LiDAR-Inertial Based Navigation and Mapping for Precision Landing

Timothy P. Setterfield, Robert A. Hewitt, Po-Ting Chen,
Antonio Terán Espinoza, Nikolas Trawny, Anup Katake
Jet Propulsion Laboratory
California Institute of Technology 4800 Oak Grove Dr, Pasadena, CA 91109
timothy.p.setterfield@jpl.nasa.gov

Abstract*—Future lander missions will travel to ambitious, scientifically interesting locations near rough and dangerous terrain. They will need to operate with limited prior information about the terrain, and under varying lighting conditions. Landing safely and precisely in the face of these challenges is difficult for existing vision-based landing systems, which require detailed orbital reconnaissance, *a priori* hazard maps, and impose time-of-day restrictions on landing to ensure similar lighting conditions in orbital and descent imagery. Advanced 3D imaging LiDAR systems currently under development, and originally intended for single-scan hazard detection, have the potential to be operated continuously from altitudes of up to 5 km. Used together with existing inertial measurement units (IMUs), these sensors open a path-to-flight for a full navigation and mapping system, which could replace or augment a traditional landing sensor suite. A landing system based around these sensors can perform accurate altimetry, map-relative localization (MRL), LiDAR-inertial odometry, and map refinement in an illumination-insensitive manner, over unknown or partially known terrain. This paper outlines preliminary work on a LiDAR-inertial landing system that: estimates the spacecraft trajectory during entry, descent, and landing (EDL); and maps the topography of the terrain below, for future use in hazard detection and avoidance. An incremental, factor graph based, smoothing approach is used to solve for the *maximum a posteriori* trajectory of spacecraft states. Integrated IMU measurements and features tracked in adjacent range and intensity images are used to estimate motion (LiDAR-inertial odometry). LiDAR scans are binned into motion-corrected digital elevation models (DEMs), which are matched to an existing orbital topographic map to provide absolute position information (MRL). The estimated trajectory is then used to project the LiDAR scans into the map frame, creating a variable-resolution quadtree topographic map suitable for hazard detection and avoidance. Existing topographic maps from throughout the solar system (i.e., Earth, the Moon, Mars, Ceres, Vesta, Europa, Enceladus, and Eros) are upsampled for use in EDL simulations. The Mars 2020 Lander Vision System Simulator (LVSS) is extended to simulate LiDAR-inertial data for realistic EDL trajectories. Results of the algorithm operating on the simulated data are presented. Estimated spacecraft trajectory and refined map are compared to ground truth to assess estimation accuracy.

TABLE OF CONTENTS

978-1-7281-7436-5/21/$31.00
Governemnt sponsorship acknowledged.

1. INTRODUCTION

When landing a spacecraft on another planetary body, the primary concern is safety of the mission. Early interplanetary missions targeted expansive, flat areas to accommodate large landing uncertainties. However, the most scientifically interesting regions on other planetary bodies – the craters, the canyons, the mountains, the deltas, and the lava tubes – are typically home to rugged, hazardous terrain.

For the Mars 2020 mission, visual terrain relative navigation was introduced to allow the Perseverance rover and Ingenuity helicopter to land safely in the scientifically interesting, but relatively hazardous, Jezero Crater. To make this possible, high-resolution (up to 0.5 m/px) orbital maps are used to pre-identify hazardous regions. A downward-facing camera takes images which are matched against onboard orbital maps, localizing the lander during descent. The lander is then able to divert to the nearest hazard-free region.

While this is an elegant solution for the Mars 2020 mission, it is not widely applicable to all bodies in the solar system. Orbital reconnaisance of the resolution required for pre-indentification of lander-sized hazards does not exist for many bodies of interest such as Europa, Enceladus, asteroids, and comets. As evidenced by the baselining of a scanning LiDAR for hazard detection on the Europa Lander pre-phase A project [1], this may be true even *after* a dedicated mission (i.e., Europa Clipper) has been used to map landing sites. Additionally, visual terrain relative navigation matches images from a passively lit scene, and is thus dependent on the lighting conditions at the time of image acquisition. This imposes the constraint that landing must occur when the landing site is well lit, in a similar season and a similar time of time of day as when orbital images were obtained.

These challenges can be mitigated or solved by replacing the downward-facing camera with an active, dense 3D LiDAR sensor, and the high-resolution visual orbital map with a low-resolution (e.g., 32 m/px) topographic map. LiDAR sensors provide ranging and laser return intensity over an entire field of points, creating a range and intensity image; unlike cameras, they provide direct, high-confidence depth measurements and are invariant to lighting conditions. Used together with an inertial measurement unit (IMU) and a star tracker, LiDAR could replace or augment a traditional landing sensor suite. In addition to estimation of the lander's navigation state during descent, a LiDAR-inertial based landing system can also build a high-resolution topographic map for onboard hazard detection. This means that the system will enable landing on bodies for which there is little or no *a priori* knowledge of the topography, under any lighting conditions.

The objective of the work described in this paper is to develop and mature a LiDAR-inertial based navigation and mapping system for safe and precise landing on planetary bodies. The remainder of this paper is structured as follows. Section 2 provides an overview of related work. Section 3 goes over the necessary background material drawn on to implement our technical approach. Section 4 outlines the technical approach that we took. Section 5 presents and discusses the technical results. Section 6 concludes and suggests future work.

2. Related Work

Related work on space-qualified LiDARs, visual terrain relative navigation, and LiDAR-inertial navigation, described below, are heavily drawn upon in this project. The novelty of this work in filling gaps present in the literature is also discussed below.

Space-Qualified LiDARs

JPL is overseeing contracts to develop two space-qualified LiDARs for the Europa Lander mission, to be used for hazard detection and avoidance. They are both 11.42° field of view scanning LiDARs with scan durations $\leq$ 2 s. They are baselined to perform coarse (200×200 px, 5 m/px, 25 cm 1σ accuracy) mapping at 5 km altitude and fine (2000×2000 px, 5 cm/px, 5 cm 1σ accuracy) mapping at 500 m altitude. While they are currently required to only measure range, after some development work, at least one of the designs may be able to provide laser return intensity. The possibility of using these LiDARs continuously during descent was one of the motivations behind this work.

The Osiris-REx mission's Goldeneye LiDAR uses the Advanced Scientific Concepts (ASC) 3D 128×128 px space-qualified flash LiDAR, with a nominal update frequency of 5 Hz, a range of up to 3 km inclusive (with higher range possible) [2]. It measures range (5 cm 1σ accuracy [2], [3]) and laser return intensity, and can be run at up to 30 frames/s [4]. The field of view depends on the optics, but has been tested at 1, 3, 5, and 8.6° [3], [5], [6]. This sensor was also used in the NASA Autonomous Landing Hazard Avoidance Technology (ALHAT) program [5], [6], and is the inspiration behind the flash LiDAR simulated in this R&TD.

Other space qualified LiDARs include the Osiris-REx mission's Laser Altimeter (OLA), the first scanning LiDAR to fly on a planetary mission [7], and the Jena Optronik RVS 3000-3D scanning LiDAR, in development to provide ranging to non-cooperative orbital targets [8].

Visual Terrain Relative Navigation

State-of-the-art visual terrain relative navigation (TRN) is performed on the Mars 2020 mission using the Lander Vision System (LVS). A preliminary version of these algorithms is detailed in [9]. A systems-level view of LVS is provided in [10], and results of its field test are given in [11]. In [9], the authors describe an Extended Kalman Filter (EKF) for localization with respect to map landmarks that uses a camera tightly coupled with an IMU in a manner similar to the Multi-State Constraint Kalman Filter (MSCKF) [12]. Unlike the MSCKF, a pose history does not need to be stored and landmarks are matched to the map rather than to previous frames. SIFT features or correlation-based approaches are discussed, with the latter requiring an onboard altimeter and orientation estimate to appropriately de-warp the image into the map frame. Image processing delays are mitigated by temporarily cloning the state. LVS is tested using a simulation tool called the LVS simulator (LVSS) which takes a spacecraft trajectory and visual and topographic maps as inputs, and generates images and IMU measurements as outputs. LVSS was augmented in this work for generation of LiDAR-inertial datasets.

Johns Hopkins Applied Physics Laboratory (APL) has developed APLNav to perform correlation-based map relative localization [13], [14], [15]. Visual images taken on the lander are compared to lighting-corrected orbital imagery to localize the spacecraft during descent.

The Charles Stark Draper Laboratory, as part of the NASA Safe and Precise Landing Integrated Capabilities Evolution (SPLICE) program, developed a correlation-based matching map relative localization system, and performed a flight test on a Masten Space Systems Xodiac vehicle [16].

Ted Steiner, in his PhD thesis [17], outlines a smoothing-based approach to TRN. He incorporates map landmarks as well as ad-hoc opportunistic landmarks for accuracy improvement, and uses graph-based techniques to solve for the entire trajectory. He explicitly estimates the positions of all landmarks, adding tight priors to map landmarks since their location is well-known *a-priori*. Steiner makes use of the open-source Georgia Tech Smoothing and Mapping (GTSAM) software library [18], which is also used in this work, to implement his smoothing algorithm.

LiDAR-Inertial Navigation for EDL

LiDAR-inertial navigation has been previously investigated in the context of EDL. In the aerospace community, estimation of different components of the spacecraft state during EDL is traditionally subdivided into different sensing modalities. Orientation is estimated using a star tracker and gyroscope; altitude measurement is performed using radar or a laser rangefinder and is referred to as altimetry; horizontal position is measured using visual TRN (aka map relative localization (MRL)); velocity is estimated using either a dedicated instrument such as Doppler radar or LiDAR or through velocimetry (frame-to-frame matching of images). At lower altitudes, hazard detection and avoidance (HDA) refers to the building of a local hazard map and diverting to a hazard-free region, and the term hazard relative navigation (HRN) is often used when referring to performing TRN in this local hazard map.

The ALHAT program investigated the use of LiDAR-based TRN for landing. In [6], the authors suggest using different sensors for different phases: a flash LiDAR for altimetry between 20 km and 100 m altitude, for TRN between 15 km and 5 km, and for HDA/HRN between 1 km and 100 m; a Doppler LiDAR for velocimetry and altimetry between 2.5 km and 10 m; and a laser altimeter for altimetry between 20 km and 100 m, and for TRN between 15 km and 5 km.

Some instruments were tested for their ability to perform HDA/HRN on the NASA Morpheus vertical take off and landing (VTOL) vehicle [19] at NASA Kennedy Space Center's shuttle landing facilities. In the tests outlined in [19], an ASC Goldeneye LiDAR with 1° field of view was gimballed to create a 60×60 m, 10 cm/px DEM over the landing field at approximately 250 m altitude. During 6 s scan, vehicle dynamics were "sufficiently damped" to allow for map creation. HRN was performed with respect to this local map and used to guide the spacecraft to a selected safe landing site within it.

In [20] the authors offer a survey of passive and active approaches for lunar landing for the ALHAT project. Included in this survey are frame-to-frame DEM matching and frame-to-map DEM matching. The authors' investigation includes simulated matching of contours as well as DEMs generated in a pushbroom manner, where a single degree-of-freedom LiDAR scanner sweeps back and forth perpendicular to the direction of travel (cross-track) and the along-track motion is used as the second scan degree of freedom (i.e., a pushbroom scan). The authors conclude that 2D range images provide better matching accuracy that 1D measured contours. Despite recognizing the requirement for motion compensation, the method for doing this is not discussed.

In [21], the authors describe field testing of LiDAR-based TRN with both a laser altimeter and a flash LiDAR (the ASC Goldeneye, configured with a 3° field of view) mounted on a single degree-of-freedom gimbal. Data was obtained at a test velocity of approximately 60 m/s at altitudes of 2, 4, and 8 km over Death Valley and Nevada. LiDAR point clouds, originating from multiple LiDAR scans and binned to 5 m/px were matched to topographic maps of the same resolution. To project into the UTM map frame, the ground truth flight trajectory was interpolated (i.e., motion compensation was performed with ground truth knowledge). Frame-to-map template matching was performed using a floating point matching methodology that handles missing data [22] (normalized cross-correlation); then a bi-quadratic fit was used for sub-pixel accuracy. This is identical to the method used herein for MRL. Interestingly, matching performance of the laser altimeter contours and the comparable to that of the flash LiDAR swaths was similar. This is likely because the flash LiDAR, with its 3° field of view, formed only narrow swaths which were at most slightly more informative than the laser altimeter contours.

In [23] the authors use a scanning LiDAR without feature matching to perform velocity estimation. A pseudo-Doppler technique tracks surface normals, and measures local range rate in three directions (similar to Doppler LiDAR). This technique relies on rapidly scanning a very small azimuth-elevation range twice in a row, determining the surface normal, and then using range finite difference to obtain velocity along the surface normal. While useful for determining velocity, this technique requires a specific scan pattern, and is thus of little utility for the other objectives of this work.

In our recent work [24], [25] we investigate the feasibility of performing single-scan MRL at 5 km for the Europa Lander mission. This work assumes that at the start of the scan, orientation is known to within 0.5° 3σ from a star tracker and gyroscope propagation, velocity is known to within 10 m/s 3σ from a previous visual feature-based velocimetry step, and the Europa Lander LiDAR specifications as described above. High resolution (0.5 m/px) Europan terrains were generated by artificially adding high frequency terrain content to low-resolution source data. This study indicated that localization results on par with visual MRL could be obtained using LiDAR and 32 m/px maps at 5 km altitude on Europa. In this work, we make use of the artificial terrain generation and MRL techniques from this previous work; however, while the previous work used DSENDS [26] for simulation and MATLAB for estimation, herein we use LVSS for simulation and C++ for estimation.

Novelty of this Work

In developing our LiDAR-inertial navigation and mapping algorithm, it was possible to draw on the related work outlined above. However, there were some significant gaps that were addressed in order to accomplish our objectives.

One prominent gap was absence of a LiDAR simulator within the existing EDL simulation options. To mitigate this we augmented the existing LVSS software to include the simulation flash and scanning LiDAR, outputting both range and laser return intensity data.

In order to demonstrate the onboard refinement of maps to resolutions suitable for hazard detection and avoidance throughout the solar system, we needed a diverse set of data with lander-scale pixel size. Outside of Earth, the Moon, and Mars, appropriate ground truth topographic maps were not available. To create maps of the requisite resolution, we first obtained topographic maps from eight planets at the best resolution available. We then used a technique that we pioneered in [27] to upsample 81 DEMs to 0.5 m/px.

Previous work on visual-inertial odometry, both using filtering [12], and smoothing [28], [29] have employed "structureless" methods of including opportunistic landmark correspondences, which apply pose constraints without explicitly requiring estimation of landmark positions. However, these methods apply exclusively to monocular cameras. In this work, we extend the structureless approach for inclusion of range and bearing correspondences detected by a flash LiDAR.

Previous approaches LiDAR-inertial odometry with a scanning LiDAR employ three different methods: assuming a constant velocity during the LiDAR scans [30], [31] (which is unrealistic during EDL); using an additional global shutter sensor to de-warp the LiDAR scans [32] (which increases complexity, size, weight, and power); or evaluating IMU measurement error at every IMU measurement [33] (which is computationally expensive). In this work, we develop a method for directly including feature correspondences from scanning IMU into a factor graph that is not subject to any of these assumptions or drawbacks.

While existing TRN and hazard mapping systems require special flight modes in which the spacecraft is close to nadir-facing and relatively stable, the method developed herein is designed to navigate and map during all phases of EDL, including powered approach. This is accomplished by performing LiDAR-inertial odometry and MRL throughout the entire trajectory, and by including only the highest-quality measurements.

3. BACKGROUND

This section provides some relevant background information used in the development of our approach in Section 4. It discusses flash and scanning LiDAR, the existing Lander Vision System Simulator (LVSS), and smoothing-based estimation.

Flash and Scanning LiDAR

A LiDAR instrument is used to measure the distance to a surface by recording the time of flight of a pulsed laser beam or the phase delay of a modulated beam. Often, the intensity of the reflected laser beam, which is a function of range, laser power, surface albedo, and surface normal, is also measured.

In the case of flash LiDAR, a wide laser pulse is emitted, and camera-like optics are used to simultaneously measure the laser returns over an array of azimuth and elevation

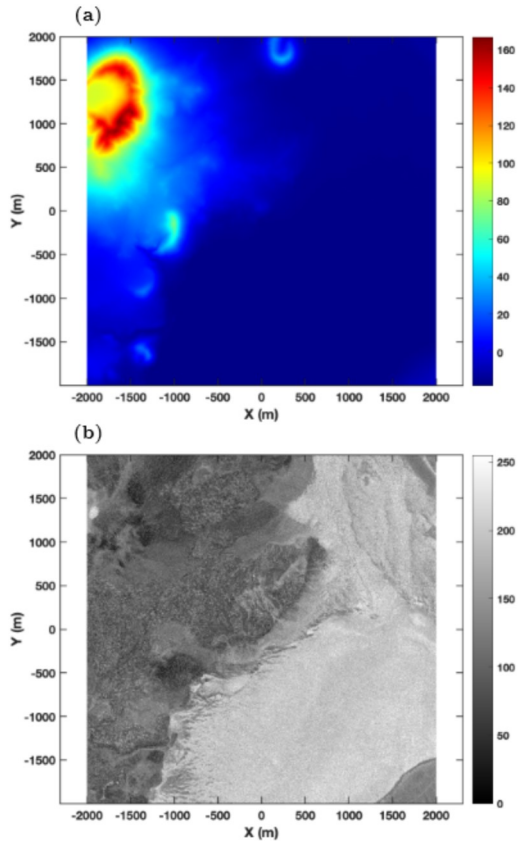


Figure 1. Input DEM (a), and SRM (b).

angles. The global shutter nature of this sensor significantly simplifies the use of the scan data, as the points do not require motion compensation. However, since a large amount of power is required to emit wide laser pulses, the maximum range and field of view of flash LiDAR are relatively low.

In the case of conventional scanning LiDAR, a narrow laser pulse is emitted and a single sensor is used to measure the laser return. A movable platform or mirror is used to rapidly scan the laser and sensor across a set of azimuth and elevation angles, following a "scan pattern". This scan pattern can be open-loop, where the set of azimuth and elevation angles are predefined; or it can be closed-loop, where the azimuth and elevation angles are defined at the time of data acquisition, in order to actively compensate for sensor motion. As a result of the narrow laser pulse, the range of scanning LiDAR tends to exceed that of flash LiDAR. Additionally, the field of view can be defined by the scan pattern, and thus designed to exceed that of flash LiDAR. The drawback to scanning LiDAR is that if either the sensor and/or the scene are moving, this motion needs to be corrected for, either actively by controlling the scan pattern while sensing, or retroactively when processing the data. Since motion compensation requires an accurate state estimate, the utility of scanning LiDAR data is dependent on the accuracy of the state estimate.

Hybrid systems, such as those currently in development for the Europa Lander project, use small sensing arrays mounted on scanning platforms to improve upon the limited resolution typical of flash and scanning LiDAR systems, while not requiring excessively long scan durations.

The Lander Vision System Simulator (LVSS)

The simulation environment used for this work is the Lander Vision System Simulation (LVSS). LVSS was developed to support the technology maturation of the Lander Vision System (LVS) for the Mars 2020 program [10]. The inputs to LVSS are the nominal lander trajectory and a terrain map both given with respect to the Planet-Centered-Planet-Fixed (PCPF) frame. Each lander trajectory data entry consists of a time stamp, an XYZ position, and an attitude quaternion. Given the trajectory, LVSS calculates the gravitational acceleration using a spherical gravity model with J_2 perturbation. The lander inertial measurements are provided at 200 Hz. The IMU error model includes: quantization errors, scale factor errors, angle random walk, rate random walk, rate bias, velocity random walk, acceleration random walk, and acceleration bias.

This simulator is capable of generating camera images of the surface terrain using a DEM and a co-registered 8-bit surface reflectance map (SRM). Figure 1a and Figure 1b show an example of the DEM and the SRM respectively. There is also an altimeter model in LVSS, which provides line-of-sight range measurement. The altimeter model served as a starting point for the LiDAR model development.

Smoothing-Based Estimation

The algorithm developed herein adopts a smoothing-based approach to estimate spacecraft states and landmark locations during EDL. Smoothing-based estimation attempts to find the *maximum a posteriori* (MAP) estimate of a set of random variables $\boldsymbol{\Theta}$ given a set of noisy measurements $\mathbf{Z}$. The problem of finding the optimal values $\boldsymbol{\Theta}^*$ is stated formally below.

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\, P(\boldsymbol{\Theta} \mid \mathbf{Z}) = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\, P(\mathbf{Z} \mid \boldsymbol{\Theta})\, P(\boldsymbol{\Theta}) \quad (1)$$

Assuming that each measurement is Gaussian, and using the monotonic natural logarithm function, this problem can be reduced to the problem of minimizing the sum of several probabilistic factors [34], [35].

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}}\; -\log(P(\mathbf{Z} \mid \boldsymbol{\Theta})\, P(\boldsymbol{\Theta})) \quad (2)$$

$$= \underset{\boldsymbol{\Theta}}{\operatorname{argmin}}\; \frac{1}{2} \sum_{f_z \in \mathcal{F}_z} \| h(\boldsymbol{\theta}_{f_z}) \boxminus \mathbf{z}_{f_z} \|^2_{\boldsymbol{\Sigma}_{f_z}} + \frac{1}{2} \sum_{f_p \in \mathcal{F}_p} \left\| \boldsymbol{\theta}_{f_p} \boxminus \tilde{\boldsymbol{\theta}}_{f_p} \right\|^2_{\boldsymbol{\Sigma}_{f_p}} \quad (3)$$

$$= \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} \sum_{f_z \in \mathcal{F}_z} f_z(\boldsymbol{\theta}_{f_z}, \mathbf{z}_{f_z}) + \sum_{f_p \in \mathcal{F}_p} f_p(\boldsymbol{\theta}_{f_p}, \tilde{\boldsymbol{\theta}}_{f_p}) \quad (4)$$

Above, the Mahalanobis distance $\|\boldsymbol{\epsilon}\|^2_{\boldsymbol{\Sigma}}$ is the inverse covariance weighted norm from the multivariate Gaussian distribution; it is defined as follows.

$$\|\boldsymbol{\epsilon}\|^2_{\boldsymbol{\Sigma}} = \boldsymbol{\epsilon}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\epsilon} = \|\boldsymbol{\Sigma}^{-1/2} \boldsymbol{\epsilon}\|^2 \quad (5)$$

Each factor f_z in Equation 4 corresponds to a single measurement $\mathbf{z}_{f_z} \in \mathbf{Z}$, and each factor f_p corresponds to a single prior expected value $\tilde{\boldsymbol{\theta}}_{f_p} \in \tilde{\boldsymbol{\Theta}}$. Whereas $\mathbf{Z}$ and $\tilde{\boldsymbol{\Theta}}$ are considered constant in the optimization, optimal values of $\boldsymbol{\theta}_{f_z} \in \boldsymbol{\Theta}$ and $\boldsymbol{\theta}_{f_p} \in \boldsymbol{\Theta}$ are minimized. The factors $\mathcal{F}$, together with the variables $\boldsymbol{\Theta}$ and edges connecting them $\mathcal{E}$, form a bipartite graph $\mathcal{G}$ called a factor graph. A factor graph encodes the conditional dependency structure of an estimation problem.

The measurement model $h(\cdot)$ in Equation 3 is not necessarily a linear function. Linearization about the current estimate is thus performed in the same way as in the extended Kalman filter. Further, a $\boxminus$ operator is used in Equation 3 instead of the expected $-$ operator because measurements and priors are not necessarily in a vector space. Often, the measurements live on a manifold, such as SE(3). The $\boxminus$ operator is used herein to denote that subtraction takes place in the tangent space of the appropriate manifold. As shown in Equation 5, premultiplication of an error term ϵ by the inverse square root of measurement or prior covariance (denoted "whitening") can be used to turn the Mahalanobis distance into the L2 norm. Thus the problem becomes a sparse nonlinear least squares problem, for which there are a wealth of efficient methods to solve. The incremental (as opposed to batch) solution of the problem above is referred to as incremental smoothing and mapping (iSAM) [35]. The most advanced open source software library implementation of iSAM is called Georgia Tech Smoothing and Mapping (GTSAM) [18]; this library is used herein to incrementally perform smoothing-based estimation.

4. APPROACH

In order to develop and demonstrate an algorithm for LiDAR-inertial navigation and mapping for precision landing, it was necessary to generate the appropriate LiDAR-inertial datasets, design and implement the estimation algorithm, and test it on the generated datasets. The two former topics are covered in this section; the latter is covered in Section 5. Specific attention is given to the novel aspects of our work outlined in Section 2.

LiDAR-Inertial Dataset Generation

This section details the generation of LiDAR-inertial datasets, from creation of the ground truth DEMs, through augmentation of existing EDL simulation software, to data formatting and storage.

High-Resolution Terrain Generation— The simulation of LiDAR range and intensity measurements requires co-registered ground truth topographic and albedo maps at high resolution. Our target map resolution and size were 0.5 m/px and 4×4 km respectively. We also sought a variety of terrains from destinations throughout the solar system. Topographic data at our target resolution is not widely available for locations other than Earth, and to a lesser extent the Moon and Mars. Thus, we leveraged a technique for artificial high-frequency terrain generation presented in [24], [25], extrapolating the low-frequency roughness properties to our target resolution. High-resolution albedo maps are generally unavailable, with the exception of our Nevada data, where LiDAR intensity returns are available at 0.5 m/px. For all other locations we adhered to the following procedure for creation of albedo maps: if co-registered visual images were available, they were used; if they were not, we input a uniform albedo across the map. No attempt was made to upsample the visual images.

Publicly available data from the Planetary Data System (PDS), OpenTopography, University of Arizona, Massachusetts Institute of Technology, and data from private communications with Paul Schenk (LPI), Yang Chen (JPL), and Nickolaos Mastrodemos (JPL) was used to generate 81 topographic and albedo maps. The input resolutions, and thus the amount of terrain content that required artificial generation varied widely between datasets. The Nevada and

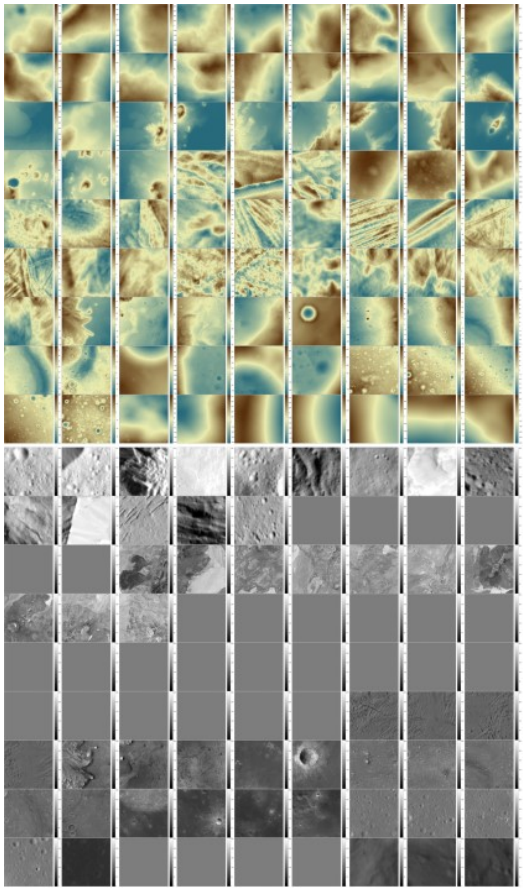


Figure 2. Mosaic of all 4×4 km, 0.5 m/px elevation (top) and albedo (bottom) maps. In order (left-to-right, top-to-bottom) the maps are from Ceres (15), Earth (16), Enceladus (3), Eros (3), Europa (19), Mars (10), the Moon (13), and Vesta (3).

Antarctica datasets, generated using airborne LiDAR, are the most reliable, since they were sampled at 0.5 m/px, and thus required no artificial terrain generation. The data from the Moon and Mars are nearly as reliable, since their ground sample distances were $\leq$ 5 m/px and thus required minimal augmentation. Typical resolutions for the remainder of the data were between 20–80 m/px, meaning that significant addition of high-frequency content was required to create the 0.5 m/px maps. Mosaics of all 81 elevation and albedo maps are shown in Figure 2 (top) and 2 (bottom) respectively.

Lander Vision System Simulator (LVSS) Augmentation— Building on the existing capabilities of the LVSS described in Section 3, we added generic flash and scanning LiDAR models that provide range and scaled intensity measurements. The LiDAR-specific parameters include number of detector pixels, detector field of view, number of anti-aliasing sub-pixels, scan duration, and scan pattern. The true range from the LiDAR sensor to the terrain surface is obtained by performing ray interception with a DEM. The LiDAR intensity measurement is obtained by using the Lambertian scattering assumption, the line-of-sight range, the surface reflectance, and the incident angle of the laser beam. These measurements can be stitched together to form range and intensity images needed for MRL and LiDAR odometry. A simple star tracker model is also added to LVSS that provides attitude measurements. The star tracker-specific parameters

include noise-equivalent-angle (NEA), sensor misalignment error and sample rate.

The detector pixel-array arrangement is formed by discretizing the specified detector field of view in terms of azimuth and elevation angles. The range and the intensity measurements for each pixel is obtained by ray-tracing and ray-interception techniques. Each ray starts from the origin of the LiDAR sensor frame and expands radially outward. The model also supports sub-pixel sampling, which computes the range measurement of a single detector pixel by averaging multiple sub-pixel ranges.

The LiDAR model also provides scaled intensity measurements. They are calculated using the formula from [36], which states the intensity I of a laser return adheres to the following relationship.

$$I \propto \frac{\rho \cos(\alpha)}{d^2} \tag{6}$$

Above, ρ is the surface reflectance from SRM, d is the range from the LiDAR to the surface, and α is the angle between the laser beam direction $\mathbf{n}_b$ and the surface normal $\mathbf{n}_s$ (i.e., $\alpha = \cos^{-1}(\mathbf{n}_b \cdot \mathbf{n}_s)$). The surface normal vector $\mathbf{n}_s$ is obtained by fitting a plane to the nearest 3×3 elevation values centered about the ray-interception point. A collection of raw intensity measurements is grouped together to form the scaled azimuth-elevation intensity image. Each intensity image is normalized and scaled independently to fill the full 8-bit range of values between 0–255.

Algorithm Design and Implementation

The algorithm designed in this work uses a factor graph based smoothing framework to estimate the entire state history (i.e., pose, velocity, accelerometer and gyroscope bias) at the LiDAR scanning frequency; this choice was mainly made to ensure that the entire trajectory is well-estimated, enabling map generation through LiDAR point reprojection.

The open source C++ Georgia Tech Smoothing and Mapping (GTSAM) software library was used as the basic framework for smoothing. In its existing form, it allows for incorporation of several measurement types as probabilistic factors in a factor graph: star tracker orientation, IMU acceleration and angular velocity, MRL position, and flash LiDAR range and bearing to a landmark. However, there were certain areas where GTSAM was lacking. We fixed a bug in the incorporation of Coriolis and centripetal acceleration in a planet-centered planet-fixed frame. We implemented a structureless flash LiDAR range and bearing factor, which eliminates the need to explicitly estimate the position of landmark correspondences and thus decreases the computational cost of the factor. We also designed and implemented a factor that incorporates scanning LiDAR range and bearing to a landmark, and an associated outlier rejection routine. This is made difficult by the fact that landmark detections occur during the scan, and thus between estimated states. To incorporate such landmark detections, local preintegration of the IMU is added to the factor in a novel manner.

A single star tracker fix at the beginning of the trajectory and raw IMU measurements are input into the graph using pose rotation prior and IMU preintegration factors [29], respectively. LiDAR data is binned into azimuth-elevation range and intensity images, where conventional feature matching algorithms are used to perform frame-to-frame tracking of features from the first detection until the feature leaves the frame. A RANSAC-based outlier rejection is used to eliminate outlying correspondences. We denote the addition of these ad-hoc, "opportunistic" landmarks into the factor graph using range and bearing factors as LiDAR odometry. To perform MRL, the state estimate at the time of the LiDAR scan, and, in the case of scanning LiDAR, the local IMU measurements, are used to project the LiDAR data into a point cloud with the same orientation as the map frame. This point cloud is further projected into the map frame's x-y plane as a variable resolution 2D quadtree [37]. Finally, the quadtree is sampled at a suitable resolution for correlation-based matching with a prior map. Finally, the matched position is added to the factor graph as a pose position factor. In the case of scanning LiDAR, accurate velocity knowledge is required to "de-warp" the scan. Thus, for scanning LiDAR, we delay the MRL step by a fixed time so that LiDAR-inertial odometry has been able to determine a velocity estimate.

The topography of the terrain is mapped throughout the duration of EDL. In order to accommodate the inevitable variable resolution of the map, a 2D quadtree data structure [37] is used as the primary map representation. If a prior orbital map is available, the quadtree is initialized with it. LiDAR range and bearing data is projected into the quadtree based on the current best trajectory estimate. To reduce the computational processing associated with mapping, LiDAR points obtained in previous scans are reprojected only when the state change exceeds a prescribed threshold.

Software implementation is performed in C++. In addition to GTSAM, the OpenCV library was leveraged for image processing. A modular architecture was used for easily separable parts of the software library, such as data acquisition and estimation. Unit testing was employed to ensure functionality of each component.

The remainder of this section provides additional details on the algorithm design and implementation, including: coordinate frames and navigation state, smoothing algorithm design, map relative localization, LiDAR odometry, structureless factors, mapping, software architecture, and library implementation.

Coordinate Frames and Navigation State—The frames and navigation state used in this work are visualized in Figure 3.

The planet-centered-inertial coordinate frame is denoted W, for the "world" frame, and does not rotate. The planet's rotation is captured by the planet-centered-rotating coordinate frame P, which is assumed to rotate at a constant rate ${}^P\boldsymbol{\omega}_{P/W} = [0\ 0\ \omega_P]^T$. The origin of W and P are coincident. The map frame M is attached to the planet frame P. It is located close to the surface and is the base coordinate frame both for any existing orbital maps and for maps generated during EDL.

The body frame B is a selected reference frame from the lander's CAD design. The inertial measurement unit (IMU) frame I is attached to B and has axes aligned with the frame in which the IMU senses acceleration and angular velocity. The star tracker frame S is also attached to B, and senses the star tracker attitude in the world frame W. With the assumed availability of ephemeris information, this can be converted to a measurement in the planet frame (i.e., a measurement of ${}^P_S\mathbf{R}$). The LiDAR frame L represents the sensing frame of the LiDAR, and is attached to the body frame B.

Denoting the space of navigation states as $\mathbb{N}$\$, the spacecraft

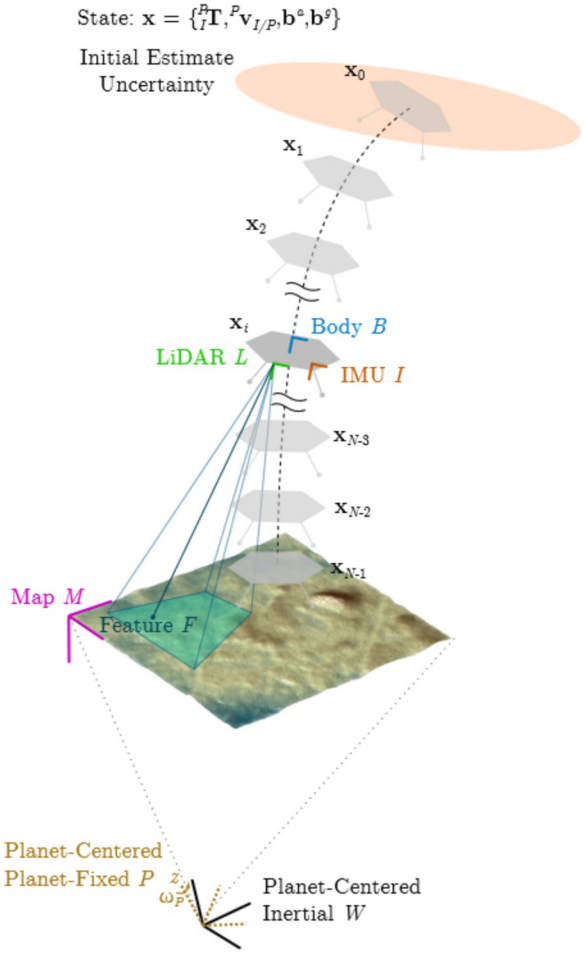


Figure 3. Visualization of frames and states for LiDAR-Inertial navigation and mapping.

navigation state $\mathbf{x}$ to be estimated is described as follows:

$$\mathbf{x} = \left\{ {}^{P}_{I}\mathbf{T}, {}^{P}\mathbf{v}_{I/P}, \mathbf{b}^a, \mathbf{b}^g \right\} \in \mathbb{N}\$, \text{ where} \tag{7}$$

$${}^{P}_{I}\mathbf{T} = \begin{bmatrix} {}^{P}_{I}\mathbf{R} & {}^{P}\mathbf{t}_{I/P} \\ \mathbf{0}_{3\times 1} & 1 \end{bmatrix} \in \mathrm{SE}(3) : \text{ IMU pose}, \tag{8}$$

$${}^{P}\mathbf{v}_{I/P} \in \mathbb{R} : \text{ IMU velocity}, \tag{9}$$

$$\mathbf{b}^a \in \mathbb{R} : \text{ accelerometer bias}, \tag{10}$$

$$\mathbf{b}^g \in \mathbb{R} : \text{ gyroscope bias}. \tag{11}$$

Smoothing Algorithm Design—Smoothing was chosen as the preferred estimation method over filtering for several reasons. Whereas filtering finds the best estimate of the latest state, smoothing finds the best estimate of the entire trajectory. As stated above, the primary reason smoothing was chosen is that we are reprojecting LiDAR data from throughout the trajectory. If we are able to improve upon our estimates of past states, we are then able to better reproject the LiDAR data obtained at those past states. However, smoothing also provides a way to recover absolute position when an initial MRL fix is acquired, but accurate velocity estimates are not available until later. Additionally, it provides a natural way to include measurements delayed either by design or by computational processing time; this in contrast to a filter, which requires that navigation state be either cloned or rewound so that the measurement can be applied at the correct time.

To obtain a smoothed solution, we define the trajectory $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N-1}\}$ as a sequence of states $\mathbf{x}_i$, and a collection of opportunistic landmark positions $\mathbf{L} = \{\mathbf{l}_0, \mathbf{l}_1, \ldots, \mathbf{l}_{M-1}\}$, where $\mathbf{l}_m \in \mathbb{R}$. We attempt to solve the optimization problem posed in Section 3, with variables $\mathbf{\Theta} = \{\mathbf{X}, \mathbf{L}\}$. The set of measurements $\mathbf{Z} = \{\mathbf{Z}^s, \mathbf{Z}^a, \mathbf{Z}^g, \mathbf{Z}^l\}$ contains star tracker measurements $\mathbf{Z}^s$, accelerometer measurements $\mathbf{Z}^a$, gyroscope measurements $\mathbf{Z}^g$, and LiDAR measurements $\mathbf{Z}^l$. With these definitions, we then write out the *maximum a posteriori* optimization problem for the structured optimization problem.

$$\{\mathbf{X}^*, \mathbf{L}^*\} = \underset{\mathbf{X},\mathbf{L}}{\operatorname{argmax}} \left\{ J = P(\mathbf{X}, \mathbf{L} \mid \mathbf{Z}^s, \mathbf{Z}^a, \mathbf{Z}^g, \mathbf{Z}^l) \right\} \tag{12}$$

It is possible to factor the cost function J into the convenient form below.

$$J \propto P(\mathbf{X})P(\mathbf{Z}^s \,|\, \mathbf{X})P(\mathbf{X} \,|\, \mathbf{Z}^a, \mathbf{Z}^g)P(\mathbf{Z}^l \,|\, \mathbf{X}, \mathbf{L}, \mathbf{Z}^a, \mathbf{Z}^g) \tag{13}$$

Note that in the above expressions the prior $P(\mathbf{Z}^l)$ is discarded since it does not depend on either $\mathbf{X}$ or $\mathbf{L}$. The prior $P(\mathbf{L})$ has also been discarded, since we have no prior expectation on the position of the ad-hoc, opportunistic landmarks. Each term in Equation 13 represents a set of factors of the same type. The first term, $P(\mathbf{X})$ represents our prior expectation of the trajectory. We only assume that we only have prior information on the first state, and thus $P(\mathbf{X}) = P(\mathbf{x}_0)$. In our case, star tracker measurements, which apply measurements of orientation, are only applied to the first state, and thus $P(\mathbf{Z}^s|\mathbf{X}) = P(\mathbf{z}^s_0|\mathbf{x}_0)$. The second term is the the IMU preintegration factor from [29]. This factor constrains two adjacent states through the one-time preintegration of IMU measurements. Since the only gravity model available in GTSAM was gravity as a constant vector, we augmented the IMU preintegration with a J_2 gravity model.

The final factor in Equation 13 represents the contribution of LiDAR measurements to the optimization problem, and is where the majority of the development was focused. LiDAR measurements were used both for LiDAR odometry and map relative localization. In the case of flash LiDAR, an entire scan is obtained instantaneously at timestep i when the spacecraft has a pose ${}^{P}_{I_i}\mathbf{T}$, which is directly estimated. In this case the factor does not depend on IMU measurements and can be rewritten as $P(\mathbf{Z}^l|\mathbf{X}, \mathbf{L})$. However, in the case of scanning LiDAR, the scan is obtained at a high rate between timesteps i and j. Since state is only directly estimated at timestep i and timestep j, local IMU measurements must be incorporated, explaining the form in Equation 13.

Factors for both MRL and LiDAR odometry take the form of this final factor. For MRL, LiDAR measurements are used to determine the spacecraft's position and applied directly to the state's pose as a position factor. In the scanning case, the factor's dependence on IMU measurements comes in the form of requiring that the LiDAR scan be "de-warped" using a state estimate and local IMU measurements. For LiDAR odometry, multiple observations of opportunistic landmarks in azimuth-elevation space are applied to the graph as range and bearing measurements. Naturally, these measurements are dependent on the locations of landmarks $\mathbf{L}$ as well as the state $\mathbf{X}$. For the flash LiDAR case, these measurements can be applied using an existing range and bearing factor in GTSAM. For the scanning LiDAR case, these measurements again depend on local IMU measurements. For this we formulate a novel factor, augmenting the GTSAM library,

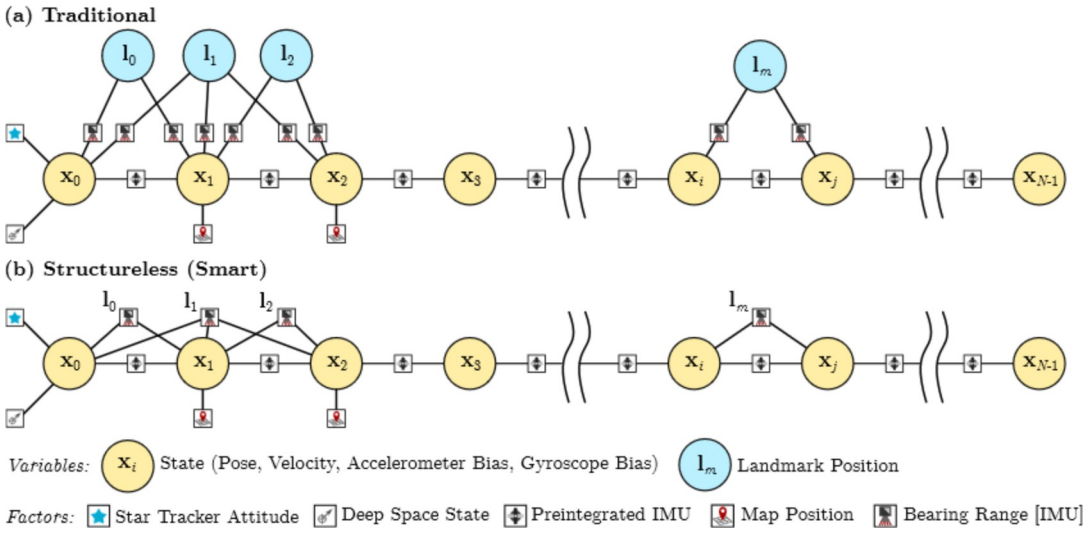


Figure 4. Pictorial representation of the traditional (a), and structureless (b) factor graphs solved in this project.

which can, in a probabilistically correct manner, incorporate the dependence of these measurements on preintegrated IMU measurements.

Figure 4a is a pictorial representation of the factor graph $\mathcal{G}$ for our problem. Note that all states are tied together by preintegrated IMU factors, and thus there is no requirement that a state be attached to any other factor (see, for example, state $\mathbf{x}_3$). Map relative localization is accomplished through the application of map position factors, while LiDAR odometry is accomplished through the repeated observation of landmarks and the application of bearing range factors. Note that in the case of scanning LiDAR the latter factor is called the bearing range IMU factor, to indicate that it also depends on IMU measurements. Figure 4b shows the structureless (aka smart) factor graph, which is discussed further below.

LiDAR Odometry—Herein, LiDAR odometry tracks SURF features [38] (aka opportunistic landmarks) in 8-bit intensity and high-passed range images. Each feature's trail across the image is tracked long as a feature matching the original descriptor lies in the image. Features tracked in both the intensity and range images are converted into a range and bearing measurement via subpixel interpolation of the range image. Inlying features are added to the factor graph as landmarks and contribute to the optimization problem via the final term in Equation 13. Outlying features are rejected using a frame-to-frame RANSAC-based outlier rejection. The outlier rejection methodology varies between flash and scanning LiDAR.

Flash LiDAR Outlier Rejection: For flash LiDAR, outlier rejection is performed in a similar way as in stereo visual odometry [39], using RANSAC with Horn's absolute orientation algorithm [40] to find the largest set of inlying features.

Scanning LiDAR Outlier Rejection: For scanning LiDAR, outlier rejection requires the development of a novel approach. Previous methods for scanning LiDAR [30] and rolling shutter camera [41] outlier rejection assumed constant linear and angular velocity over the duration of data acquisition. This assumption precludes tolerance of long scans and large potential accelerations present in our application. We instead formulate a method based on local IMU preintegration and estimation of the initial velocity. While IMU mechanization was leveraged in [33] and [42], these works required that either state be estimated at the IMU rate or parametrized using b-splines; our method requires neither.

Flash LiDAR Range and Bearing Factor: For flash LiDAR, the observation of an opportunistic landmark provides a single range and bearing measurement, which is directly associated with a LiDAR pose ${}^{P}_{L_i}\mathbf{T}$. This factor then simply requires that the range and bearing to the estimated landmark position $\mathbf{l}_m$ be compared with the measurement. The range is compared in $\mathbb{R}$, and the bearing is compared in the tangent space of the unit sphere S^2. The traditional version of this factor, which requires that landmark position $\mathbf{l}_m$ be explicitly estimated in the factor graph, was already implemented in GTSAM.

Scanning LiDAR Range, Bearing, and IMU Factor: For scanning LiDAR, the observation of an opportunistic landmark again provides a single range and bearing measurement; however, since the measurement occurs between estimated states $\mathbf{x}_i$ and $\mathbf{x}_j$, local preintegration of the IMU is necessary to predict the pose ${}^{P}_{L_p}\mathbf{T}$ of the LiDAR at the pulse time t_p. For this purpose, GTSAM's IMU preintegration factor was adapted to cache the entire history of IMU measurements, local navigation state, associated state covariances, and bias Jacobians. With this variation, the IMU pose at pulse time, its associated uncertainty, and its dependence on estimated pose, velocity, and IMU bias can be calculated. With the above modifications to the measurement model, the same error function as in the flash LiDAR case can be applied.

Structureless Factors—In traditional SLAM, both the states $\mathbf{X}$ and landmark locations $\mathbf{L}$ are estimated. However, herein we are interested only in the states $\mathbf{X}$ and a dense topographic map from reprojected LiDAR measurements. Sparse feature positions $\mathbf{L}$ are estimated only to improve estimates of state $\mathbf{X}$ via LiDAR odometry. Thus it is possible to denote the states $\mathbf{X}$ as our *target* variables and the landmarks $\mathbf{L}$ as our *support* variables, and leverage the concept of "smart" (aka "structureless") factors first presented in [29], [28].

In our application, this means that the landmarks $\mathbf{L}$ can be eliminated from the factor graph (i.e., $\mathbf{\Theta} = \mathbf{X}$ in the structureless case), reducing the dimensionality of the problem, and increasing the computational performance. Instead, all observations of a single landmark are converted into a single factor which constrains the associated states. Mathematically, this is performed by exploiting the nullspace trick pioneered as part of the MSCKF [12], and re-introduced in the context of smoothing in [28]. Although not estimated probabilistically, the location of the landmarks are required in the factors. For this, a simple average of all projected feature locations is performed. The structureless factor graph is depicted in Figure 4b. In this work we only apply the structureless technique to the flash LiDAR case.

Map Relative Localization— Map Relative Localization (MRL) is utilized when a prior elevation map of the landing site is available. MRL allows position priors to be included in the optimization problem by matching incoming LiDAR data during landing to the prior map. The prior map is loaded into memory when the program is started. As new LiDAR measurements become available, each scan is projected into the map frame using the associated state estimate. For flash LiDAR measurements all points arrive simultaneously, and matching to the prior map is performed immediately. With scanning LiDAR, points making up a scan arrive individually throughout the scan duration. To account for this, preintegrated IMU measurements are used to de-warp the LiDAR scan into a point cloud in the map frame. Additionally, MRL is delayed when using scanning LiDAR data to ensure the availability of accurate velocity estimates when the scan is de-warped.

Once the measurements have been projected into the map frame as a point cloud, they are inserted into a new quadtree. Optionally, the 2D convex hull in the that contains all points is computed and only points that lie in the largest inscribed square of the convex hull are inserted into the quadtree; this ensures no large regions of the quadtree are have a relatively low resolution. This quadtree is used to generate a 32-bit floating point scan map image at a resolution between 0.5 m/px and 32 m/px. These images are then be used to match against the prior map. Additionally, the difference between the center position of the projected LiDAR image and the state estimate is computed.

Normalized Cross Correlation (NCC) is used to find the region in the prior map that has the highest similarity to the LiDAR data at pixel level accuracy. After obtaining an initial match, quadratic interpolation is performed on the correlation surface values in the 5×5 px region surrounding the match to find a sub-pixel match at the maxima. Finally, the 3D lander position is computed by applying the offset computed earlier from the LiDAR measurements to the position matched in the prior map. The uncertainty in this measurement is computed in the horizontal direction by first linearizing the NCC and quadratic interpolation calculations around the individual elevation measurements in the 5×5 px match regions of both images; this linearization is then used to transform the known uncertainty of the LiDAR and prior map measurements to a single horizontal uncertainty in the map frame [43]. The vertical uncertainty is computed as the standard deviation of the difference between the matched region of the prior map elevation values and the LiDAR elevation values. Both the position and uncertainty are then transformed from the map frame to the PCPF frame and applied as a position factor in the optimization problem.

Before they are applied as a position factor in the optimization problem additional criteria must be met to ensure that no incorrect matches are accepted. First, the scan's footprint on the terrain is checked; scans with footprints less than 5×5 map pixels are deemed to small for accurate matching. Second, the correlation value of the match is checked to be at least a specified threshold and that no secondary peaks exist above another specified threshold. This helps avoid accepting matches where no good match exists or where self-similar terrain is present. Next, the root mean square (RMS) of the 3σ horizontal covariance ellipse principal axes is computed. If this RMS value is greater than three map pixels the match is rejected. If the norm is below this threshold, a final check is performed to ensure the elevation standard deviation is less than two times the sum of the prior map and LiDAR measurement standard deviations.

Mapping—Mapping is performed by projecting LiDAR measurements into a quadtree that spans a specified area in the map reference frame. If a prior (orbital) map exists, it is loaded into the quadtree when the program starts such that the quadtree depth is uniform across the area down to the prior map's ground sampling distance. The elevation values of the prior map serve as the initial elevation value at the quadtree depth corresponding to the ground sampling distance of the map, while higher levels are the mean elevation of all leaf nodes below.

When new LiDAR measurements are received, these points are projected based on their associated state estimate into 3D points in the map frame and inserted into the quadtree. Additionally, each point in the quadtree has an identifier that indicates the original scan and state that it is associated with. As the points travel to leaf nodes in the quadtree, their elevation values are used to adjust the mean elevation estimate at each level.

At a specified interval, the map is updated by cycling through each of the prior state estimates and checking if they have changed by a specified threshold since the last time they were used to project their associated LiDAR measurements. If any of the parts of the state exceed their threshold, their associated points are removed, reprojected into the map frame according to the new state estimate, and then re-inserted into the quadtree. In this way the map is continually being refined and improving in accuracy.

5. Results & Discussion

This section presents the methodology used to test our algorithm and the resulting performance. We begin by discussing the specifics of the generated LiDAR-inertial datasets. We test LiDAR-inertial odometry, LiDAR-inertial odometry with MRL, and finally, LiDAR-inertial odometry with MRL and mapping. We leave the task of mapping completely unknown terrain to future work; this would be a relatively straightforward addition to our algorithm.

LiDAR-Inertial Datasets

Our datasets, simulated in LVSS, are designed to show that our algorithm works with realistic instruments with minimal or no impact on mission con-ops.

Instrument Specifications—The simulated LiDAR specifications are given in Table 1. The flash LiDAR is based on the space-qualified 128×128 px ASC Goldeneye LiDAR that was flown on Osiris-REx and used in the NASA ALHAT

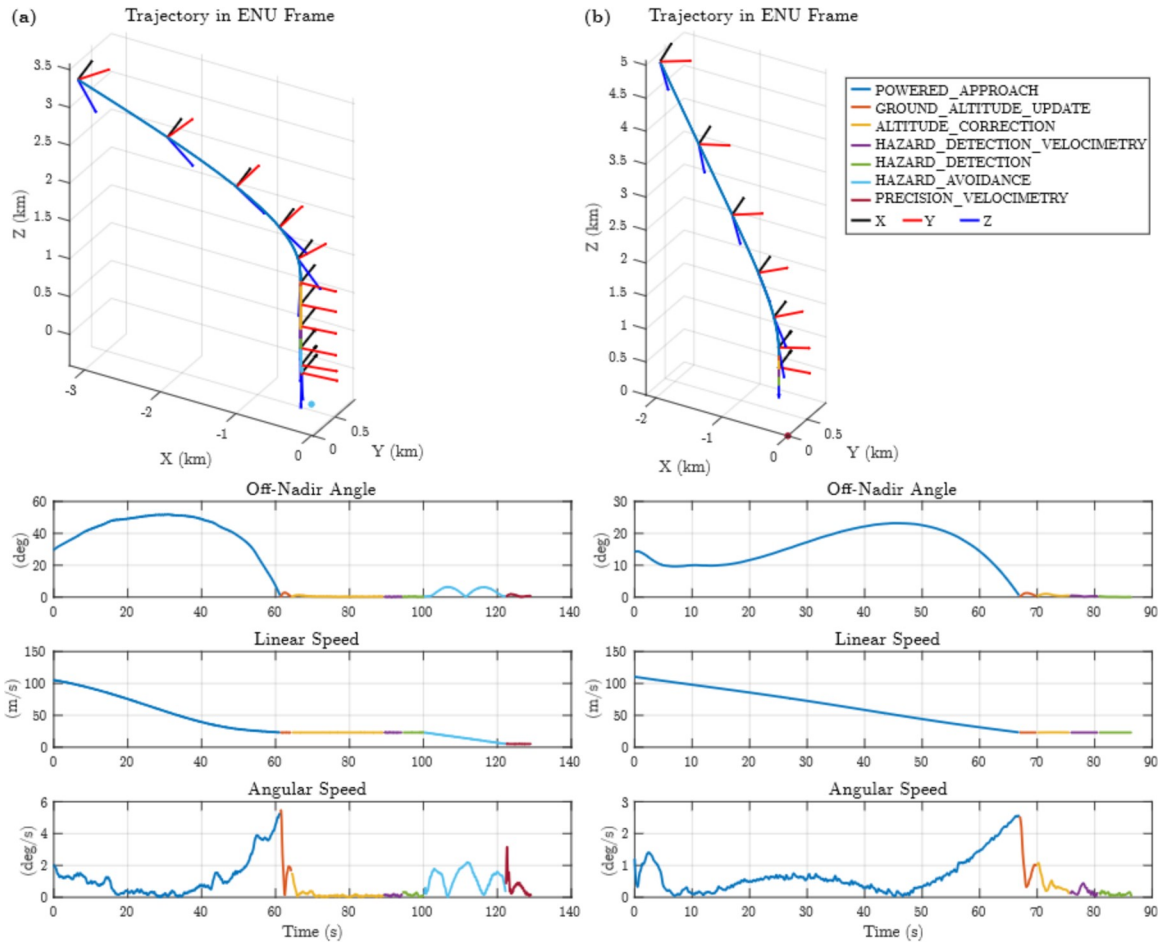


Figure 5. Summary of Trajectory-1 (a) and Trajectory-4 (b).

project. Our chosen field of view of 6° falls within the range of previous applications, but we have assumed that it can operate at this field of view up to 5 km, which would require a more powerful laser than used previously. The frequency and range noise are well within the capabilities of the sensor.

Table 1. Flash and scanning LiDAR specifications.

Type	Res.	Field of View	Scan Freq.	Noise 1σ	Max. Range	Scan Pattern
Flash	128 px	6°	2 Hz	0.25 m	5 km	flash
Scanning	128 px	11.42°	0.5 Hz	0.25 m	5 km	bi-raster

The scanning LiDAR is based on those in development for the Europa Lander project, operating in open-loop (i.e., without motion compensation) in the coarse hazard mapping mode mode. Our specifications match or are inferior to the planned LiDARs. Since the scan patterns in development are proprietary, we have opted for a standard open-loop bidirectional raster scan pattern, which scans one row in a left-to-right direction, the next in a right-to-left direction, and so on.

The simulated IMU is based on the Honeywell MIMU used for the Mars 2020 rover, except with bias and noise specifications which have been artificially degraded for export control reasons. The star tracker sensor is based on the Jena Optronik ASTRO APS, running at 8 Hz, with cross-boresight noise of 1 arcsecond 1σ and a boresight noise of 10 arcseconds 1σ.

Trajectories—To test the algorithm, we employ two different trajectories, named Trajectory-1 and Trajectory-4. Both are provided by Gurkirpal Singh at JPL from Monte-Carlo simulations of the Europa Lander mission. Except for the small portion of the trajectory in which the spacecraft is in hazard detection mode, these trajectories are in no way tailored to terrain relative navigation.

Trajectory-1, plotted in Figure 5a, is low and fast, starting at approximately 3.3 km altitude, and reaching linear and angular velocities in excess of 100 m/s and 5°/s respectively. It is used to test LiDAR-inertial odometry. Trajectory-4, plotted in Figure 5b, starts at approximately 5 km altitude and descends steeply. This steep descent allows us to fit the LiDAR footprint within our 4×4 km DEM. It is used to test LiDAR-inertial odometry with MRL.

LiDAR-Inertial Odometry

LiDAR-inertial odometry is most important at low altitudes, where the scan footprints are smaller, and thus MRL is less viable. LiDAR odometry is most challenging when linear and angular velocities are high, since high velocities reduce the overlap between subsequent scans and increase the scan distortion (in the scanning LiDAR case). Trajectory-1 then represents a relevant and challenging case. The test datasets

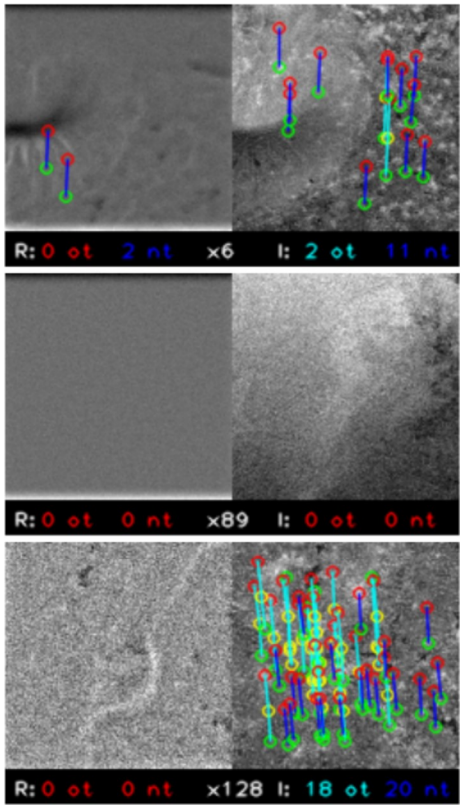


Figure 6. Example flash LiDAR feature range (left) and intensity (right) images and associated feature trails. Feature detections are marked as circles (green for first detection, red for last detection, and yellow for in between). New feature trails (nt), with only two feature detections are connected by blue lines. Old feature trails (ot), with more than two feature detections are connected by cyan lines.

are simulated as if the spacecraft was landing on a terrain in the Lunar Crater Volcanic Field regeion in Nevada (denoted Nevada-1 herein).

Flash LiDAR—Figure 6 shows example feature trails from the Nevada-1, Trajectory-1 flash LiDAR dataset. Range images are high-passed by subtracting a Gaussian blurred copy of themselves, and discretized to 8-bit so that SURF features can be detected in OpenCV. Intensity images are also 8-bit, and normalized to span the entire range from 0–255. Feature trails only contain correspondences that have passed outlier rejection. While there are many feature trails at state $\mathbf{x}_6$ (Figure 6 top) and state $\mathbf{x}_{128}$ (bottom), there are none in the relatively benign terrain at state $\mathbf{x}_{89}$ (center). It is clear from these images that when both range and intensity images are discretized to 8-bit, intensity images typically provide richer data for feature correspondence.

Figure 7 shows a comparison between inertial odometry and flash LiDAR-inertial odometry. Both are given perfect pose priors but incorrect zero IMU bias priors. Inertial odometry is given a perfect initial velocity prior, whereas LiDAR-inertial odometry is given a loose (100 m/s 1σ) zero velocity prior. The thick dot-dashed lines show the instantaneous state estimate, the dashed lines show the 3σ marginal uncertainty of the instantaneous state estimate, and the semi-transparent solid lines show the smoothed trajectory estimate at each time step. The instantaneous state estimate is the best estimate

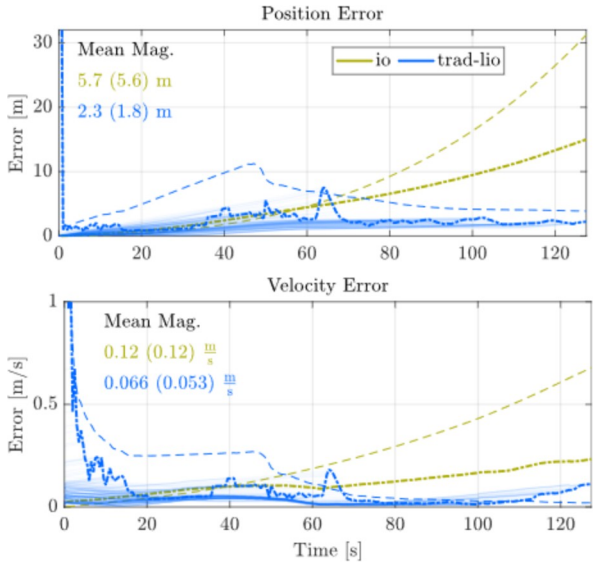


Figure 7. Comparison of inertial odometry (io) with a perfect velocity prior with flash LiDAR-inertial odometry (trad-lio) with a zero velocity prior.

of state at the time. Since the entire trajectory is estimated at each timestep, obtaining the instantaneous state estimate simply involves extracting the estimated value of the most recent state. The mean error magnitude is noted for the instantaneous state estimate (following 4 s of convergence), and for the entire smoothed trajectory (in parentheses).

It can be seen that after approximately 70 s, LiDAR-inertial odometry outperforms inertial odometry, even in the unrealistic case that inertial odometry is given a perfect velocity prior. This demonstrates that LiDAR-inertial odometry can be used effectively for velocimetry with flash LiDAR. It is noted that at some points in the trajectory, particularly past 110 s in the velocity plot, the 3σ marginal uncertainty provided by GTSAM is overconfident in the estimate. While the reason for this is not fully understood, it is likely an artifact of the simulation; at very low altitudes the 0.5 m/px ground truth terrain resolution is insufficient for rendering the scan, the range and intensity images are highly pixellated, and measured feature positions are less accurate. Since measurement range and bearing covariances are set to a constant value at all altitudes, this likely leads to the observed overconfidence in state estimate at low altitudes.

Structureless Smoothing: The primary goal of structureless smoothing is to remove the estimation of opportunistic landmark positions from the LiDAR-inertial odometry problem, and thus reduce the computational cost of the algorithm. Figure 8 compares the traditional and structureless (smart) approaches to LiDAR-inertial odometry. The position and velocity errors are virtually indistinguishable between approaches, but the cumulative computation time devoted to inference in the structureless case is more than a factor of four less than in the traditional case. This computational reduction results from the drastic reduction in the number of factors and number of variables in the structureless case, shown at the bottom of Figure 8. The traditional approach estimates three variables per scan (pose, velocity, and IMU bias) plus one variable per opportunistic feature, of which there are thousands; by contrast, the structureless approach estimates only three variables per scan. Also evident in

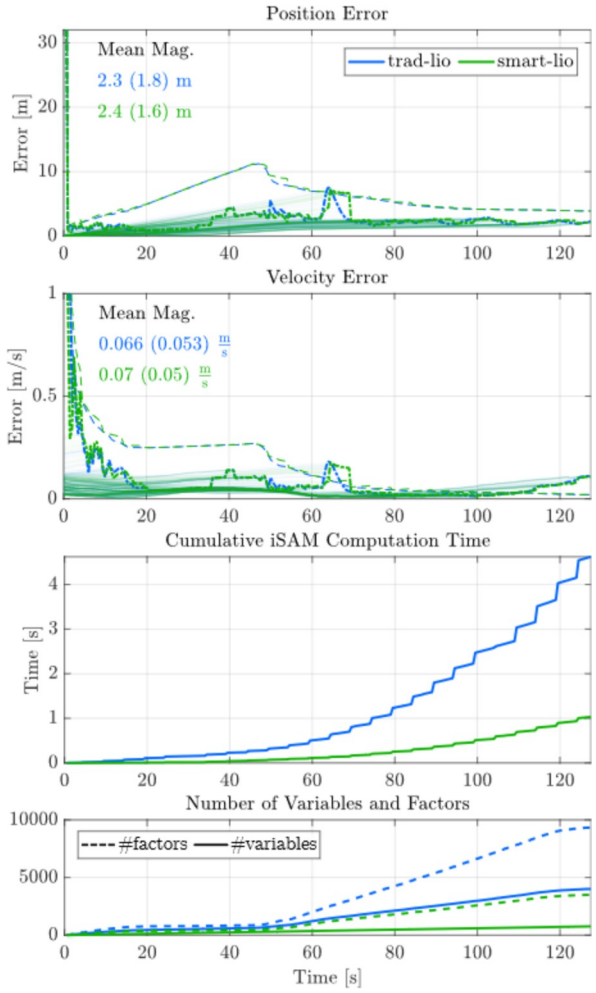


Figure 8. Comparison of traditional (trad-lio) and structureless (smart-lio) LiDAR-inertial odometry with a zero velocity prior.

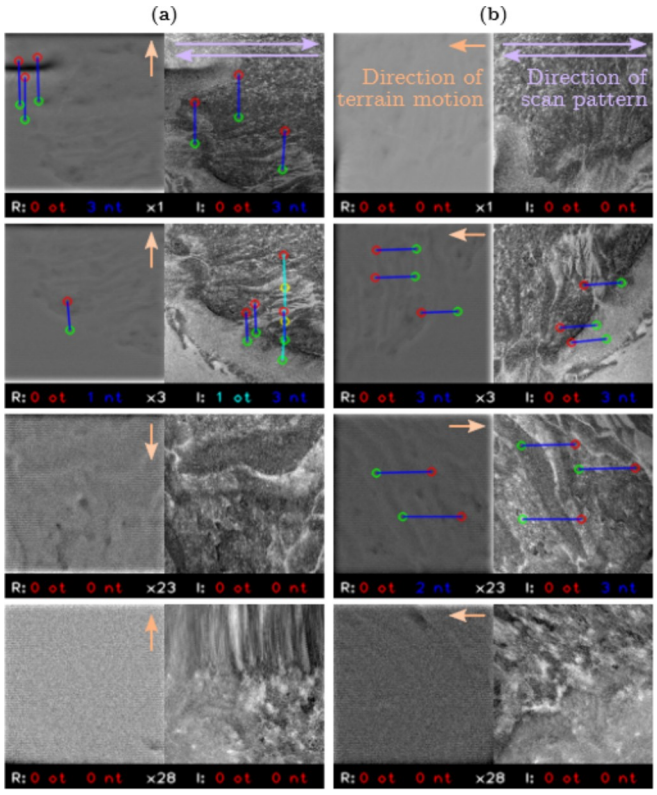


Figure 9. Example LiDAR feature range (left) and intensity (right) images and associated feature trails for scanning LiDAR with a cross-track scan pattern (a), and an along-track scan pattern (b). Feature trail coloring and annotation is the same as in Figure 6. Direction of terrain motion in the LiDAR frame is shown by the light orange arrows. Scan pattern direction is shown by the light purple arrows.

Figure 8 is that there is a significant increase in the inclusion of landmarks after 50 s; this is terrain specific, with the scanned terrain being relatively benign from 0–50 s for the Nevada-1, Trajectory-1 dataset.

Scanning LiDAR—Despite the fact that our scanning LiDAR operates four times slower than our flash LiDAR, there is still significant overlap between subsequent images. There are two reasons for this: the field of view of the scanning LiDAR is almost double that of the flash LiDAR; and the continuous nature of data collection causes scan distortion that can actually aid in image overlap.

Figure 9 shows the distortion in the azimuth-elevation binned range and intensity images resulting from using scanning LiDAR on a moving spacecraft. When rows of the scan are perpendicular to the sensor's velocity (i.e., in the cross-track direction), motion will tend to compress or stretch the images (see Figure 9a). When rows of the scan are parallel with the sensor's velocity (i.e., in the along-track direction), motion will tend to skew the images (see Figure 9b). The type and severity of distortion affects the abundance and accuracy of frame-to-frame feature correspondences. Interestingly, all permutations of successful feature tracking exist: cross-track only (first row of Figure 9), both (second row of Figure 9), along-track only (third row of Figure 9), and neither (final row of Figure 9). We thus investigate both cross-track and along-track sensor orientations to ensure that LiDAR-inertial odometry functions in the face of both compression and skew distortions.

Figure 10 shows a comparison between scanning LiDAR-inertial odometry with the sensor oriented in the cross and along-track directions. The along-track experiment was performed by rotating the LiDAR frame L by 90° along its boresight in the spacecraft body frame B. The performance between the two is relatively similar, indicating that neither compression or skew distortion of the range and intensity images precludes accurate LiDAR-inertial odometry, even during aggressive powered approach. Despite having access to a quarter of the number of scans, the final smoothed trajectory error is similar to that of flash LiDAR odometry shown in Figure 7. However, the instantaneous state estimate has difficulty fully converging in the initial 40 s, over which the terrain is benign and there are few feature correspondences.

LiDAR-Inertial Odometry with MRL

Building on LiDAR-inertial odometry demonstrated in the previous section, we add MRL for absolute position determination, completing the navigation portion of our algorithm. We use the same priors as for LiDAR-inertial odometry, except for position, for which we apply a prior of 4 km

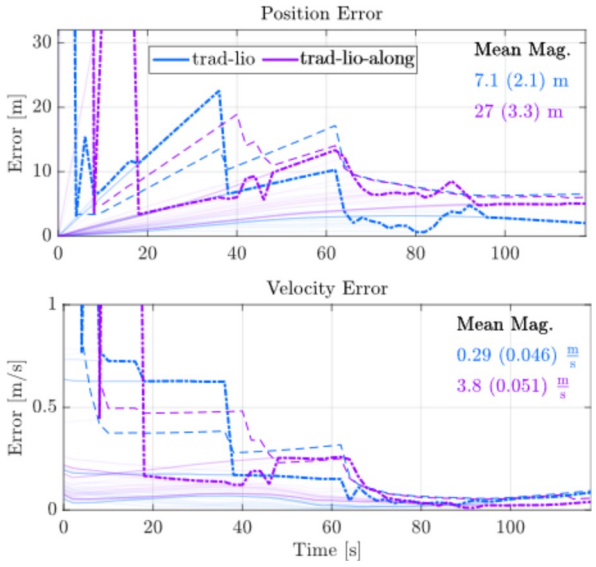


Figure 10. Comparison of traditional scanning LiDAR-inertial odometry with a zero velocity prior for cross (trad-lio) and along-track (trad-lio-along) scanning directions.

altitude above the center of the map.

When using flash LiDAR, MRL position estimates are calculated and applied immediately following availability of the LiDAR data; when using scanning LiDAR we delay the calculation and application of MRL position estimates by six scans (i.e., upon receipt of the seventh scan, the first scan is processed for MRL, and, if it passes outlier rejection, included in the factor graph). This was found to be a conservative delay in MRL, which allowed the estimated velocity to converge, and facilitated accurate de-warping of the scanning LiDAR scan. Note that to account for this lag, when calculating the mean magnitude of the current state estimate errors, we exclude the states with time stamps prior to 14 s (i.e., after seven scans at 0.5 Hz). For consistency we do this for both flash and scanning LiDAR data evaluation.

Algorithm performance is highlighted by executing a simulated descent over the e86-1 terrain on Europa. We then test the algorithm over all terrains, with different prior map fidelities, to see how it performs for different destinations across the solar system with different data quality. All tests are done using Trajectory-4, which allows the estimation process to start at the limit of LiDAR range, and gives MRL the greatest chance at making successful detections.

Europa Example—Example visualizations of attempted MRL matches over the example Europan e86-1 terrain with a 4 m/px prior map are shown in Figure 11. Each example displays the scan map in the top-left corner and the matched prior map area in the bottom-left corner. The central image displays the entire prior map, accepted (green crosshairs) position estimates, and rejected (red crosshairs) position estimates. The crosshairs for the current position estimate are larger than those for previous ones. The right image shows the thresholded correlation map, which is purple for all values under the minimum correlation. The bottom informational bar shows the matching resolution (i.e., 4 m/px), the dimension of the matched template (i.e., between 16×16 px and 64×64 px in the examples), the maximum correlation value

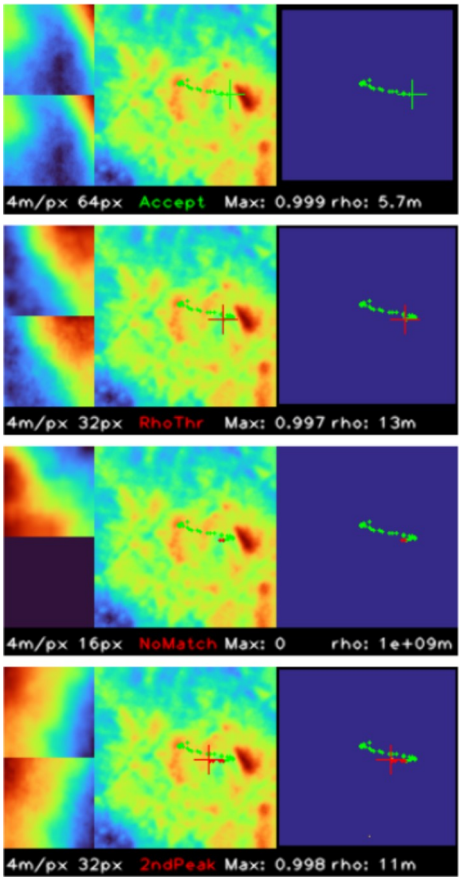


Figure 11. Examples of attempted MRL matches.

(when it surpasses the minimum correlation threshold), and `rho`, the RMS value of the 3σ horizontal uncertainty ellipse principal axes.

The top example shows an accepted match, with high correlation, low `rho`, and no secondary peaks. The second example shows a match which was rejected because its `rho` value was in excess of its threshold of three map pixels (i.e., $\rho = 13 \text{ m} > 3 \text{ px} \cdot 4 \text{ m/px}$). The third example shows a case in which no match above the minimum correlation threshold was found. Finally, the bottom example shows a case in which an otherwise valid match was rejected because of the presence of a second peak above the allowed threshold. The second peak is small, but visible as an orange dot in the purple thresholded correlation map.

Results for flash LiDAR-inertial odometry with MRL are shown in Figure 12. Results from two different map resolutions are shown: 4 m/px, the highest resolution tested, and 32 m/px, the lowest resolution tested. Independent white Gaussian noise was added to each prior map: with a standard deviation of 0.5 m for the 4 m/px map, and with a standard deviation of 4 m for the 32 m/px map. For Europa, the low-resolution map is representative of the products that will be available for potential lander sites after the Europa Clipper mission. The high-resolution map is representative of higher quality products, as available on Earth, the Moon, and Mars.

When using the 4 m/px map, inlying MRL position estimates, denoted by solid circles in Figure 12, begin immediately, and are obtained late into the run. When using the 32 m/px map,

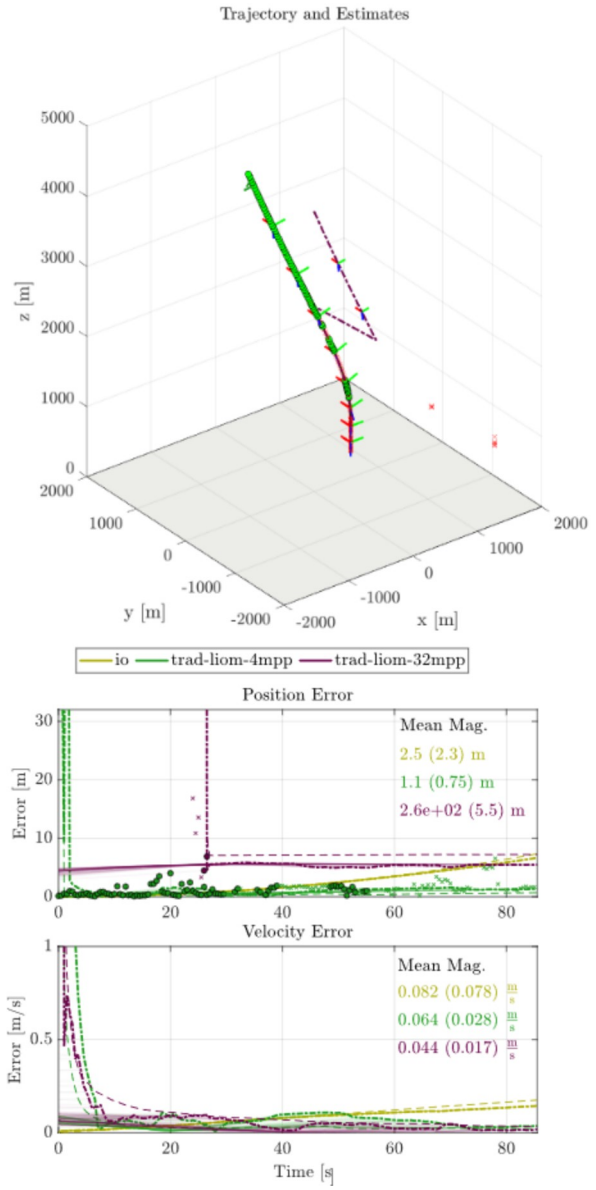


Figure 12. Comparison of inertial odometry and flash LiDAR-inertial odometry with MRL, for both a 4 m/px, 0.5 m 1σ error map, and a 32 m/px, 4 m 1σ error map. Accepted MRL position estimates are shown with solid circles; rejected ones are shown with $\times$s.

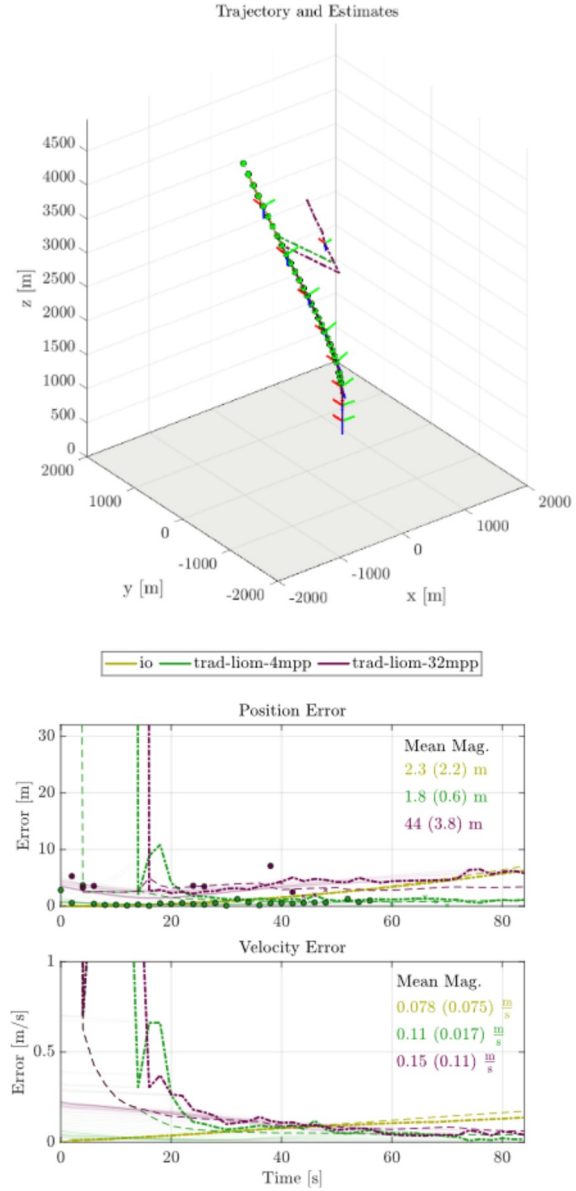


Figure 13. Comparison of inertial odometry and scanning LiDAR-inertial odometry with MRL, for both a 4 m/px, 0.5 m 1σ error map, and a 32 m/px, 4 m 1σ error map. Accepted MRL position estimates are shown with solid circles; rejected ones are shown with $\times$s.

only two inlying MRL position estimates are obtained, and not until after 20 s into the run. Accurate template matching for MRL is typically harder with less information [27]. Thus, the fewer pixels and higher noise of the low-resolution map make the task of MRL more difficult, and lead to lower position accuracies. The result is that the position estimate begins from the prior position with only LiDAR-inertial odometry, and snaps to the correct position upon receipt of the first MRL position estimate (see 3D plot in Figure 12).

Despite being influenced by MRL position factors, velocity estimation is mainly dependent on the observation of opportunistic landmarks in LiDAR-inertial odometry. Thus it remains relatively similar between the two methods.

Results for scanning LiDAR-inertial odometry with MRL, again for two different map resolutions, are shown in Figure 13. Here, and for all subsequent runs, the scanning direction was cross-track. Despite having a data frequency four times less than flash, performance is comparable. An inlying MRL position estimate is found at an early state for both map resolutions. That an early estimate is obtained in the 32 m/px map, where it was not with flash LiDAR, can likely be attributed to the scanning LiDAR's wider field of view. A wider field of view allows more map pixels, and thus more information, to be included. It is evident from these results that the scan is being adequately de-warped, even at velocities up to 100 m/s, to facilitate accurate MRL.

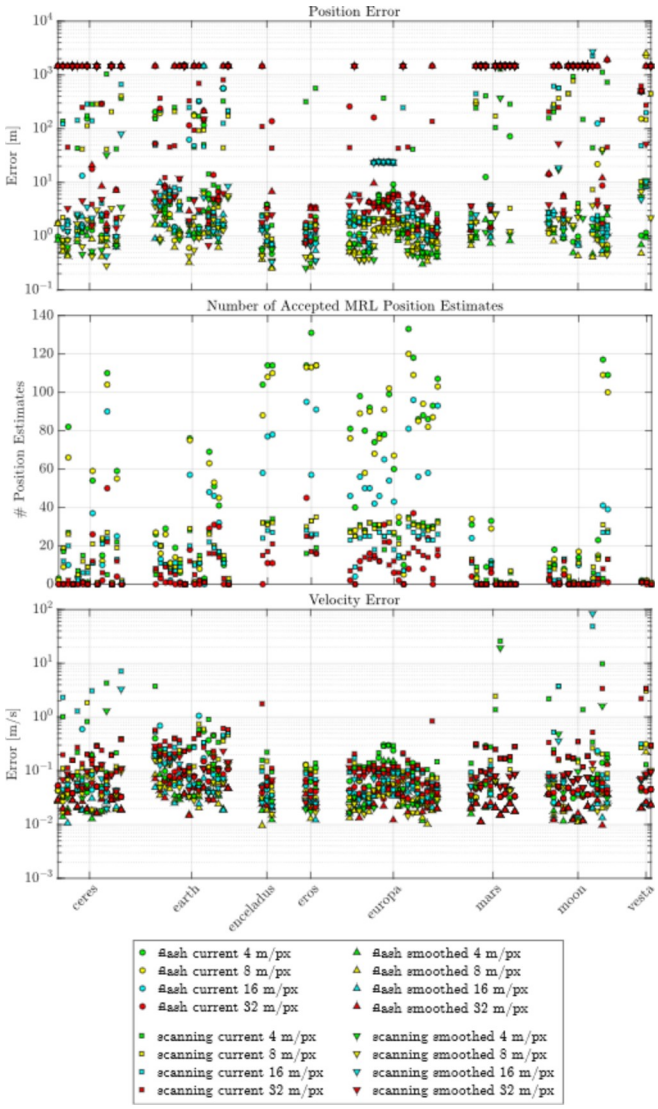


Figure 14. Summary of results from LiDAR-inertial odometry with MRL for all planetary terrains and different prior map fidelities.

Performance Across All Planetary Bodies—Identical tests to those performed for the Europa example were performed for all 81 planetary terrains. For each planetary terrain, prior maps with resolutions $\{4,\ 8,\ 16,\ 32\}$ m/px and corresponding map noise standard deviations of $\{0.5,\ 1,\ 2,\ 4\}$ m were used as the base maps for MRL.

A summary of results is shown in Figure 14. The mean magnitude of current and smoothed estimates, which correspond to the values displayed on error plots above (smoothed estimate errors in parentheses), are displayed. Additionally, the number of MRL position estimates is shown for each run.

Unsurprisingly, position accuracy, map resolution, and number of accepted MRL position estimates are all positively correlated. Position errors of around 1.5 km are indicative of cases where no MRL position estimates were obtained. Failure to obtain a singe MRL position estimate is unlikely for resolutions better than 32 m/px. Planetary bodies with highly textured terrains, such as Enceladus, Eros, and Europa, perform better at low resolution than those with more bland terrain such as Earth, Mars and the Moon. When selecting terrains for simulation, we chose some sites that had already been landed at, such as the Apollo 11 landing site on the Moon, and others on which landings have been proposed, such as Oxia Planum on Mars (for ExoMars). Interestingly, these terrains were likely selected specifically because they are relatively safe and benign, and our algorithm did not perform well on them. This suggests that our algorithm is most accurate in the exact scenarios in which it is designed to operate: over rough, hazardous terrain.

Again, velocity accuracy remains relatively uniform across different prior map resolutions, since MRL position estimates play a much smaller role in velocimetry than does LiDAR-inertial odometry. Velocity accuracy also remains relatively uniform across different terrains, including terrains such as Europa, for which the albedo is set to a uniform value in our simulations. Thus, variation in terrain normal angles provide enough texture in the intensity image to accurately perform LiDAR-inertial odometry. Velocity estimation errors above 1 m/s are rare and typically only occur when using scanning LiDAR. Typical errors are less than 10 cm/s.

A tiny minority of cases result in catastrophic failure of the smoothing algorithm. Typically, these failures appear to be driven by the coupling between orientation and the rest of state, which we plan to eliminate in future work. In very rare instances, incorrect MRL estimates make it through outlier rejection and are the cause of errors above 1.5 km in position.

LiDAR-Inertial Odometry with MRL and Mapping

In this section, mapping is demonstrated for the same scenario as the Europa example above. The program loads the 4×4 km resolution, 0.5 m/px resolution, noiseless ground truth map and resamples it to 32 m/px. It then applies additive zero mean Gaussian noise with 4 m standard deviation. This 32 m/px topographic map is used to initialize both an immutable MRL (aka prior) map (for localization) and a refined (aka estimated) map (which will be updated with LiDAR measurements). When a new LiDAR scan is received, its points are projected into the map frame using its associated state estimate. To evaluate mapping performance, we compare the resultant estimated map with the ground truth map at a specified resolution and time interval, in this case at 4 m/px every 10 s. At the same 10 s interval, the algorithm corrects previous LiDAR scan projections if, since the previous update, any part of their associated state estimates have changed by the corresponding threshold in Table 2.

Table 2. State estimate change thresholds for mapping updates.

Attitude	Position	Velocity	Gyro. Bias	Accel. Bias
0.001 rad	4 m	1 m/s	0.00001 rad/s	0.01 m/s^2

Qualitatively, the improvement in the final estimated map over the prior map in terms of resolution can be seen in Figure 15. To quantify this, at the 10 s mapping interval the root mean squared error (RMSE) of the refined map compared to the ground truth map is computed at a specified ground sampling distance (4 m/px was selected herein). To calculate this error, we generate a map from the quadtree of the refined map at 4 m/px resolution, interpolate our ground truth map to the same resolution, and compute absolute differences for all pixels. An example of the absolute error in the refined map

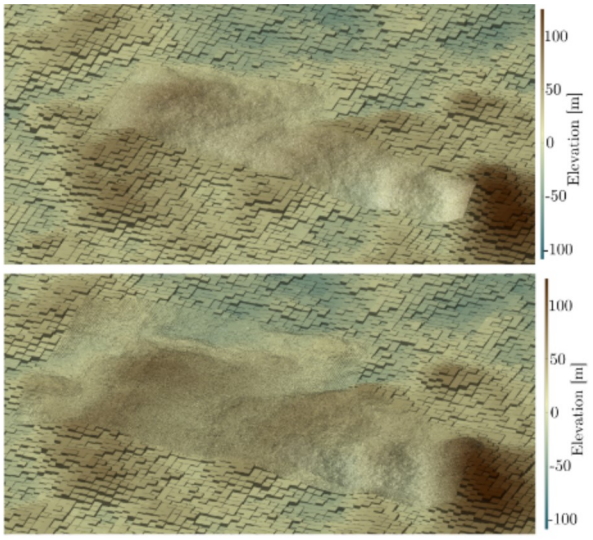


Figure 15. Example of final estimated map at 0.5 m/px for flash (top) and scanning (bottom) LiDAR measurements. The surrounding low resolution information is from the 32 m/px prior orbital map.

throughout the run is shown in Figure 16. These images show how the map error is reduced in regions the LiDAR observes and improves over time as state estimates improve.

6. Conclusions & Future Work

This report outlines the development and demonstration of a novel LiDAR-inertial based navigation and mapping algorithm for precision landing. Several LiDAR inertial datasets have been generated throughout the solar system by upsampling existing topographic maps and running simulations using an augmented version of LVSS. A smoothing framework has been designed and implemented to estimate the entire trajectory of a landing spacecraft. Topographic map refinement has been achieved through the reprojection of LiDAR scans.

LiDAR-inertial odometry results are promising, indicating that velocimetry performs reliably on a variety of terrains with both flash and scanning LiDAR. While our work relies on measurement of the laser return intensity, we suggest that if only range data is available, normalized cross-correlation of range images in azimuth-elevation space may provide a viable alternative. A structureless approach for flash LiDAR odometry was demonstrated to reduce the algorithm's computational expense by a factor of four.

Map relative localization provided accurate spacecraft position estimates in cases where the combination of terrain roughness and map fidelity was satisfactory. For rough terrain, such as that on Enceladus, Europa, and Eros, maps with 32 m/px resolution and 4 m elevation noise standard deviation were typically sufficient for localization. Elsewhere, maps with 16 m/px resolution and 2 m noise standard deviation or 8 m/px resolution and 1 m noise standard deviation were adequate. For standalone use of this landing system on a mission, in-depth study of the terrain roughness and map information content would be required to ensure reliable MRL.

Map refinement through reprojection of LiDAR points into the map frame was demonstrated to improve map resolution and accuracy. These results are a preliminary step in acquiring lander-scale resolution hazard detection and avoidance maps. In general, mapping at lander-scale resolutions will require lander-scale localization accuracy, which is achieved in some but not all of our experiments. The interconnectedness of localization and mapping means that improvements in LiDAR-inertial odometry and MRL will facilitate improvements in mapping accuracy.

Several such improvements will be the subject of future work on this landing system. Most notably, for orientation estimation, it is difficult to outperform estimates provided by an initial star tracker measurement and the integration of an EDL-class gyroscope, even when gyroscope bias is incorrectly assumed to be zero. As a result, we will likely decouple orientation estimation from the rest of the smoothing problem in the future.

Another potential improvement to the current system is the checking of frame-to-frame feature correspondences for separation within the azimuth-elevation range and intensity images. Large separation of tracked features promotes accurate LiDAR odometry. In the future, we could include a check for separation in the outlier rejection step, preferring sets of feature correspondences whose convex hulls subtend a larger solid angle.

There is additional work to be done on the subject of terrain mapping. Firstly, the state estimate change thresholds can be further tuned to properly balance computational expense and map accuracy. Secondly, in the absence of an MRL match to the prior orbital map, matching with respect to the refined map could be attempted. This would facilitate the inclusion of MRL position estimates in the case when there is no prior orbital map or in the case where the spacecraft has descended to an altitude that makes matching to the low-resolution prior orbital map difficult or impossible. Matching to the refined map is likely to improve mapping by ensuring that the scan projections are properly aligned before being incorporated into the quadtree.

Following these improvements, the larger tasks for future work are the adaptation of the software for real-time operation on real-world data, which will be pursued at JPL in the coming year.

Acknowledgments

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). © 2019 California Institute of Technology. Government sponsorship acknowledged. The authors would like to thank Yang Cheng, Nickolaos Mastrodemos, Paul Schenk, Marissa Cameron, and Cynthia Philips for providing the imagery and DEMs, Andrew Johnson for his insights into the Mars 2020 LVS system, and David Sternberg and Christian Liebe for reviewing this manuscript. The information presented about the Europa Lander and other potential future missions is pre-decisional and is provided for planning and discussion purposes only.

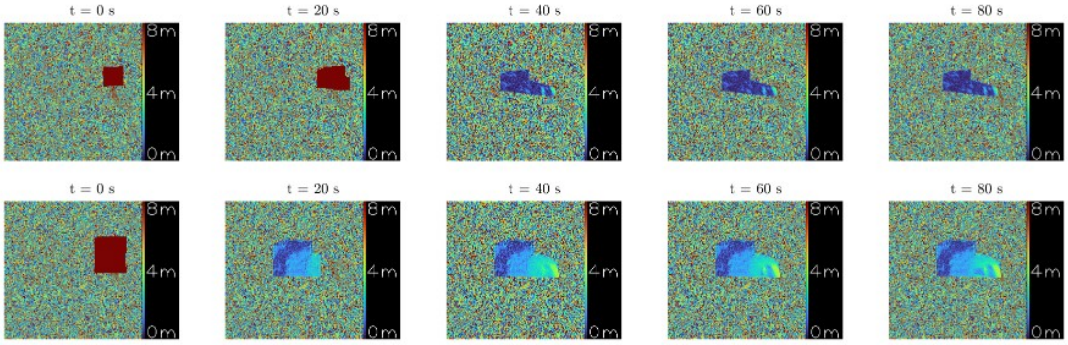


Figure 16. Example showing the progression of estimated absolute map error for flash LiDAR (top row) and scanning LiDAR (bottom row) over the duration of the trajectory. Note that error visualization is capped at two times the prior map's error standard deviation.

REFERENCES

[1] P. T. Chen, S. Kaki, A. M. San Martin, D. Skulsky, A. Katake, and N. Trawny, "Lidar-Generated Digital Elevation Models for Hazard Detection - Requirements and Analysis," in *AAS Guidance Navigation and Control Conference*, 2019, pp. 19–032.

[2] ASC, "Goldeneye 3d flash lidar space camera," 2015, http://www.advancedscientificconcepts.com/products/portable.html.

[3] I. Poberezhskiy, A. Johnson, D. Chang, E. Ek, D. Natzic, G. Spiers, S. Penniman, and B. Short, "Flash lidar performance testing: configuration and results," *Laser Radar Technology and Applications XVII*, vol. 8379, no. May 2012, p. 837905, 2012.

[4] A. Board of Regents, "Instruments - OSIRIS REx Mission," https://www.asteroidmission.org/objectives/instruments/.

[5] J. A. Keim, S. Mobasser, E. S. Bailey, A. E. Johnson, and G. Khanoyan, "Test implementation to evaluate technologies for safe lunar landing," *IEEE Aerospace Conference Proceedings*, 2011.

[6] F. Amzajerdian, L. B. Petway, G. D. Hines, V. E. Roback, R. A. Reisse, and D. F. Pierrottet, "Lidar sensors for Autonomous Landing and Hazard Avoidance," *AIAA SPACE 2013 Conference and Exposition*, 2013.

[7] M. G. Daly, O. S. Barnouin, C. Dickinson, J. Seabrook, C. L. Johnson, G. Cunningham, T. Haltigin, D. Gaudreau, C. Brunet, I. Aslam, A. Taylor, E. B. Bierhaus, W. Boynton, M. Nolan, and D. S. Lauretta, "The OSIRIS-REx Laser Altimeter (OLA) Investigation and Instrument," *Space Science Reviews*, vol. 212, no. 1-2, pp. 899–924, 2017.

[8] C. Schmitt, S. Dochow, M. Windmuller, J. Both, and O. Mongrard, "RVS 3000-3D Lidar Gateway Rendezvous and Lunar Landing," in *AAS GN&C Conference*, 2020.

[9] N. Trawny, A. I. Mourikis, S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery, "Vision-Aided Inertial Navigation for Pin-Point Landing using Observations of Mapped Landmarks," *Journal of Field Robotics*, vol. 24, no. 5, pp. 357–378, 2007.

[10] A. Johnson, S. Aaron, J. Chang, Y. Cheng, J. Montgomery, S. Mohan, S. Schroeder, B. Tweddle, N. Trawny, and J. Zheng, "The Lander Vision System for Mars 2020 Entry Descent and Landing," in *AAS Guidance Navigation and Control Conference*, Breckenridge, CO, 2017, pp. AAS–17–038.

[11] A. Johnson, N. Villaume, C. Umsted, A. Kourchians, D. Sternberg, N. Trawny, Y. Cheng, E. Geipel, and J. Montgomery, "The Mars 2020 Lander Vision System Field Test," in *AAS Annual Guidance and Control Conference*, 2020, pp. AAS 20–105.

[12] A. I. Mourikis and S. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," *Robotics and Automation, 2007*, pp. 1–20, 2007.

[13] D. Adams, T. B. Criss, and U. J. Shankar, "Passive optical Terrain Relative Navigation using APLNav," *IEEE Aerospace Conference Proceedings*, 2008.

[14] T. G. McGee, U. Shankar, S. Shapiro, W. J. Shyong, C. J. Krupiarz, D. Reid, and J. T. Kaidy, "Performance analysis of the APLNav system for passive optical lunar navigation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2009.

[15] T. G. McGee, T. B. Criss, P. Rosendall, A. Hill, W. J. Shyong, N. Mehta, C. Reed, G. Chavers, M. R. Hannan, and C. Epp, "APLNav: Development status of an onboard passive optical terrain relative navigation system," *AIAA Guidance, Navigation, and Control Conference*, 2015.

[16] M. Fritz, A. Olguin, K. Smith, R. Lovelace, R. R. Sostaric, S. Pedrotty, J. Estes, T. Tse, and R. Garcia, "Operational Constraint Analysis of Terrain Relative Navigation for Landing Applications," in *AIAA SciTech Forum*, 2020.

[17] T. J. Steiner, "Utility-based Map Reduction for Ground and Flight Vehicle Navigation," Ph.D. dissertation, Massachusetts Institute of Technology, 2015.

[18] F. Dellaert and C. Beall, "GTSAM 4.0," 2020, https://github.com/borglab/gtsam.

[19] J. M. Carson, E. Robertson, N. Trawny, and F. Amzajerdian, *Flight Testing ALHAT Precision Landing Technologies Integrated Onboard the Morpheus Rocket Vehicle*. American Institute of Aeronautics and Astronautics, 2015. [Online]. Available: http://dx.doi.org/10.2514/6.2015-4417

[20] A. E. Johnson and J. F. Montgomery, "Overview of Terrain Relative Navigation approaches for precise lunar landing," *IEEE Aerospace Conference Proceedings*, 2008.

[21] A. Johnson and T. Ivanov, "Analysis and Testing of a LIDAR-Based Approach to Terrain Relative Navigation for Precise Lunar Landing," in *AIAA Guidance, Navigation, and Control Conference*, Portland, 2011.

[22] J. P. Lewis, "Fast normalized cross-correlation," 1995.

[23] J. De Lafontaine, D. Neveu, and J. F. Hamel, "Pseudo-Doppler velocity navigation for Lidar-based planetary exploration," *Collection of Technical Papers - AIAA/AAS Astrodynamics Specialist Conference, 2006*, vol. 3, no. August, pp. 1627–1641, 2006.

[24] T. P. Setterfield, R. A. Hewitt, and N. Trawny, "Map Relative Localization for Europa Lander," Jet Propulsion Laboratory, California Institute of Technology, Tech. Rep., 2020.

[25] R. A. Hewitt, T. P. Setterfield, and N. Trawny, "LiDAR-Based Map Relative Localization Performance Analysis for Landing on Europa," in *IEEE Aerospace Conference*, 2021, p. Submitted.

[26] J. Balaram, R. Austin, P. Banerjee, T. Bentley, D. Heriquez, B. Martin, E. McMahon, and G. Sohl, "DSENDS - A High-Fidelity Dynamics and Spacecraft Simulator for Entry, Descent and Surface Landing," in *IEEE Aerospace Conference*, 2002.

[27] T. P. Setterfield, R. A. Hewitt, and N. Trawny, "Fundamental Limits of Map Relative Localization during Spacecraft Entry , Descent , and Landing," *AIAA Journal of Guidance, Control, and Dynamics*, vol. Submitted, 2020.

[28] L. Carlone, Z. Kira, C. Beall, V. Indelman, and F. Dellaert, "Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4290–4297, 2014.

[29] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Forster, Christian, et al. "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, 2016.

[30] S. W. Anderson, "Batch Continuous-Time Trajectory Estimation," Ph.D. dissertation, University of Toronto, 2017.

[31] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, 2014.

[32] ——, "Laser-visual-inertial odometry and mapping with high robustness and low drift," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1242–1264, 2018.

[33] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1104–1119, 2012.

[34] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[35] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[36] B. Höfle and N. Pfeifer, "Correction of laser scanning intensity data: Data and model-driven approaches," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, no. 6, pp. 415–433, 2007.

[37] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in *Proc. of the ICRA ...*, 2010. [Online]. Available: http://ais.informatik.uni-freiburg.de/publications/papers/wurm10octomap.pdf

[38] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Computer Vision-ECCV*, 2006.

[39] T. P. Setterfield, "On-Orbit Inspection of a Rotating Object Using a Moving Observer," PhD Thesis, Massachusetts Institute of Technology, 2017.

[40] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of Optical Society of America*, vol. 4, no. 4, pp. 629–642, 1987.

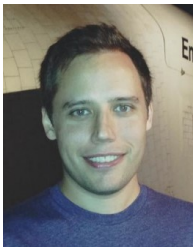
[41] M. Li, B. H. Kim, and A. I. Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4712–4719, 2013.

[42] R. Zlot and M. Bosse, "Efficient Large-scale Three-dimensional Mobile Mapping for Underground Mines," *Journal of Field Robotics*, vol. 31, no. 5, pp. 758–779, 2014.

[43] H. Zhang, D. Grießbach, J. Wohlfeil, and A. Börner, "Uncertainty Model for Template Feature Matching," in *Pacific-Rim Symposium on Image and Video Technology*. Springer, 2017, pp. 406–420.

BIOGRAPHY

Timothy P. Setterfield *is a Guidance and Control Engineer at the Jet Propulsion Laboratory in Pasadena, CA. He holds a BScE in Mechanical Engineering from Queen's University, Kingston, ON, Canada, a MASc in Mechanical and Aerospace Engineering from Carleton University, Ottawa, ON, Canada, and a PhD in Aeronautics and Astronautics from the Massachusetts Institute of Technology. He has previously worked at Nanometrics Seismological Instruments developing a miniature broadband seismometer, at Carleton University developing a planetary micro-rover prototype, at the European Space Agency coordinating graduate student research in micro and hypergravity, and at MIT researching vision-based navigation using the SPHERES-VERTIGO platform.*



Robert Hewitt *received his Ph.D. in Electrical and Computer Engineering in 2018 from Queens University (Kingston, Canada), after a M.A.Sc.in Aerospace Engineering from Carleton University (Ottawa, Canada) and a B.A.Sc in Engineering Physics from the University of Saskatchewan (Saskatoon, Canada). He is currently a robotics technologist at the Jet Propulsion Laboratory. He develops autonomous navigation systems for aerial and ground vehicles used in prospective planetary missions. His interests include state estimation, perception, and instrumentation.*

Po-Ting Chen *received his B.S. degree (2009) and M.S. degree (2010) from the University of California, San Diego, and a Ph.D. degree (2018) from the University of California, Los Angeles all in aerospace engineering. He is currently an engineer with the Guidance and Control Section at NASA Jet Propulsion Laboratory with expertise in stochastic processes, estimation, and dynamics simulation.*

Antonio Terán Espinoza *is a CONACyT fellow and doctoral candidate at the MIT Space Systems Laboratory and Marine Robotics Group. He received his B.S. from the National Autonomous University of Mexico (UNAM), and his S.M. degree from MIT as a Fulbright student. He is currently an integral member of the SPHERES team, and is working towards his Ph.D. in AeroAstro with a focus in autonomous navigation, mapping, and perception.*

Nikolas Trawny *received a Diploma of Aerospace Engineering from the University of Stuttgart, Germany, in 2004, and his M.Sc. and Ph.D. degrees in Computer Science and Engineering from the University of Minnesota in 2008 and 2010, respectively. Since then, he has been working at the NASA Jet Propulsion Laboratory on projects involving estimation and navigation for safe and precise spacecraft landing, including ALHAT, ADAPT, Mars 2020 (Lander Vision System), and the Europa Lander preproject.*

Anup Katake *is a member of the Technical Staff in the Guidance and Control Section at Jet Propulsion Lab in Pasadena, California. He has more than 14 years of expertise in developing high-fidelity electro-optical sensor systems such as star sensors, lidar technologies and vision-based sensor systems for precision autonomous navigation of spacecraft. At JPL, Dr. Katake is the GN&C heritage sensors lead for the Mars 2020 mission. He is also the lead for the development of a next generation lidar for hazard detection for Europa. He has worked on various aspects of lidar design hardware and algorithms for navigation. Prior to joining JPL, he was a Team Leader for Imaging Systems at Physical Optics Corporation where he designed and developed sensors for GPS denied navigation, day/night star sensors for azimuth estimation and omnidirectional high-resolution surveillance. While at StarVision Technologies Inc., he led the development of a firstofakind 100 Hz stellar gyroscope and autonomous Landing Sensor for Mars/Moon for the NASA ALHAT program. He has a B. Tech. in Electrical Engineering from the Indian Institute of Technology in Bombay, and a M.S. and Ph.D. in Aerospace Engineering from Texas A&M University, in 2002 and 2006 respectively.*