{bmatrix}.$$

Here, $\boldsymbol{U} \in \mathbb{R}^{2N}$ contains the observed image-plane velocities, while $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{2N \times 3}$ describe the translational and rotational contributions, respectively.

The estimation is performed for each pair of consecutive frames, with the resulting velocity associated with the temporal midpoint between them. The feature positions at this midpoint are approximated by the average of their tracked positions in the two frames. We obtain the corresponding spacecraft attitude, angular velocity, and range measurements by linearly interpolating the simulated states at the two frame timestamps.

Linear least-squares estimation

For a fixed depth model, the rotational contribution can be subtracted from the observed optical flow, leaving a linear system in the three unknown translational velocity components. The velocity is estimated by minimizing the squared residuals between the observed and predicted image-plane velocities:

$$\boldsymbol{v}_{\text{est}} = \arg\min_{\boldsymbol{v}} \|\boldsymbol{A}\boldsymbol{v} - (\boldsymbol{U} - \boldsymbol{B}\boldsymbol{\omega})\|_2^2 = \boldsymbol{A}^{+} (\boldsymbol{U} - \boldsymbol{B}\boldsymbol{\omega}), \tag{6}$$

where $\boldsymbol{A}^{+}$ denotes the Moore–Penrose pseudoinverse. The system is solved using NumPy's `linalg.lstsq` [44] with its default singular-value threshold (`rcond=None`). All feature constraints are weighted equally.

Nonlinear least-squares estimation

When the plane orientation is estimated jointly with velocity, the depth at each tracked feature depends on the unknown surface inclination. Since the plane normal is a unit vector, only two of its three components are independent. Estimating these

alongside the three translational velocity components therefore results in a nonlinear optimization problem with five unknowns, requiring at least three tracked features.

For each candidate solution, the corresponding depth model predicts the motion field, which is then compared with the observed optical flow. The parameters are iteratively adjusted to minimize these residuals. SciPy's `optimize.least_squares` [45] is used with the trust-region reflective algorithm and a Huber loss to reduce the influence of large optical-flow errors. The loss is applied to the horizontal and vertical displacement residuals individually, with the default Huber transition scale of 1 px.

Constraints and initialization

The velocity components are unconstrained, while the two estimated components of the plane normal are each bounded to $[-0.9, 0.9]$. The convergence tolerances are set to `ftol` = `xtol` = `gtol` = 10^{-12}. For the first frame pair, the initial velocity is $(1, 1, 1)^\top$ m s^{-1}, and the two plane-normal components are initialized to 10^{-3} and -10^{-3}, respectively. For subsequent frame pairs, the previous solution is used as the initial guess.

Both estimation methods account for the known camera attitude when converting between camera-frame and inertial-frame velocities. The resulting translational velocity estimates are expressed in the simulation's inertial reference frame.

4.3 Evaluation

4.3.1 Rangefinder configurations

Multi-beam rangefinders were simulated by selecting fixed beam directions within the camera's field of view. We investigated two spatial layouts: a *ragged grid*, which distributes beams approximately uniformly over the image using rows of varying lengths, and a *circle-packing layout*, which maximizes separation between beam locations by

fitting non-overlapping circles within the image boundaries. Both layouts were evaluated with 1 to 9, 25, and 49 beams. We selected the circle-packing layout for the main comparisons because it produced lower errors in the evaluated scenarios. The single-beam configuration measures along the camera's optical axis and serves as the baseline.

Rather than simulating each rangefinder separately, we extracted measurements directly from the dense PANGU LiDAR snapshots. For each image frame and beam direction, we used the corresponding pixel's slant range as the rangefinder measurement, without spatial interpolation or added measurement noise. Beam positions remained fixed relative to the camera throughout each trajectory. The resulting measurements were used to reconstruct the depth models described in Section 4.2.3.

We evaluated two additional reference configurations: *Full*, which samples depth directly from the dense LiDAR data at the tracked features, and *ground-truth depth*, which additionally uses the simulated scene geometry to reproject tracked features to the temporal midpoint of each frame pair. These reference cases do not represent finite-beam rangefinders. To separate the effect of the surface model from that of the number of samples, the single plane was also fitted to the dense LiDAR depth at all tracked features.

4.3.2 Evaluation protocol

We evaluated each finite-beam count, beam layout, and corresponding depth-model variant across all 20 landing scenarios, each a descent sequence of 260 camera frames. We used the same image sequences and optical-flow method for every configuration. The beam arrangement and associated depth model varied, with the motion-field inversion procedure determined by whether the surface orientation was estimated jointly with velocity or fixed by the range measurements (Section 4.2.4).

Unlike the broader evaluation of Cowan et al. [29], which also covered Hohmann transfers and an end-to-end transfer-to-landing trajectory, we evaluate only the landing phase here, where local terrain geometry and the resulting depth estimates matter most. We compared the estimated velocities with the corresponding simulated translational velocities, expressed in the same reference frame and evaluated at the same temporal midpoints. This procedure allows us to examine the influence of beam number, spatial arrangement, terrain, and camera pitch separately or in combination. Run times were measured on one core of an Intel Xeon w7-3565X workstation processor (up to 4.8 GHz), with a single thread, over the 259 frame pairs of the *Flat* and *Cliff* descents at 0° pitch, excluding image decoding.

4.3.3 Evaluation metrics

We quantify velocity estimation accuracy by the Euclidean norm of the difference between the estimated and ground-truth translational velocities. Let $\boldsymbol{v}_{\mathrm{est}}$ denote the estimated velocity and $\boldsymbol{v}_{\mathrm{gt}}$ the corresponding ground-truth velocity. The absolute velocity error norm is

$$e_v = \|\boldsymbol{v}_{\mathrm{est}} - \boldsymbol{v}_{\mathrm{gt}}\|\,, \tag{7}$$

where $\|\cdot\|$ denotes the Euclidean norm. This metric represents the magnitude of the three-dimensional velocity error and is expressed in $\mathrm{m\,s^{-1}}$. To identify directional differences, we also split the velocity error into a horizontal error, the norm of its two horizontal components, and a vertical error, the absolute value of its vertical component, both in the inertial frame, with its z axis vertical at the landing site. To relate the velocity errors to the depth models, the relative inverse-depth error

$$e_z = \left| \frac{z_{\mathrm{gt}}}{z_{\mathrm{est}}} - 1 \right| \tag{8}$$

is also computed at each tracked feature used in the inversion, where z_{est} is the depth given by the depth model and z_{gt} is the depth from the dense LiDAR snapshot at the same feature, and its median is taken over the features of each frame. For the slope-estimation planar model, z_{est} uses the plane orientation estimated in the inversion. We summarize error distributions over the frame pairs of each trajectory and compare them across the evaluated scenarios and rangefinder configurations. For a single descent, the mean of e_v over its frames is reported. When the frames of several scenarios are pooled, the median and the 25th–75th percentiles are reported instead, since a few frames with very large errors, even with ground-truth depth, would dominate the mean.

Code availability

All the code used to obtain the results and to generate the plots presented in this paper is open-source and available on GitLab: https://gitlab.com/EuropeanSpaceAgency/flowtothemoon

Acknowledgements The authors received no specific funding for this work.

Author contributions DI and PF jointly conceptualized the study. JU developed the implementation of the multi-beam estimator based on pre-existing code developed by ESA's Advanced Concepts Team, and PF further developed the implementation. JU and PF performed all numerical experiments. DI, JU and PF analysed and interpreted the results. DI supervised the work. PF wrote the first draft of the manuscript and established its narrative structure. All authors contributed to revising the manuscript and read and approved the final version.

Competing interests Author DI serves as Associate Editor of this journal and was not involved in the journal's review of, or decisions related to, this manuscript. The authors declare no competing financial or non-financial interests.

References

[1] Kanu, N. J., Gupta, E. & Verma, G. C. An insight into india's moon mission–chandrayan-3: The first nation to land on the southernmost polar region of the moon. *Planetary and Space Science* **242**, 105864 (2024).

[2] Zhang, H., Li, J., Wang, Z. & Guan, Y. Guidance navigation and control for chang'e-5 powered descent. *Space: Science & Technology* (2021).

[3] Smith, M. *et al. The artemis program: An overview of nasa's activities to return humans to the moon. 2020 IEEE aerospace conference*, 1–10 (IEEE, 2020).

[4] Fuller, S., Lehnhardt, E., Zaid, C. & Halloran, K. Gateway program status and overview. *Journal of Space Safety Engineering* **9**, 625–628 (2022).

[5] HAKUTO-R Mission 1 — ispace — ispace-inc.com. https://www.ispace-inc.com/mission-1/. [Accessed 15-10-2026].

[6] Blue Ghost Mission 1 - Firefly Aerospace — fireflyspace.com. https://fireflyspace.com/missions/blue-ghost-mission-1/. [Accessed 15-10-2026].

[7] Karlgaard, C. D. *et al.* Navigation Doppler lidar performance assessment and trajectory reconstruction of the Intuitive Machines IM-1 lunar landing. Technical Memorandum NASA/TM-20250010290, NASA Langley Research Center (2025).

[8] Hossain, M. & Kauranen, I. Competition-based innovation: the case of the x prize foundation. *Journal of Organization Design* **3**, 46–52 (2014).

[9] Ishida, T. *et al.* Vision-based navigation and obstacle detection flight results in SLIM lunar landing. *Acta Astronautica* **226**, 772–781 (2025).

[10] HAKUTO-R Mission 2 — ispace — ispace-inc.com. https://www.ispace-inc.com/mission-2/. [Accessed 15-10-2026].

[11] Chelmins, D. T., Welch, B. W., Sands, O. S. & Nguyen, B. V. A kalman approach to lunar surface navigation using radiometric and inertial measurements. Tech. Rep., NASA, Glenn Research Center (2009).

[12] Jaskot, K. & Babiarz, A. The inertial measurement unit for detection of position. *Przeglad Elektrotechniczny* **86**, 323–333 (2010).

[13] Epp, C. D. & Smith, T. B. *Autonomous precision landing and hazard detection and avoidance technology (ALHAT). Proceedings of the IEEE Aerospace Conference*, 1–7 (2007).

[14] Frapard, B. & Mancuso, S. *Vision navigation for european landers and the npal project. Guidance, Navigation and Control Systems*, Vol. 606, 0–8 (2006).

[15] Verweld, M. J. *Relative optical navigation for a lunar lander mission. Advances in Aerospace Guidance, Navigation and Control: Selected Papers of the Second CEAS Specialist Conference on Guidance, Navigation and Control*, 661–679 (2013).

[16] Xu, W. *et al.* Navigation doppler lidar sensor for precision landing of china's chang'e-5 lunar lander. *Applied optics* **59**, 8167–8174 (2020).

[17] Li, S., Cui, P. & Cui, H. Autonomous navigation and guidance for landing on asteroids. *Aerospace science and technology* **10**, 239–247 (2006).

[18] Li, S., Cui, P. & Cui, H. Vision-aided inertial navigation for pinpoint planetary landing. *Aerospace Science and Technology* **11**, 499–506 (2007).

[19] Johnson, A. E. & Montgomery, J. F. *Overview of terrain relative navigation approaches for precise lunar landing. 2008 IEEE Aerospace Conference*, 1–10 (2008).

[20] Maass, B., Woicke, S., Oliveira, W. M., Razgus, B. & Krüger, H. Crater navigation system for autonomous precision landing on the Moon. *Journal of Guidance, Control, and Dynamics* **43**, 1414–1431 (2020).

[21] Grabe, V., Bülthoff, H. H. & Giordano, P. R. *Robust optical-flow based self-motion estimation for a quadrotor uav. 2012 IEEE RSJ International Conference on Intelligent Robots and Systems*, 2153–2159 (2012).

[22] Grabe, V., Bülthoff, H. H., Scaramuzza, D. & Robuffo Giordano, P. Nonlinear ego-motion estimation from optical flow for online control of a quadrotor UAV. *The International Journal of Robotics Research* **34**, 1114–1135 (2015).

[23] Panahandeh, G. & Jansson, M. Vision-aided inertial navigation based on ground plane feature detection. *IEEE/ASME transactions on mechatronics* **19**, 1206–1215 (2013).

[24] Wang, T., Wang, C., Liang, J., Chen, Y. & Zhang, Y. Vision-aided inertial navigation for small unmanned aerial vehicles in gps-denied environments. *International Journal of Advanced Robotic Systems* **10**, 276 (2013).

[25] Williams, J. G., Boggs, D. H. & Currie, D. G. Next-generation laser ranging at lunar geophysical network and commercial lander payload service sites. *The Planetary Science Journal* **3**, 136 (2022).

[26] Miso, T., Hashimoto, T. & Ninomiya, K. Optical guidance for autonomous landing of spacecraft. *IEEE Transactions on aerospace and electronic systems* **35**, 459–473 (1999).

[27] Ho, H. W., de Croon, G. C. H. E. & Chu, Q. Distance and velocity estimation using optical flow from a monocular camera. *International Journal of Micro Air Vehicles* **9**, 198–208 (2017).

[28] Chiodini, S., Giubilato, R., Pertile, M. & Debei, S. Retrieving scale on monocular visual odometry using low-resolution range sensors. *IEEE Transactions on Instrumentation and Measurement* **69**, 5875–5889 (2020).

[29] Cowan, S. *et al.* Vision-guided optic flow navigation for small lunar missions (2025). Preprint at https://arxiv.org/abs/2511.17720.

[30] Thomson, B. *et al. Characterization of the intuitive machines (im-1) lunar landing site near the south pole. EGU General Assembly Conference Abstracts*, EGU25–20315 (2025).

[31] Zhang, H., Li, J., Wang, Z. & Guan, Y. Guidance navigation and control for chang'e-5 powered descent. *Space: Science & Technology* (2021).

[32] Xiangyu, H. *et al.* Autonomous navigation and guidance for change-3 soft landing. *Journal of Deep Space Exploration* **1**, 52–59 (2014).

[33] Pollard, B. *et al.* Ka-band radar terminal descent sensor. Tech. Rep., NASA's Jet Propulsion Laboratory (2007).

[34] Martin, I. M., Dunstan, M. N. & Sanchez Gestido, M. *Planetary surface image generation for testing future space missions with PANGU. 2nd RPI Space Imaging Workshop*, 1–13 (Sensing, Estimation, and Automation Laboratory, Saratoga Springs, NY, USA, 2019).

[35] Shang, Y. & Palmer, P. L. Lunar lander's three-dimensional translation and yaw rotation motion estimation during a descent phase using optical navigation. *Acta Astronautica* **68**, 149–159 (2011).

[36] Fourer, R., Gay, D. M. & Kernighan, B. W. *AMPL: A Modeling Language for Mathematical Programming* 2 edn (Thomson Brooks/Cole, Pacific Grove, CA, 2003).

[37] Gill, P. E., Murray, W. & Saunders, M. A. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review* **47**, 99–131 (2005).

[38] Neumann, G. A. Lunar Orbiter Laser Altimeter (LOLA) gridded digital elevation models. NASA Planetary Data System (2010). Data set LRO-L-LOLA-4-GDR-V1.0.

[39] Hapke, B. Bidirectional reflectance spectroscopy: 1. theory. *Journal of Geophysical Research: Solid Earth* **86**, 3039–3054 (1981).

[40] Horn, B. K. P. Motion fields are hardly ever ambiguous. *International Journal of Computer Vision* **1**, 259–274 (1988).

[41] Shi, J. & Tomasi, C. *Good features to track. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 593–600 (1994).

[42] Lucas, B. D. & Kanade, T. *An iterative image registration technique with an application to stereo vision. Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vol. 2, 674–679 (1981).

[43] Bouguet, J.-Y. Pyramidal implementation of the Lucas–Kanade feature tracker: Description of the algorithm. Tech. Rep., Intel Corporation, Microprocessor Research Labs (2001). URL https://robots.stanford.edu/cs223b04/algo_tracking.pdf.

[44] Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).

[45] Virtanen, P. *et al.* SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods* **17**, 261–272 (2020).