Landing Site Mapping and Selection with a Quadtree Map using Planar Elements

Corey L. Marcus * and Renato Zanetti†
The University of Texas at Austin, Austin, TX, 78712

Timothy P. Setterfield‡
NASA Jet Propulsion Laboratory, Pasadena, CA, 91109

This work proposes a filter-based combined terrain mapping and landing site selection system for planetary landing. We consider a spacecraft acquiring flash lidar measurements of the surface during terminal descent from which it must select a landing site. A quadtree is utilized to create a variable resolution map of the terrain in which each element contains a planar estimate. This greatly simplifies the landing site selection system by providing a direct estimate of terrain slope. Our system is validated with flight test data gathered by a rocket-powered vehicle approaching an artificial Lunar environment.

I. Introduction

Many scientifically interesting destinations for interplanetary landers feature poorly characterized hazards. These include the rocks and craters of the Moon or the crevasses of Europa. These hazards could be identified *a priori* and avoided using orbital imagery as shown by the Mars2020 mission [1], but this process is time consuming and expensive. Light-time delays make remote control almost impossible. Future robotic missions should have the capability to autonomously map terrain and select landing sites during descent.

Significant research is ongoing to select landing sites based on maps generated or refined during descent. NASA's Autonomous Landing and Hazard Avoidance Technology (ALHAT) and Safe and Precise Landing (SPLICE) programs [2, 3] contain broad forays into the subject and demonstrated many of their algorithms with test flights. Traditional map representations often utilize a fixed-resolution Digital Elevation Map (DEM) [4, 5]. The DEM is a 2D discretization of horizontal space into a set of rectangular elements. Each element contains an elevation estimate. Extraterrestrial terrain often consists of hazardous rocks and craters separated by relatively large areas of flat terrain in which a lander could safely touch down. The sensors commonly used to sense hazards, cameras and lidar, naturally provide higher resolution information of the ground at lower altitudes. The combination of these two facts motivates the use of a variable resolution map representation. Terrain scanned from high altitude or containing sparse features can be represented

*Graduate Student, Aerospace Engineering and Engineering Mechanics; cmarcus@utexas.edu
†Associate Professor of Aerospace Engineering and Engineering Mechanics and AIAA Associate Fellow; renato@utexas.edu
‡Robotics Systems Engineer

at a low resolution, limiting computational complexity and memory requirements. Terrain scanned at a low alitude or containing complex features can be represented at a high resolution, preserving map fidelity. The quadtree [6] is a variable-resolution data structure which can be adapted to represent a variable resolution DEM.

Almost all landing site selection (LSS) systems assume slope and roughness constraints for landing. Most systems utilize DEMs modeling terrain within each cell as a single elevation. This forces the LSS system to derive a slope estimate from the map. Typically, LSS systems provide this estimate by fitting a plane to the local terrain [7–12]. To simplify the LSS process, we propose that the mapping system should estimate slope directly. Roughness quantification remains the responsibility of the LSS system.

In this work we present a system which jointly maps terrain and selects landing sites. By designing the two sub-systems in tandem, a more efficient design has been produced. The mapping sub-system utilizes a quadtree map representation which allows efficient yet accurate representation of terrain. Instead of each quadtree containing a single elevation estimate, it contains a planar estimate of the terrain. These estimates can be updated recursively as new measurements are acquired. The estimated normal vector of each plane reduces LSS complexity by encapsulating all required slope information. Surface normals are estimated with the technique developed by Hertzberg [13] which avoids singularities and redundant parameters. Our system is validated with flight test data gathered by a rocket-powered vertical take off and landing (VTOL) vehicle flying over an artificial Lunar terrain.

This work is a continuation of two others; the first of which concerns terrain mapping [14]. The previous mapping work demonstrated a similar quadtree mapping system in which each cell contained a planar estimate. Here, we present three key improvements upon that system. Where the previous system updated its planar estimates with an Extended Kalman Filter, this one presents fuses estimates in a manner found to have more consistent convergence behavior. The previous system only contained planar estimates for the lowest levels of the tree, whereas this work shows that lower level estimates can be recursively fused to provide estimates at any level of the tree. Finally, the previous system was only tested in simulation, this work demonstrates system performance with flight data.

The second continued work concerns LSS [12]. The previous LSS work demonstrated a system optimized to work in tandem with a quadtree-based navigation and mapping system. The old mapping system did not directly estimate terrain slope. The previous LSS system was required to estimate slope through a triangulation system applied to the estimated elevations. Because the current mapping system estimates slope, this complex triangulation system can be eliminated, streamlining the LSS process. In addition, we introduce a new technique for estimating terrain features which are too small to for accurate representation by the smallest quadtree cells. This allows more accurate terrain representation without unnecessary division while enabling the LSS system to avoid these small scale features.

II. Related Work

The aforementioned quadtree navigation and mapping system was presented in detail by Setterfield *et al* [15]. This system differs from ours in that Setterfield *et al* use a single elevation estimate for each cell. Furthermore, these elevation estimates are found as an average of all measurements binned to the cell, as opposed to this work which process measurements in a recursive filter. Finally, their quadtree is divided as a function of measurement density instead of a quality of fit metric.

Terrain mapping has been extensively researched, particularly in the field of Simultaneous Localization and Mapping (SLAM). Various authors have explored the utilization of planar elements in their map construction. For instance, Viejo and Cazorla [16] derived planar elements from point cloud data and applied the Iterative Closest Point algorithm [17] to align the planes and achieve a SLAM solution. However, their approach did not involve organizing with quadtree or updating the planes using a recursive filter. Several other authors [18, 19] presented SLAM systems that incorporated planar elements updated through the Extended Kalman Filter (EKF). It is worth noting that these systems utilized a spherical coordinate system to parameterize the normal vectors of the planes, whereas we adopted a parameterization developed by Hertzberg [13]. Furthermore, similar to Viejo and Cazorla, these systems did not employ a structured data representation for organizing the planes. Another recent SLAM system based on planes was demonstrated by Grant [20], which focused on effectively utilizing lidar data and also accommodated individual points when planes were not suitable. Rather than solving the SLAM problem with an EKF, they formulated it as a factor-graph optimization and employed iSAM2 for solving [21].

Another mapping application that has leveraged our unit vector representation [13] is the work conducted by Gallant and Marshall [22]. In their study, they adopted the same representation to map the orientation of local planar surfaces and find sensor heading estimates. However, their focus did not involve estimating the positions of the surfaces or organizing them into a structured representation.

LSS, along with terrain mapping, has been a subject of extensive research. In a previous study by Johnson et al., a LSS system was described which utilized a scanning lidar [23]. The authors employed a plane fitting method to estimate surface slope and roughness, working directly with the lidar-generated point cloud and assuming prior knowledge of the vehicle's position. Following site selection, a guidance module was utilized to generate a trajectory towards the chosen location. The effectiveness of their algorithm was verified through simulation experiments.

Another study conducted by Serrano explores the topic of autonomous LSS [7]. In this research, a probabilistic approach is employed, utilizing a Bayesian Network that incorporates measurements from radar, camera, and lidar sensors. The characterization of terrain slope and roughness involves fitting planes to small subsets of the map. Craters are detected in camera images through edge and shadow detection techniques and matched with a predefined map. The presence of rocks is modeled based on shadow detection and knowledge of the sun angle. The spacecraft's performance capabilities and the current terrain information are taken into account to establish bounds for reachable and marginally

reachable terrain. Additionally, a scientific interest map defined prior to the flight is considered. This map enables mission designers to influence the algorithm's bias towards or away from specific areas of the map based on their scientific value. All these factors are combined to assign a landing site value to each point on the map.

To address the computational demands, Serrano adopts a supervised learning technique for assigning landing site values using a digital elevation map (DEM), hazard map (craters and rocks), and scientifically interesting map. Furthermore, in a subsequent study by Serrano similar results are obtained by employing fuzzy logic instead of supervised learning [24]. Another learning-based approach is demonstrated by Moghe et al., who utilize Semantic Segmentation, specifically employing a convolutional neural network, to identify secure landing sites [25]. Their approach involves training a model with sample data consisting of DEM inputs and corresponding safety map outputs. Subsequently, the trained model is capable of generating safety maps for new DEMs.

NASA's recent investigations into autonomous LSS encompass the Autonomous Landing Hazard Avoidance Technology (ALHAT) and Safe and Precise Landing – Integrated Capabilities Evolution (SPLICE) programs. The ALHAT approach to LSS is outlined by Cohanim and Collins [8]. The algorithm identifies multiple potential landing sites based on criteria such as proximity to hazards, estimated fuel consumption, and distance to points of scientific interest. Although the creation of these metrics is not explicitly discussed, they serve as inputs for the selection algorithm. Notably, the algorithm incorporates a feature that allows control over the spacing between the selected landing sites. Instead of explicitly modeling footpads and engine bells, the authors represent the vehicle's footprint as a circle. They argue that overly detailed modeling may lead to hazardous spacecraft maneuvers, and attempting to straddle a hazard is not a worthwhile pursuit.

Ivanov et al. conducted a separate investigation within the ALHAT program, resulting in a high-fidelity probabilistic approach to autonomous LSS [26]. In their study, they adopt a probabilistic framework to model the probability of safe landings across various landing orientations during site evaluation. Furthermore, they consider the anticipated navigation uncertainty, acknowledging the possibility of the spacecraft unintentionally landing on adjacent and unsafe terrain. By incorporating these factors, their approach provides a more comprehensive assessment of landing site safety.

In the pursuit of selecting the optimal landing site, Cui et al. take into account numerous safety factors, as described in their study [9]. To assess the terrain, they employ well-established plane fitting techniques to quantify slope and roughness. Additionally, a polynomial guidance law is utilized to determine the fuel consumption associated with reaching various points on the map. The researchers explicitly consider touchdown performance by simulating execution errors and accounting for constant wind disturbances. To identify the landing site with the lowest cost, they conduct a traversal search using a bubble sort algorithm.

Luna et al. present a study where they demonstrate the efficacy of the Simple Safe Site Selection (S4) algorithm, initially developed by Johnson and Mandalia [10, 27]. Their findings reveal comparable results to the algorithms developed under the ALHAT program, showcasing the algorithm's effectiveness. The S4 algorithm prioritizes

computational simplicity and utilizes a single flash lidar image to extract information on slope, roughness, and rock locations. These details are then utilized to generate a landing safety map. Unlike other parameters such as fuel consumption, S4 focuses solely on these factors. Similarly, Mango, Opromolla, and Schmitt propose another lidar-based algorithm [11]. Their approach involves employing a larger digital elevation map (DEM) generated from multiple lidar images. Additionally, they introduce a technique that initially performs a coarse resolution search, narrowing down the search space for subsequent fine resolution searching.

Jung et al. propose a unique approach in which constant elevation contour lines are fitted to a triangular Digital Terrain Map (DTM) [28]. These contour lines serve to identify rocks and slope characteristics. The authors acknowledge the challenges associated with their method, particularly related to the choice of contour step and the potential loss of hazards between contours. To address these issues, the algorithm involves the evaluation of multiple conditionals which results in a relatively complex decision tree.

In practical applications, the maps used for LSS often involve a combination of offline and online mapping techniques. Yang et al. explicitly consider this scenario by integrating LSS with Simultaneous Localization and Mapping (SLAM) [29]. They process a 3D point cloud to derive a 2D Digital Elevation Map (DEM). The selection of a landing site is based on identifying a region within the DEM that exhibits sufficient flatness. This entire process is performed online on a unmanned aerial vehicle (UAV). However, it should be noted that due to the different platform characteristics certain spacecraft-specific factors such as fuel consumption are not taken into account.

Schoppmann et al. present a detailed algorithm for multi-resolution landing site detection specifically designed for unmanned aerial vehicles [30]. Their system utilizes a multi-resolution map generated by an on-board structure-from-motion program. The map is organized into a Laplacian pyramid, dividing terrain features into layers based on their frequency content. The LSS process incorporates considerations of slope, roughness, and mapping quality.

Trawny et al. implement a complete LSS system that demonstrates real-time performance on a flight vehicle, specifically the Morpheus rocket powered vertical take-off and landing vehicle within the ALHAT program [5]. Morpheus was equipped with a gimbaled flash lidar system to scan the terrain. Lidar measurements are fed into the aforementioned LSS algorithm developed by Ivanov et al [26]. Then a landing site is selected and the vehicle autonomously executes the touchdown. The same lidar measurements are employed in this work to validate our system.

During the Chinese National Space Administration's Chang'E 3 mission to the moon, a hazard detection and avoidance scheme was employed which utilized both camera images and lidar measurements [31]. The lidar measurements are processed using a least squares method to fit a ground plane, enabling the detection of hazards. A spiral search pattern is then employed on the ground plane to evaluate potential landing sites based on their fuel requirements and the safety indicated by the lidar measurements.

Several Martian landers have also implemented hazard detection and avoidance techniques. For instance, the Mars 2020 (Perseverance) rover utilized camera-based terrain relative navigation to localize itself with respect to an onboard

map, enabling the selection of the safest reachable landing site within a co-registered hazard map [1]. Notably, these hazards were identified prior to flight from orbital imagery while this work aims to identify hazards during descent. The Tianwen-1 lander is similar to our system in that it combined camera and lidar images to identify hazards and select a safe landing site during its hover mode [32]. Following the hover mode, the lander entered a hazard relative navigation phase, actively avoiding hazards and maneuvering toward the designated safe landing site. That lander's hazard detection scheme was inherited from the Chang'E 3 mission and does not make use of quadtrees or planar estimates for small terrain sections like ours does.

III. Preliminaries

A. Quadtrees

The quadtree [6] can be thought of as a variable-resolution DEM, an example is shown in Figure 1. To achieve variable resolution, each cell can be divided to four children as demonstrated in Figure 1. The variable resolution allows for efficient representation of terrain which varies in complexity. This covers environments such as the Moon or Mars where sparse rocks and craters are separated by relatively flat and featureless plains. The rocks and craters can be accurately captured at a high resolution while the plains are efficiently represented at a low resolution. The natural behavior of terrain measurements during descent also justifies quadtree useage. Sensors such as lidar or cameras typically produce higher density terrain measurements at lower altitude. A higher measurement density implies higher fidelity model of the terrain. The quadtree naturally complements this behavior by allowing terrain scanned at lower altitudes to be seamlessly represented with a higher resolution.

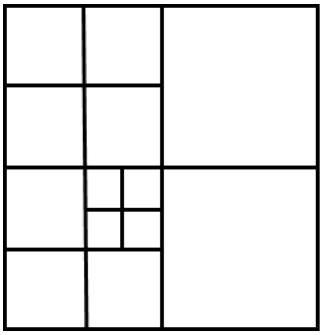


Fig. 1 A quadtree whose western cells and northeastern subcell of the southwestern cell have been divided.

B. Parameterization of Unit Vectors

Unit vectors exist on a 2D manifold in 3D space, the unit sphere, $\mathcal{S}^2$. This causes difficulties when applying traditional estimation techniques, which provide updates within an unrestricted space. To alleviate these issues, we represent unit vectors with the parameterization proposed by Hertzberg *et. al.* [13] and further developed by Gallant

[33] and Hewitt [34]. A vector $\boldsymbol{n} \in S^2$ is separated into a vector component, $\boldsymbol{\kappa} \in \mathbb{R}^2$, lying on the x-y plane and scalar component, $\lambda = \cos(\phi)$, in the z direction. Here, ϕ, is the angle between $\boldsymbol{n}$ and the z axis.

$$\boldsymbol{n} = \begin{bmatrix} \boldsymbol{\kappa} \\ \lambda \end{bmatrix} \tag{1}$$

We consider the notion of an identity unit vector, $\boldsymbol{o} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. $\boldsymbol{n}$ may be used to construct a rotation matrix $\boldsymbol{R_n}$ between $\boldsymbol{o}$ and $\boldsymbol{n}$ such that $\boldsymbol{n} = \boldsymbol{R_n o}$.

$$\boldsymbol{R_n} = \begin{bmatrix} \boldsymbol{I}_2 - \frac{\boldsymbol{\kappa\kappa}^T}{1+\lambda} & \boldsymbol{\kappa} \\ -\boldsymbol{\kappa}^T & \boldsymbol{\lambda} \end{bmatrix} \tag{2}$$

By defining a unit vector direction $\hat{\boldsymbol{\kappa}} = \boldsymbol{\kappa}/||\boldsymbol{\kappa}||$ we can parameterize $\mathcal{S}^2$ without redundancy using a two dimensional vector $\boldsymbol{\phi} = \phi\hat{\boldsymbol{\kappa}}$ which is a member of a set we term $\mathfrak{s}^2$. Exponential and logarithmic mappings exist which map between $\mathcal{S}^2$ and $\mathfrak{s}^2$.

$$\boldsymbol{n} = \exp(\boldsymbol{\phi}) = \begin{bmatrix} \operatorname{sinc}(\phi)\,\boldsymbol{\phi} \\ \cos(\phi) \end{bmatrix} \tag{3}$$

$$\boldsymbol{\phi} = \log(\boldsymbol{n}) = \begin{cases} \boldsymbol{0}_{2\times 1} & \text{if } \boldsymbol{\kappa} = \boldsymbol{0}_{2\times 1} \\ \operatorname{atan2}(||\boldsymbol{\kappa}||,\, \lambda)\,\hat{\boldsymbol{\kappa}} & \text{otherwise} \end{cases} \tag{4}$$

The sinc function is unnormalized and $\operatorname{sinc}(0) = 1$. The jacobian of a unit vector $\boldsymbol{n} \in \mathcal{S}^2$ with respect to its parameterization $\boldsymbol{\phi} \in \mathfrak{s}^2$ is described in [13] as

$$\frac{\partial \boldsymbol{n}}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \chi \frac{\boldsymbol{\phi}\boldsymbol{\phi}^T}{\phi^2} + \operatorname{sinc}(\phi)\,\boldsymbol{I}_2 \\ \operatorname{sinc}(\phi)\,\boldsymbol{\phi}^T \end{bmatrix} \tag{5}$$

$$\chi = \cos(\phi) - \operatorname{sinc}(\phi) \tag{6}$$

$$\left.\frac{\partial \boldsymbol{n}}{\partial \boldsymbol{\phi}}\right|_{\boldsymbol{\phi}=\boldsymbol{0}_{2\times 1}} = \begin{bmatrix} \boldsymbol{I}_2 \\ \boldsymbol{0}_{2\times 1} \end{bmatrix} \tag{7}$$

Additional operators mimic addition, $\boxplus : \mathcal{S}^2 \times \mathfrak{s}^2 \to \mathcal{S}^2$, and subtraction, $\boxminus : \mathcal{S}^2 \times \mathcal{S}^2 \to \mathfrak{s}^2$. We denote λ_m as the

scalar component of another unit vector $\boldsymbol{m}$.

$$\boldsymbol{m} = \boldsymbol{n} \boxplus \boldsymbol{\phi} = \boldsymbol{R_n} \exp(\boldsymbol{\phi}) \tag{8}$$

$$\boldsymbol{\phi} = \boldsymbol{n} \boxminus \boldsymbol{m} = \begin{cases} \log\left(\boldsymbol{R}_{\boldsymbol{m}}^T \boldsymbol{n}\right) & \text{if } \boldsymbol{n}^T \boldsymbol{m} \geq 0,\ \lambda_m \geq 0 \\ \log\left(-\boldsymbol{R}_{-\boldsymbol{m}}^T \boldsymbol{n}\right) & \text{if } \boldsymbol{n}^T \boldsymbol{m} \geq 0,\ \lambda_m < 0 \\ \log\left(\boldsymbol{R}_{-\boldsymbol{m}}^T \boldsymbol{n}\right) & \text{if } \boldsymbol{n}^T \boldsymbol{m} < 0,\ \lambda_m \geq 0 \\ \log\left(-\boldsymbol{R}_{\boldsymbol{m}}^T \boldsymbol{n}\right) & \text{if } \boldsymbol{n}^T \boldsymbol{m} < 0,\ \lambda_m < 0 \end{cases} \tag{9}$$

$\boxplus$ can be used to represent uncertainty in a unit vector with respect to its mean value or perturb it with noise. Consider an estimated unit vector with mean $\bar{\boldsymbol{n}} \in \mathcal{S}^2$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$. The estimate is related to the true vector with a perturbing Gaussian random variable.

$$\boldsymbol{n} = \bar{\boldsymbol{n}} \boxplus \delta\boldsymbol{\phi} \tag{10}$$

$$\delta\boldsymbol{\phi} \sim \mathcal{N}\left(\boldsymbol{0}_{2\times 1},\ \boldsymbol{\Sigma}\right) \tag{11}$$

Using $\boxminus$ we may find a representation of the error between two unit vectors.

$$\delta\boldsymbol{\phi} = \boldsymbol{n}_1 \boxminus \boldsymbol{n}_2 \tag{12}$$

C. Terrain Measurements

This work utilizes terrain measurements produced via lidar. Our system can accommodate either a scanning or flash lidar. Scanning lidar contains a small number of detectors mounted on a gimbal. This gimbal enables scanning the surrounding terrain to produce a larger field of view. The gimbal will typically move through a fixed pattern, collecting a large number of measurements within a small time frame. Flash lidar provides a 2D array of range measurements which are gathered simultaneously. We refer to the scanning measurements in one iteration of the fixed pattern or the flash array of measurements collectively as a single scan. Each measurement within the scan is calibrated to an azimuth, θ_{AZ}, and elevation, θ_{EL}, relative to the lidar's principle axis. We assume zero azimuth and elevation calibration error, but that each range measurement, $\hat{\rho}_i$, has been corrupted with additive Gaussian noise according to Equation (13). In this work we assume lidar measurements are unbiased; an assumption which may not be valid for all systems. Future work will seek to account for these biases.

$$\hat{\rho}_i = \rho_i + \nu_{\rho_i} \qquad \nu_{\rho_i} \sim \mathcal{N}\left(0,\ \Sigma_\rho\right) \tag{13}$$

A measurement in range, azimuth, elevation space can be mapped to Cartesian space in the lidar defined coordinate frame according to Equation (14). Since scanning measurements are not collected simultaneously, a correction must be applied using estimates of the lidar's motion such that each measurement is transformed into a common frame. This problem is not trivial and our example uses flash lidar data so we have not implemented such a correction.

$$\boldsymbol{r}^{\mathcal{L}} = h_1\left(\rho,\, \theta_{\mathrm{AZ}},\, \theta_{\mathrm{EL}}\right) = \begin{bmatrix} \rho \sin\left(\theta_{\mathrm{AZ}}\right) \cos\left(\theta_{\mathrm{EL}}\right) \\ \rho \sin\left(-\theta_{\mathrm{EL}}\right) \\ \rho \cos\left(\theta_{\mathrm{AZ}}\right) \cos\left(\theta_{\mathrm{EL}}\right) \end{bmatrix} \tag{14}$$

Measurements are collected by spacecraft-mounted lidar. In this work we denote the inertial frame $\mathcal{I}$ and the lidar frame $\mathcal{L}$. The lidar's pose in inertial space is represented with an estimated position, $\hat{\boldsymbol{m}}^{\mathcal{I}}$, and transformation matrix, $\hat{\boldsymbol{R}}_{\mathcal{I}\leftarrow\mathcal{L}}$. A measurement in the lidar defined coordinate frame can be mapped to the inertial one according to Equation (15).

$$\boldsymbol{r}^{\mathcal{I}} = h_2\left(\boldsymbol{r}^{\mathcal{L}},\, \boldsymbol{m}^{\mathcal{I}},\, \boldsymbol{R}_{\mathcal{I}\leftarrow\mathcal{L}}\right) = \boldsymbol{R}_{\mathcal{I}\leftarrow\mathcal{L}}\boldsymbol{r}^{\mathcal{L}} + \boldsymbol{m}^{\mathcal{I}} \tag{15}$$

Uncertainty in the lidar's position is modeled as additive zero-mean Gaussian noise, $\boldsymbol{\nu_m} \sim \mathcal{N}\left(\mathbf{0}_{3\times1},\, \boldsymbol{\Sigma_m}\right)$. Rotation uncertainty is expressed in the tangent space of SO(3). The tangent space perturbation, $\delta\boldsymbol{R}$, is modeled as zero-mean Gaussian noise with covariance $\boldsymbol{\Sigma_R}$ and corrupts the true lidar rotation with an exponential map according to Equation (16). Additional details are outlined by Sola *et. al.* [35].

$$\hat{\boldsymbol{R}}_{\mathcal{I}\leftarrow\mathcal{L}} = \boldsymbol{R}_{\mathcal{I}\leftarrow\mathcal{L}} \exp\left(\delta\boldsymbol{R}\right) \qquad\qquad \delta\boldsymbol{R} \sim \mathcal{N}\left(\mathbf{0},\, \boldsymbol{\Sigma_R}\right) \tag{16}$$

Linearization of $\boldsymbol{r}^{\mathcal{I}}$ allows us to approximate the uncertainty of a measurement in inertial space as $\boldsymbol{\Sigma}_{\boldsymbol{r}^{\mathcal{I}}}$.

$$\boldsymbol{\Sigma}_{\boldsymbol{r}^{\mathcal{I}}} = \frac{\partial h_2}{\partial \delta\boldsymbol{R}} \boldsymbol{\Sigma_R} \frac{\partial h_2}{\partial \delta\boldsymbol{R}}^{T} + \frac{\partial h_2}{\partial \boldsymbol{m}^{\mathcal{I}}} \boldsymbol{\Sigma_m} \frac{\partial h_2}{\partial \boldsymbol{m}^{\mathcal{I}}}^{T} + \frac{\partial h_2}{\partial \boldsymbol{r}^{\mathcal{L}}} \frac{\partial h_1}{\partial \rho} \Sigma_\rho \frac{\partial h_1}{\partial \rho}^{T} \frac{\partial h_2}{\partial \boldsymbol{r}^{\mathcal{L}}}^{T} \tag{17}$$

IV. Mapping System

Our system is designed to operate onboard a spacecraft during terminal descent to landing. The spacecraft is descending over uncertain terrain and requires a hazard-free landing zone. Using a lidar system, the spacecraft maps portions of the terrain in which it would like to land. An onboard localization system provides estimates of the spacecraft's pose in a planet fixed coordinate system. As previously mentioned, the terrain is mapped using a quadtree. The tree has a predefined minimum and maximum resolution and is automatically divided to its minimum upon initialization. The tree also has a predefined x-y domain, lidar measurements falling outside the domain are discarded.

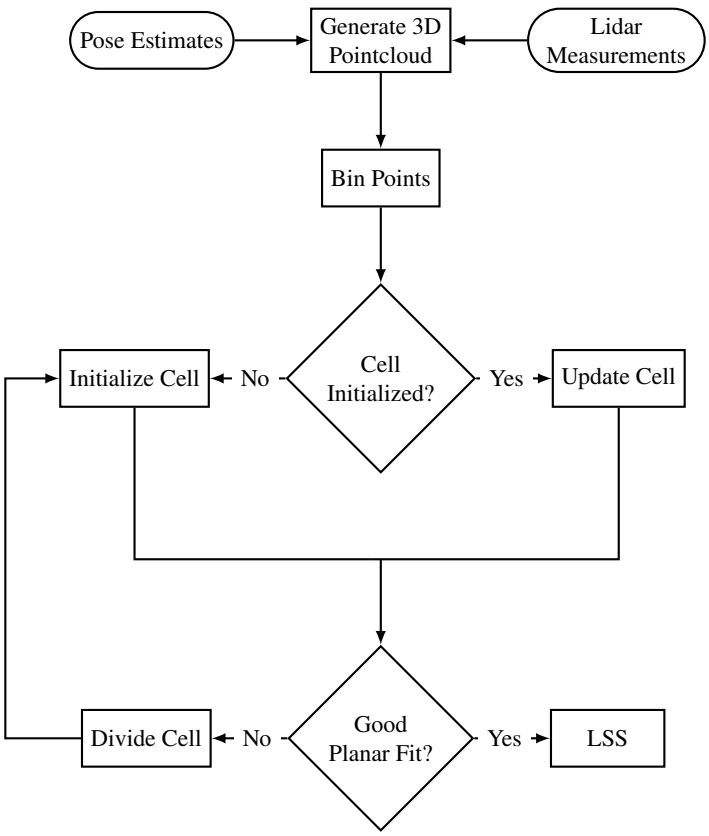


Fig. 2 An overview of the mapping system. Each node is discussed in detail in the following sections.

Each quadtree cell contains a planar estimate of its terrain. Cells are divided when the system detects the plane is not a good approximation for the underlying terrain. Parent cells contain planar estimates which are a fusion of their children. An overview of the system is shown in Figure 2.

A. Map Initialization

Measurements from each lidar scan are converted to inertial space using Equations (14) and (15). The measurements in inertial space are binned into their respective cells in the x-y domain of the quadtree. If a prior estimate does not exist and enough measurements are present, a Maximum Likelihood (ML) initialization scheme provides cell's initial estimate. If the number is insufficient, the measurements are stored to be used in an initialization with subsequent lidar scans. The fit quality between measurements and the initial estimate is used to consider instantaneous division of the new estimate.

Initialization employs a modified Newton-Raphson method to find the ML planar estimate from a set of lidar measurements mapped to a cartesian planet-fixed space. Throughout this work, we neglect planetary rotation and implicitly assume that the planet-fixed space is inertial. A plane is defined by its unit-normal vector, $\boldsymbol{n}$, and its orthogonal

distance from the origin d. Any point, $\boldsymbol{r}^*$, which lies on the plane satisfies the following constraint.

$$\boldsymbol{n}^T \boldsymbol{r}^* - d = 0 \tag{18}$$

We formulate the orthogonal distance between the i$^{\text{th}}$ lidar measurement, $\hat{\boldsymbol{r}}_i^I$, and the plane as an error:

$$\epsilon_i = \boldsymbol{n}^T \hat{\boldsymbol{r}}_i^I - d \tag{19}$$

Previous work [14] has shown how to find the ML estimate of a plane by minimizing the summed square error of all points. However, that work assumed all measurements were corrupted by independent noise. In this work, lidar pose estimate error provides the dominant contribution to measurement error in inertial space. This error source is common to all measurements within a lidar scan, invalidating the previous work's assumptions.

The error in $\hat{\boldsymbol{r}}_i^I$ can be divided into two components; one which is orthogonal to $\boldsymbol{n}$ and another which is not. We take care to note that components orthogonal to $\boldsymbol{n}$ are within the plane of interest. It can be shown that this orthogonal component does not contribute to planar measurement error and can be neglected under the assumption that the terrain is truly planar. The error component along $\boldsymbol{n}$ for the j$^{\text{th}}$ scan is denoted σ_j. It can be modeled as a bias term and we can express the new error function as Equation (20).

$$\bar{\epsilon}_i = \boldsymbol{n}^T \hat{\boldsymbol{r}}_i^I - d - \sigma_j \tag{20}$$

Correlation between measurements is captured by σ_j and we can once again assume independence between each $\hat{r}_i$. We would like to estimate each σ_j along with $\boldsymbol{n}$ and d but this system is unobservable regardless of the number of measurements obtained. Instead we estimate $\gamma_j = d + \sigma_j$.

$$\epsilon_i = \boldsymbol{n}^T \hat{\boldsymbol{r}}_i^I - \gamma_j \tag{21}$$

To recover an estimate of d, we will assume $\mathrm{E}\left[\sigma_j\right] = 0$. Our estimate of d is provided by Equation (22) where N_s is the number of scans.

$$\hat{d} = \frac{1}{N_s} \sum_{j=1}^{N_s} \hat{\gamma}_j \tag{22}$$

In real-world scenarios correlations in vehicle pose error between lidar scans may invalidate our assumption $\mathrm{E}\left[\sigma_j\right] = 0$. We assume that such correlations are small enough not to significantly degrade system performance.

The Maximum Likelihood (ML) estimate of a state $\boldsymbol{x} = \left[\boldsymbol{n},\ \gamma_1,\ \ldots,\ \gamma_{N_s}\right]^T$ given the set of measurement errors, $\boldsymbol{\epsilon}$,

can be found through a minimization of the negative log-likelihood.

$$\hat{\boldsymbol{x}}_{\mathrm{ML}} = \arg\min_{\boldsymbol{x}} \; -\log\left(p(\boldsymbol{\epsilon}|\boldsymbol{x})\right) \tag{23}$$

Independence of each ϵ_i allows the following simplification.

$$-\log\left(p(\boldsymbol{\epsilon}|\boldsymbol{x})\right) = -\log\left(\prod_{i\in\boldsymbol{\epsilon}} p(\epsilon_i|\boldsymbol{x})\right) \tag{24}$$

Because the distribution of each measurement is Gaussian, the negative log-likelihood can be further simplified as

$$-\log\left(p(\boldsymbol{\epsilon}|\boldsymbol{x})\right) \propto \sum_{i\in\boldsymbol{\epsilon}} \frac{1}{2}\epsilon_i^2 \boldsymbol{n}^T \boldsymbol{\Sigma}_{\boldsymbol{r}_i^I}^{-1} \boldsymbol{n}. \tag{25}$$

Equation (25) cannot be optimized in closed form outside the special case when $\boldsymbol{\Sigma}_{\boldsymbol{r}_i^I} \propto \boldsymbol{I}_3$. Instead, incremental optimization must be used to find $\hat{\boldsymbol{x}}_{\mathrm{ML}}$. To this end, we define $\boldsymbol{x}^k$ and γ_j^k as the estimates at the k^{th} iteration of the optimization. If we allow $\boldsymbol{n}^k$ and each γ_j^k to be perturbed by $\delta\boldsymbol{\phi}$ and $\delta\gamma_j$ we find the following expression for error.

$$\epsilon_i = \left(\boldsymbol{n}^k \boxplus \delta\boldsymbol{\phi}\right)^T \hat{\boldsymbol{r}}_i^I - (\gamma_j^k + \delta\gamma_j) \tag{26}$$

The perturbation of the normal vector can be shown to be linearized as follows

$$\boldsymbol{n}^j \boxplus \delta\boldsymbol{\phi} \approx \boldsymbol{R}_{\boldsymbol{n}^j} \begin{bmatrix} \delta\boldsymbol{\phi} \\ 1 \end{bmatrix} \tag{27}$$

where $\boldsymbol{R}_{\boldsymbol{n}^j}$ is a rotation matrix as defined in a previous section.

Substituting the Equation (27) into Equation (25) and employing linearization about $\boldsymbol{n}^k$ and each γ_j^k results in the following expression for ϵ_i.

$$\epsilon_i \approx A_i - \boldsymbol{D}_i \begin{bmatrix} \delta\boldsymbol{\phi} & \gamma_1^k & \dots & \gamma_{N_s}^k \end{bmatrix}^T \tag{28}$$

where

$$A_i = \left(\hat{\boldsymbol{r}}_i^I\right)^T \boldsymbol{n}^j - d^j \tag{29}$$

$$\boldsymbol{D}_i = \begin{bmatrix} \boldsymbol{B}_i & \boldsymbol{1}_{1\times N_s} \end{bmatrix} \tag{30}$$

$$\boldsymbol{B}_i = -\left(\hat{\boldsymbol{r}}_i^I\right)^T \boldsymbol{R}_{\boldsymbol{n}^j} \begin{bmatrix} \boldsymbol{I}_2 \\ \boldsymbol{0}_{1\times 2} \end{bmatrix} \tag{31}$$

Here $\mathbf{1}_{R\times C}$ is a matrix of ones with R rows and C columns.

Defining $\boldsymbol{\Delta}^T = \begin{bmatrix} \delta\boldsymbol{\phi}^T & \delta\gamma_1 & \dots & \gamma_{N_s} \end{bmatrix}$, the cost function in Equation (25) can be simplified into a quadratic.

$$\sum_{i\in\boldsymbol{\epsilon}} = \frac{1}{2}\epsilon_i^2 \left(\boldsymbol{n}^k\right)^T \boldsymbol{\Sigma}_{\boldsymbol{r}_i^I}^{-1} \boldsymbol{n}^k \tag{32}$$

$$= \frac{1}{2}a - \boldsymbol{\Delta}^T\boldsymbol{\beta} + \frac{1}{2}\boldsymbol{\Delta}^T\boldsymbol{H}\boldsymbol{\Delta} \tag{33}$$

where

$$a = \sum_{i\in\mathcal{I}} A_i^2 \Sigma_{\boldsymbol{\epsilon}_i}^{-1} \qquad \boldsymbol{\beta} = \sum_{i\in\mathcal{I}} \boldsymbol{D}_i^T \Sigma_{\boldsymbol{\epsilon}_i}^{-1} A_i \tag{34}$$

$$\boldsymbol{H} = \sum_{i\in\mathcal{I}} \boldsymbol{D}_i^T \Sigma_{\boldsymbol{\epsilon}_i}^{-1} \qquad \Sigma_{\boldsymbol{\epsilon}_i} = (\boldsymbol{n}^j)^T \boldsymbol{\Sigma}_{\boldsymbol{r}_i^I}^{-1} \boldsymbol{n}^j \tag{35}$$

Setting the derivative of the cost function to zero, the optimal increment, $\boldsymbol{\Delta}^*$, can be found by solving $\boldsymbol{H}\boldsymbol{\Delta}^* = \boldsymbol{\beta}$. The optimal values are then used to update the linearization points as follows:

$$\boldsymbol{n}^{k+1} = \boldsymbol{n}^k \boxplus \delta\boldsymbol{\phi}^*, \quad \gamma_j^{k+1} = \gamma_j^k + \delta\gamma_j^* \tag{36}$$

This update process is repeated until convergence is achieved. At convergence, we can quantify uncertainty in $\hat{x}_{\mathrm{ML}}$ by considering the covariance of $\boldsymbol{\Delta}$. It can be shown that $\mathrm{cov}(\boldsymbol{\Delta}) = \boldsymbol{H}^{-1} = \hat{\boldsymbol{x}}_{\mathrm{ML}}$. The estimate $\hat{d}$ is found according to Equation (22). We take its covariance as the average of the $\hat{\gamma}$ covariances. In reality d is unobservable and this methodology is only an approximation. $\hat{d}$ will be unbiased if the navigation system's pose estimate is also unbiased and uncorrelated from scan to scan. This will never be true in practice, but we find that our method provides acceptable results for the flight test data used in later sections.

B. Fusion of Planar Estimates

We consider a set of planar estimates, $\mathcal{P} = \left[\hat{\boldsymbol{P}}_1, \dots, \hat{\boldsymbol{P}}_{N_\mathcal{P}}\right]$. Each element contains an estimate of the plane's surface normal and orthogonal distance from the origin.

$$\hat{\boldsymbol{P}}_i = \begin{bmatrix} \hat{\boldsymbol{n}}_i \\ \hat{d}_i \end{bmatrix} \tag{37}$$

There is also an associated covariance estimate, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{P}_i}$ defined by Equation (38).

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{P}_i} = \mathrm{E}\left[\boldsymbol{e}_i \boldsymbol{e}_i^T\right] \qquad \boldsymbol{e}_i = \begin{bmatrix} \hat{\boldsymbol{n}}_i \boxminus \boldsymbol{n}_i \\ \hat{d}_i - d_i \end{bmatrix} \tag{38}$$

We require a technique for fusing $\mathcal{P}$ into a single estimate while considering the relative uncertainties of each element. Planar fusion has two uses within our system. First, it will be use to provide parent cells a planar estimate as a function of their children. Second, in our map update scheme, we will fit a planar estimate to incoming lidar scans. This estimate will be fused with the prior, if one exists.

We begin by defining a reference plane, $\boldsymbol{P}_0$. The error between each estimate within $\mathcal{P}$ and $\boldsymbol{P}_0$ can be found as

$$\boldsymbol{E}_i = \begin{bmatrix} \boldsymbol{n}_0 \boxminus \hat{\boldsymbol{n}}_i \\ d_0 - \hat{d}_i \end{bmatrix}. \tag{39}$$

By taking these errors as observations, we can formulate a linear least squares estimate of the error from the reference plane.

$$\boldsymbol{Y} = \left[\boldsymbol{E}_1, \ldots, \boldsymbol{E}_{N_\mathcal{P}}\right]^T \qquad \boldsymbol{R} = \mathrm{blkdiag}\{\boldsymbol{\Sigma}_{\boldsymbol{P}_1}, \ldots, \boldsymbol{\Sigma}_{\boldsymbol{P}_{N_\mathcal{P}}}\} \tag{40}$$

$$\boldsymbol{H} = [\boldsymbol{I}_3, \ldots, \boldsymbol{I}_3] \qquad \hat{\boldsymbol{E}} = \left(\boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{H}\right)^{-1} \boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{Y} \tag{41}$$

$$\hat{\boldsymbol{E}} = \left[\delta\hat{\boldsymbol{\phi}}_{\boldsymbol{E}}^T, \delta\hat{d}_{\boldsymbol{E}}\right] \qquad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{E}} = \left(\boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{H}\right)^{-1} \tag{42}$$

The fused planar estimate is found by applying $\hat{\boldsymbol{E}}$ to the reference plane.

$$\hat{\boldsymbol{n}}_{\boldsymbol{E}} = \boldsymbol{n}_0 \boxplus \delta\hat{\boldsymbol{\phi}}_{\boldsymbol{E}} \tag{43}$$

$$\hat{d}_{\boldsymbol{E}} = d_0 + \delta\hat{d}_{\boldsymbol{E}} \tag{44}$$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{P}_{\boldsymbol{E}}} = \hat{\boldsymbol{\Sigma}}_{\boldsymbol{E}} \tag{45}$$

C. Map Update

Incoming measurements in a new scan are once again binned into their respective cells in the quadtree. When a prior estimate for the plane exists, an ML estimate is found from the incoming points as described in Section IV.A. The plane estimate from the new points is fused with the prior estimate using the algorithm in Section IV.B. After updating base level cells, child cells are fused to obtain hierarchical estimates for their parents using the same algorithm.

D. Map Management

The Mahalanobis distance, Equation (46), can provide a measure of distance between a point, $\boldsymbol{p}$, and Gaussian distribution, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

$$d_M = \sqrt{(\boldsymbol{p} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{p} - \boldsymbol{\mu})} \tag{46}$$

We find the prefit residual for each measurement contributing to a planar estimate and Mahalanobis distance for each residual. By finding the average distance, we formulate a metric describing the quality of the estimate and planar approximation. For Gaussian distributions of dimension k, d_M follows a chi distribution of order k. We use the mean of the chi distribution to establish a threshold for division. For $k = 3$ the expectation of a chi distribution is $\frac{\sqrt{2}\Gamma(2)}{\Gamma(3/2)}$ where $\Gamma(\cdot)$ is the Gamma function. We define our threshold for division as λ_{div}.

$$\lambda_{\text{div}} = 1.2\frac{\sqrt{2}\Gamma(2)}{\Gamma(3/2)} \tag{47}$$

Cells are divided when the average prefit measurement residual Mahalanobis distance of all processed measurements exceeds λ_{div}.

V. Landing Site Selection System

The LSS system selects sites based on roughness and slope. A recursive solution is produced as new information is added to the map. In the following sections we define lander hazards, describe their detection, and overview how sites are identified.

A. Hazard Definition

We consider two forms of hazardous terrain; slope and roughness. Slope is defined as the angle, θ, between the spacecraft's body z-axis and the local gravity vector at touchdown. This is illustrated in Figure 3. A slope hazard occurs when θ exceeds vehicle tolerances. When cells are very small, avoiding slope hazards makes the system more

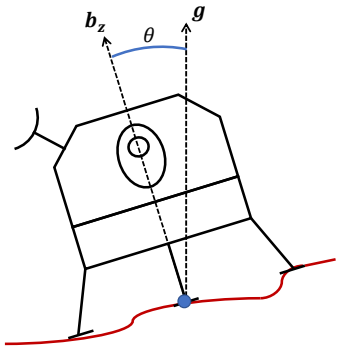


Fig. 3 An illustration of spacecraft slope at touchdown.

conservative as high slopes on very small environmental features may not endanger the vehicle. However, when cells are large avoiding only the best fit slope could obscure danger within a cell. Our mapping system is tuned to avoid such scenarios by dividing based on the quality of the planar approximation. Later, we will discuss a roughness metric, avoided exception features, and subscale features. These items are all designed to detect and avoid hazardous terrain which is obscured by a low cell resolution.

When the vehicle comes to a rest after touchdown, at least three of its landing legs will have contact with the ground. Assuming rigid legs and pads, the contacting pads will define a plane on which the vehicle rests. A roughness hazard occurs when any point of the terrain within the lander's footprint is greater than λ_R above the plane. This is illustrated in Figure 4

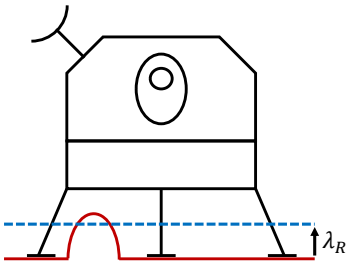


Fig. 4 An illustration of a roughness hazard.

B. Hazard Identification Using Avoided Features

Computational complexity and measurement uncertainty makes exact identification of hazards impractical. Instead, we search the estimated map for features which suggest the presence of hazardous terrain. We refer to these features and the cells which contain them as avoided. Slope hazards are located by inspecting the angle between estimated normal vectors and vertical for all bottom-level cells. An *avoided slope feature* has been found when the angle exceeds slope tolerances. Any cell containing an avoided slope feature cannot be included within a landing site. The system looks for rock-like features which deviate from the surrounding terrain by searching for *avoided elevation features*. This involves comparison of cell estimates at two quadtree levels. We find the maximum distance between estimated planes at the lowest level, and fused planar estimates at a higher one. If the distance exceeds roughness tolerances, an avoided elevation feature has been found. Cells containing avoided features cannot be included within a landing site. Avoided elevation features are illustrated with the 2D cross-section in Figure 5. A third dimension, l, represents the quadtree level and illustrates the multi-level nature of avoided elevation features. The maximum distance, d_i, between each orange base level planar estimate and some higher level green estimate is found. If the distance exceeds roughness tolerances, the base level cell is marked hazardous.

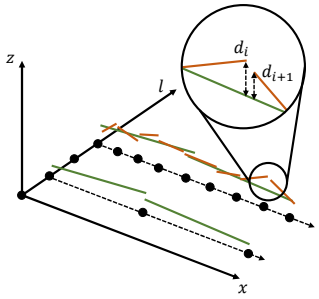


Fig. 5 A 2D cross-section of avoided elevation features.

To detect rock-like features at a scale below the smallest cell size, the system investigates individual measurements. If the orthogonal distance between a measurement and a bottom level estimate exceeds roughness tolerances, that measurement is termed an exception. If the fraction of measurements which are exceptions exceeds tolerances, the cell contains an *avoided exception feature* and cannot be included in a landing site. These features are detected probabilistically. We find the covariance of a measurement in inertial space using Equation (17) and marginalize the degrees of freedom which are parallel to the estimated plane. We are left with a one dimensional Gaussian distribution representing the point's distance from the plane. We find the probability mass, ξ_i, for each point which is outside of roughness tolerances. The fraction of measurements which are exceptions is then given as

$$E_e = \frac{\sum_{i=1}^{N_{\text{meas}}} \xi_i}{N_{\text{meas}}} \tag{48}$$

If $E_e > \lambda_e$ the cell contains an avoided exception feature. ξ_i is illustrated for several measurements in Figure 6. Blue measurements are shown along with Gaussian approximations of their uncertainty orthogonal to the planar estimate. Hashed regions denote the probability mass of each measurement which exceeds tolerances. If E_e exceeds tolerances, the cell base level cell is marked hazardous.

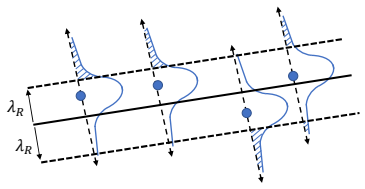


Fig. 6 An illustration of probabilistic exception hazards.

C. Subscale Feature Estimation

It is likely that a lander will encounter rocks or other features which are within allowable tolerances and would not jeopardize mission success. Nonetheless, we would like to avoid these features when selecting a landing site to minimize mission risk. If these features are significantly smaller than the minimum allowable quadtree resolution, they will not be well represented by the planar approximations. For this reason, we term these features *subscale features*.

The size and shape of these subscale features are estimated with an EKF independently from the mapping process. We model the elevation within each base level as a function of several parameters according to Equation (49). This model's form is inspired by the Gaussian distribution; however, it represents local terrain rather than a probability density and does not integrate to one.

$$z(x, y, X_s) = A + \frac{k^2}{4\pi\sigma^4} e^{-\frac{1}{2}\begin{bmatrix} x-\mu_x & y-\mu_y \end{bmatrix} \boldsymbol{P}^{-1} \begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix}} \tag{49}$$

$$\boldsymbol{P} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \tag{50}$$

The state vector for each cell and the subscale feature it may contain is

$$X_s = \begin{bmatrix} A \\ k \\ \mu_x \\ \mu_y \\ \sigma \end{bmatrix} \tag{51}$$

It is not strictly necessary for the diagonal elements of $\boldsymbol{P}$ to be identical but doing so reduces the dimension of X_s. This choice also makes the features axially symmetric about a vertical axis through their peak. This is a good assumption for the environmental features contained in the data set this work is validated with.

Our measurements for estimating $\hat{X}_s$ are the z coordinates of the lidar measurements after mapping into inertial space, $\boldsymbol{r}^I$. After each scan, we process all measurements in a batch EKF update. Estimating subscale features in this manner inherently assumes each cell contains exactly one subscale feature. There is no guarantee that this is the case. However, the presence of multiple subscale features implies the planar approximation is poor. We note that in the following flight test results there are no base level cells with more than one subscale feature as our division logic detects poor planarity and divides accordingly.

When processing the first scan, we initialize our estimates with mean $\bar{X}_s$ and covariance $\bar{\boldsymbol{\Sigma}}_s$ as shown in Equation (52).

$$\bar{X}_s = \begin{bmatrix} \text{mean}\left(\boldsymbol{r}_z^{\mathcal{I}}\right) \\ 1 \\ \check{\boldsymbol{r}}_x^{\mathcal{I}} \\ \check{\boldsymbol{r}}_y^{\mathcal{I}} \\ 0.1 \end{bmatrix} \qquad \bar{\boldsymbol{\Sigma}}_s = \boldsymbol{I}_{5\times 5} \tag{52}$$

Where $\check{\boldsymbol{r}}_x^{\mathcal{I}}$ and $\check{\boldsymbol{r}}_y^{\mathcal{I}}$ are the x and y components of the measurement with the largest $\boldsymbol{r}_z^{\mathcal{I}}$.

We detect the absence of subscale features by inspecting the parameter estimates. First, we consider $\hat{k}^2$ which is equivalent to the volume of the subscale feature. We discard our update and assume the cell is subscale feature free when $\hat{k}^2$ is small. The subscale feature's elevation above its surroundings is found as $z(\hat{\mu}_x, \hat{\mu}_y, \hat{X}_s) - \hat{A}$. Again, the subscale feature is discarded when the estimated elevation is low. Lastly, the system occasionally estimates that the hazard-less cell's entire area makes up the surface of a wide and flat subscale feature. To eliminate these cases, we discard subscale features with a large value for $\hat{\sigma}$ which is proportional to diameter.

To demonstrate subscale feature estimation performance we simulate a hemispherical object with radius ρ on a flat plane. We sample x and y uniformly over the surface of the terrain and evaluate the z coordinate according to Equation (53) to find $\boldsymbol{r}^{\mathcal{I}}$. Each $\boldsymbol{r}^{\mathcal{I}}$ is corrupted with zero-mean Gaussian noise with covariance $\boldsymbol{R}$. These noisy measurements and the posterior estimate of X_s are shown in Figure 7.

$$z = \begin{cases} \sqrt{\rho - x^2 - y^2} & \text{if } \sqrt{x^2 + y^2} < \rho \\ 0 & \text{otherwise} \end{cases} \tag{53}$$

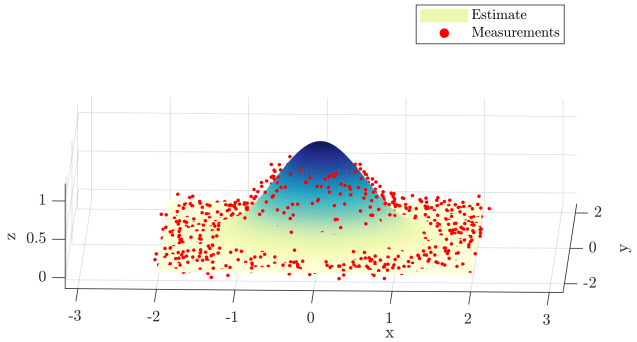


Fig. 7 Noisy measurements sampled from a hemispherical surface and the estimated subscale feature.

D. Site Identification

Site scoring is a function of several parameters. Two of these are slope, θ, and roughness, R. Slope can be found simply as the angle between a cell's normal vector and vertical according to Equation (54).

$$\theta = \cos^{-1}\left(\hat{\boldsymbol{n}}^T \boldsymbol{z}\right) \tag{54}$$

Roughness is a quantification of the deviation of lidar measurements from the cell's planar estimate. Our mapping system manages cells with variable area and measurement density. The roughness density metric must be invariant to changes in these quantities. This means that the hazard level of a large cell with a single rock and low measurement density will be the same as the hazard level of a small cell with the same rock and high measurement density. We define roughness density according to Equation (55) where d_i is the orthogonal distance between a measurement and the cell's planar estimate and A is the cell's area.

$$R = \frac{A}{N_d} \sum_{i=1}^{N_d} |d_i| \tag{55}$$

Site scores, I, are evaluated as the integral of the score density ρ^I within the landing zone. The landing zone, $\mathcal{Z}$, is defined as a circular region accounting for the vehicle's size and anticipated guidance, navigation, and control uncertainty. If any portion of $\mathcal{Z}$ touches a hazardous cell, it is not considered for landing. The optimal site is perfectly smooth, level, without subscale features, and far from any identified hazards.

$$I = \int_{\mathcal{Z}} \rho^I \, dA \tag{56}$$

ρ^I is a function of three metrics which are defined uniformly for each cell and one which is defined at a resolution higher than the base cell level. Slope, θ, roughness density, R and nearest hazard distance, d_H are defined for each cell. The final component, subscale feature elevation, z_s, is naturally defined at a higher resolution than the minimum cell level. It can be easily found by evaluating the Gaussian-inspired terrain model with the estimated parameters for each x and y point. $\hat{A}$ represents the ground-plane elevation and is neglected.

$$z_s = \frac{\hat{k}^2}{4\pi\hat{\sigma}^4} e^{-\frac{1}{2}\begin{bmatrix} x-\hat{\mu}_x & y-\hat{\mu}_y \end{bmatrix} \hat{\boldsymbol{P}}^{-1} \begin{bmatrix} x-\hat{\mu}_x \\ y-\hat{\mu}_y \end{bmatrix}} \tag{57}$$

Weighting parameters, α_θ, α_R, α_H, and α_s allow the user to select the relative importance of each metric as shown

in Equation (58). These metrics are found before sites can be scored.

$$\rho^I = \alpha_\theta \theta + \alpha_R R - \alpha_H d_H + \alpha_s z_s \tag{58}$$

The integral I is found by approximating Equation (56) as a discrete sum. Potential sites are defined by a uniform grid across the map whose spacing is controlled by the user. ρ^I is evaluated at each grid point. A $\mathcal{Z}$ is constructed at each grid point which contains $N_\mathcal{Z}$ adjacent points. To approximate I the ρ^I values within $\mathcal{Z}$ are averaged.

$$I = \sum_{i=1}^{N_\mathcal{Z}} \frac{\rho_i^I}{N_\mathcal{Z}} \tag{59}$$

VI. Quadtree and Fixed-Resolution Mapping Comparison

A. Simulation Details

To demonstrate the benefits of our variable-resolution mapping system we compare its performance to a fixed-resolution mapping system with a simulated descent. Our simulated vehicle is equipped with a 128×128, 10 degree field of view, flash lidar array. It descends onto a parabolic terrain with a flattened bottom as described by Equation (60).

$$z(x, y) = \begin{cases} 3 \times 10^{-3} \left(x^2 + y^2 - 12^2\right) & \sqrt{x^2 + y^2} > 12 \\ 0 & \text{otherwise} \end{cases} \tag{60}$$

The vehicle begins with an initial position and velocity of $\begin{bmatrix} 100 & -100 & 500 \end{bmatrix}^T$ m and $\begin{bmatrix} -5 & -5 & -5 \end{bmatrix}^T$ m s^{-1}. The vehicle moves along a smooth trajectory defined as a polynomial with zero terminal velocity and terminal position of 20m above the origin after 60 seconds. Flash lidar scans are collected once per second. The vehicle is oriented such that the flash lidar boresight always points at a virtual target located along the x-axis 10 meters from the origin. The simulated terrain and trajectory are shown in Figure 8.

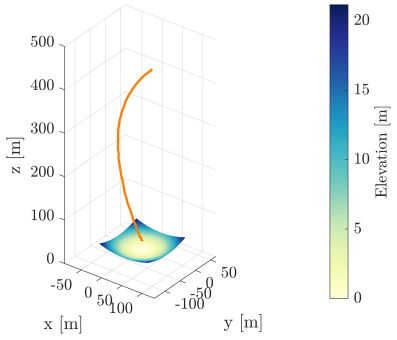


Fig. 8 The terrain and vehicle trajectory (orange) for comparing quadtree and fixed-resolution mapping.

The flash lidar measurements are corrupted using zero-mean Gaussian noise with a 8 cm 1σ. The true vehicle pose is not provided to the mapping systems. Instead, the position and orientation are corrupted with noise as described in Section III.C. For sampling the zero-mean Gaussian noise we use $\boldsymbol{\Sigma_m} = 0.5^2\boldsymbol{I}_3$ m^2 and $\boldsymbol{\Sigma_R} = 0.01^2\boldsymbol{I}_3$ deg^2.

B. Fixed-Resolution Mapping System

For the fixed resolution mapping system we use our quadtree mapping system with minimum and maximum cell sizes set to the same value. The fixed-resolution map has a maximum area of 14400 m^2 consisting of 64×64 square cells. Each cell has a side length of 1.875 m and contains a planar estimate of the terrain if at least three measurements from the same scan are present.

C. Simulation Results

Figures 9 and 10 show the boundaries of every fifth lidar scan as well as the map cell divisions for the quadtree and fixed-resolution maps. After all scans are processed, the quadtree contains 232 base level cells compared to the fixed-resolution map's 2852. The fixed-resolution map estimate therefore requires an order of magnitude larger memory footprint.

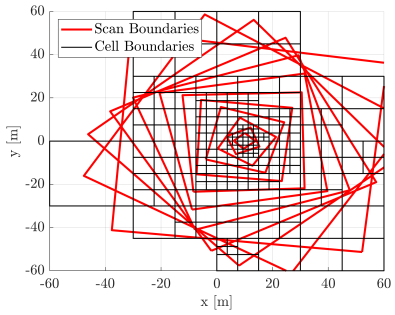


Fig. 9 The boundaries of the base level quadtree cells alongside the boundaries of every fifth lidar scan.

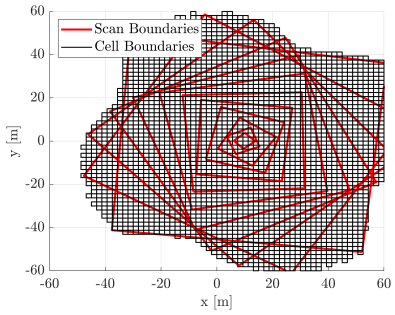


Fig. 10 The boundaries of the fixed-resolution cells alongside the boundaries of every fifth lidar scan.

Figure 11 compares lidar scan processing times on a consumer-grade laptop running MATLAB. We note that this implementation contains no timing guarantees and has not been optimized for performance in an embedded system.

Nevertheless, the quadtree map generally processes scans faster than the fixed-resolution one. Early scans are processed much faster with the quadtree as far fewer cells are required to model the terrain when the vehicle is at a high altitude. As the vehicle descends the resolution of cells updated within the quadtree approaches that of the fixed-resolution map and the two systems show similar processing times. Some scans require substantially higher processing times for the quadtree map. This is because these scans have triggered a number of quadtree cell divisions which is a computationally intensive process. Despite this, the quadtree offers faster processing overall. The time required to process all scans for the quadtree and fixed-resolution maps is 187 and 220 seconds, respectively.

Fig. 11 The required time to process each lidar scan for the quadtree and fixed-resolution maps.

Scans taken at high altitude result in a lower effective measurement density on the surface. The planar estimate quality is a strong function of the number of measurements used. With its higher resolution, fixed-resolution planar estimates are created from a low number of measurements when the vehicle's altitude is high. This generally results in higher elevation estimate error for the fixed-resolution map. Elevation estimate error for the quadtree map and fixed-resolution map is shown in Figures 12 and 13. The coloring scheme has been saturated at a maximum of 3 m to ease comparison between the two figures. Only the fixed-resolution map contains errors higher than 3 m. Elevation estimate error is generally higher in the peripheries of the fixed-resolution map due to that terrain being scanned only at high altitude when measurement density is low. One exception to this is the westernmost quadtree cell. This cell shows high error levels because lidar measurements are only present on its eastern half and so its planar estimate is only an approximation of the quadratic terrain in that half.

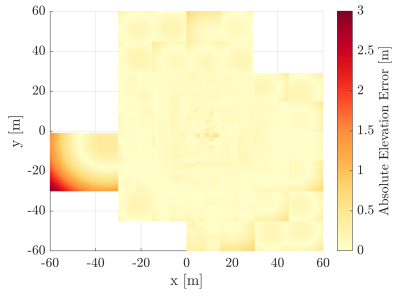


Fig. 12 The elevation estimate error for the quadtree mapping system.

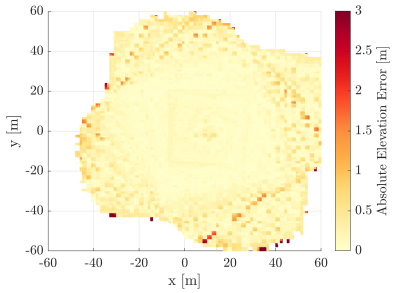


Fig. 13 The elevation estimate error for the fixed-resolution mapping system.

VII. Quadtree Mapping and Landing Site Selection with Flight-Test Data

A. Flight-Test Data

We evaluate our system with the data gathered on a rocket-powered vertical take-off and landing vehicle shown in Figure 14. This vehicle, known as Morpheus, measures 3.7 meters in diameter, has a dry mass of 1100 kilograms, and a propulsion system capable of producing 24,000 Newtons of thrust. It also has a variety of senors onboard, including a gimbal-mounted flash lidar and inertial measurement unit. Morpheus test flights were conducted under the ALHAT program between 2011 and 2014. One of the final test campaigns (free flights 10 through 14) involved flying Morpheus over a simulated lunar hazard field at Kennedy Space Center in March 2014. We use the data gathered during free flights 13 and 14 (FF13 and FF14) [5].

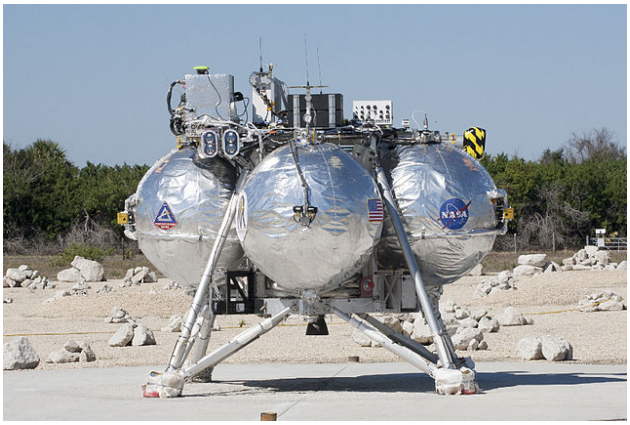


Fig. 14 The Morpheus Vertical Take-off and Landing Vehicle. Image Credit: NASA.

Each flight lasted approximately 96 seconds and included multiple phases. During a 20 second scanning phase, the gimbal-mounted lidar swept the terrain shown in Figure 15. We use lidar measurements and spacecraft pose estimates from this phase of flight to produce a landing site solution. The terrain which was scanned measures 100 by 100 meters and contains a variety of hazards such as rocks and craters. These hazards are representative of the lunar surface in size and distribution. The largest boulders are semi-spherical and approximately one meter in diameter. The largest craters are approximately a half meter in depth and five meters in diameter. The portion of the Morpheus flight profile

used for LSS can be divided into two components; scanning and feature lock. During the scanning portion, the flash LiDAR sweeps through the terrain and gathers measurements over a wide area. During the feature lock portion, the flash lidar focuses on a specific map feature for visual odometry. As a consequence, no additional measurements are taken from large portions of the map after the scanning component ends. At the end of the feature lock segment, the lidar is gimbaled upwards and away from the terrain to protect its lens from dirt and debris during landing. We only use measurements gathered during the scanning portion of flight to map and select landing sites.

Fig. 15 The man-made terrain which was scanned by Morpheus. Image Credit: NASA.

Figure 16 shows a truth DEM of the man-made terrain. Morpheus did not scan the entire man-made terrain. Instead, it scanned only the approximately 60×60 meter portion of terrain outlined in red. Notably, some of the grassy terrain shown in Figure 15 was scanned and is free of rocks or major hazards. Our landing site selection system would naturally favor this smooth terrain so we opt to manually remove any measurements of it.

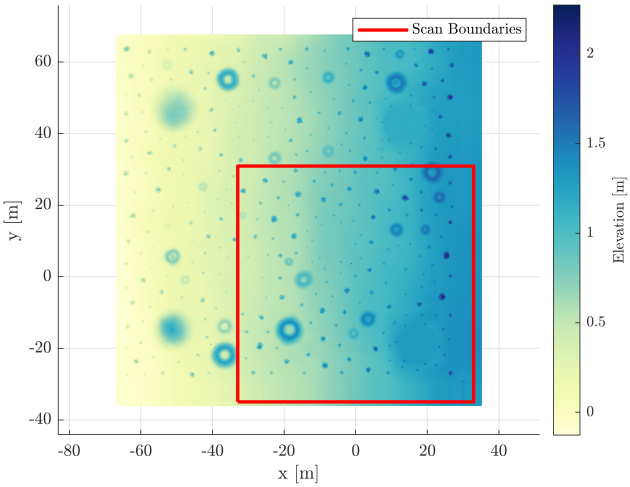


Fig. 16 A DEM of the man-made terrain. The red box outlines the portion which is scanned with lidar.

Truth data for the constructed terrain was gathered by tripod mounted scanning lidar. After data collection, a decision was made to add additional rock piles to certain portions of the terrain. These piles are present at the time of the test flights and are visible in our mapping system's output. A non-exhaustive list of their locations is found in

Table 1 A non-exhaustive list of rock piles missing from the truth data set.

x [m]	y [m]
-25	-1
-8	9
-4	11
13	6
15	-2
16	-11
15	-26

Table 1. All major missing piles are listed, but some minor piles which cannot be identified with certainty may exist. Furthermore, several scanning artifacts are found in the truth terrain which are approximately 1m tall and less than 0.25m in diameter. These artifacts are likely due to the lidar scanning its own support system and are not present in the flight test data. For these reasons, we choose to manually smooth them out of the truth data set.

B. Navigation System

Our system requires a navigation estimate to function. For this we use the LIDAR-Inertial Navigation system outlined by Setterfield *et. al.* [15]. This navigation uses an open source Incremental Smoothing and Mapping (iSAM) [21] implementation called Georgia Tech Smoothing and Mapping (GTSAM) [36]. Map relative localization (MRL) produces spacecraft pose measurements by comparing incoming lidar scans with an *a priori* known map prior. These are fused with inertial measurements in a sparse nonlinear least squares optimization scheme to produce spacecraft pose estimates. A flow chart outlining information transfer between sensors, the navigation system, the mapping system, and the LSS system is shown in Figure 17.

Most of the scanned terrain was observed only once by the lidar. Future missions may decide to scan promising portions of the terrain repeatedly and our system was designed to accommodate this type of mission profile. To demonstrate our effectiveness in both regimes we show results from three tests. The first simply use data from free flights 13 and 14 (FF13 and FF14). The third and final test concatenates data and navigation estimates from FF13 and FF14 to demonstrate improved performance when terrain is revisited.

C. Mapping Results

A key advantage to our quadtree system is its variable resolution. Figure 18 overlays the quadtree divisions upon the truth elevation map for FF13. Feature rich areas tend to exhibit a noticeably higher cell resolution than sparse ones.

The mapping system provides planar estimates of the terrain. An elevation estimate at a given point on the plane can be found by looking at its distance from the x-y plane in inertial space. The elevation estimates have maximum errors of 1.25 meters for FF14. High error regions are generally on the surface of the hazardous rock piles and arise from their

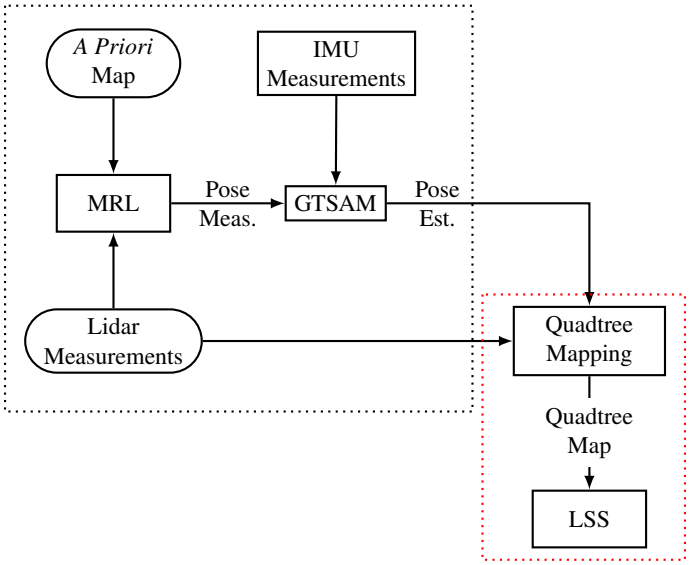


Fig. 17 A flow chart of the navigation system and our systems outlined in black and red dashes, respectively.

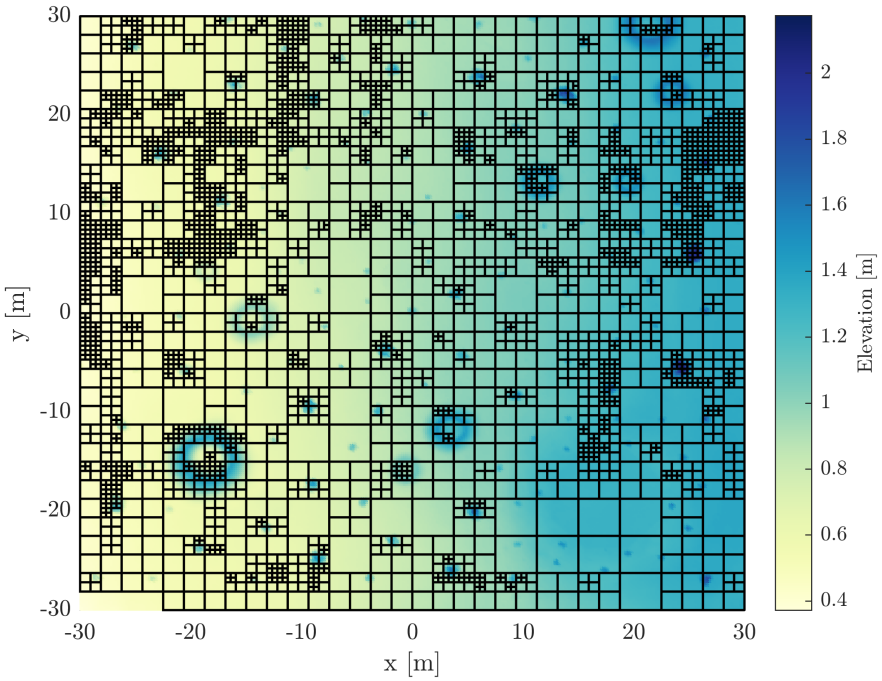


Fig. 18 The elevation truth alongside quadtree cell divisions for FF13.

highly non-planar structure. Many high error regions are a result of the missing features documented in Table 1. Figures 19 and 20, shows elevation estimates and error for FF14. Results for this test are representative of the other two. Red arrows highlight higher error at the missing rock piles.

An ideal characterization of map consistency would involve finding the estimate error with respect to a planar

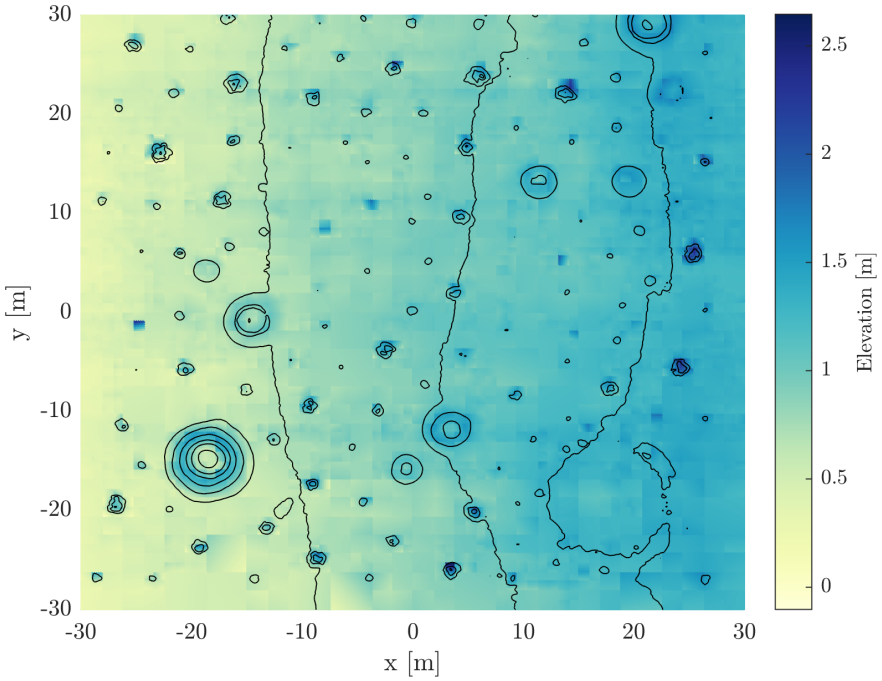


Fig. 19 The elevation estimates for FF14. Black contour lines represent the truth elevation.

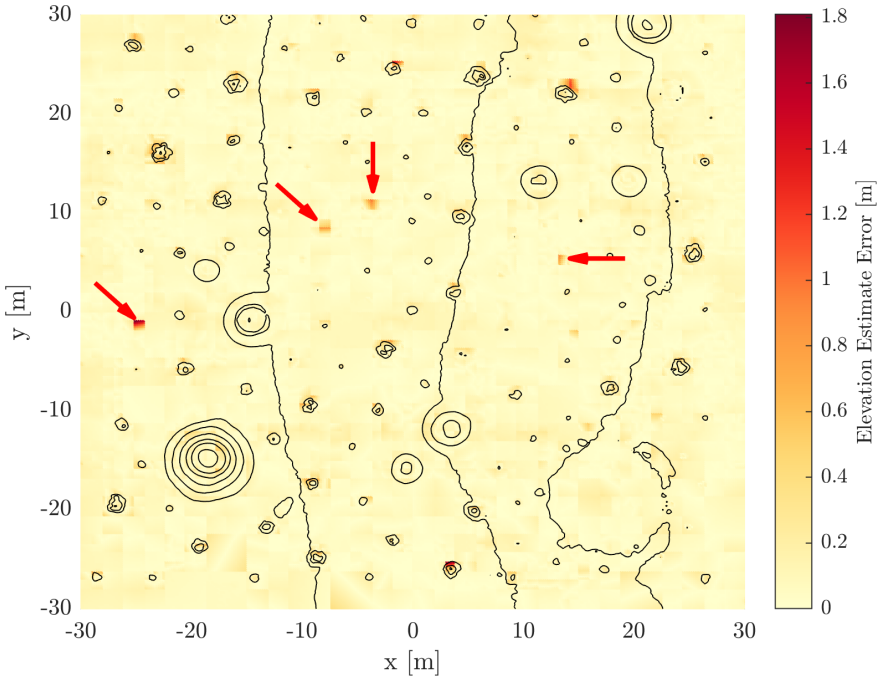


Fig. 20 The elevation estimate errors for FF14. Black contour lines represent the truth elevation.

approximation derived from the truth terrain. It is difficult to deduce meaningful statistics from this error due to the presence of missing map features in Table 1 and unknown truth mapping uncertainty. Therefore, we investigate prefit measurement residuals.

Table 2 The fraction of measurements below and above the Mahalanobis distance acceptance interval.

Flight	ρ_{r_1}	ρ_{r_2}	$\rho_{r_1} + \rho_{r_2}$
FF13	0.073	0.002	0.095
FF14	0.078	0.006	0.084
FF13 + FF14	0.072	0.004	0.076

Each measurement is converted to a point in inertial, Cartesian, space with Equation (15), $\hat{\boldsymbol{r}}^{\mathcal{I}}$. A linearized approximation of the point's covariance, $\boldsymbol{\Sigma}_{\boldsymbol{r}^{\mathcal{I}}}$, is available with Equation (17). Each point has been associated with a base level planar estimate, $\hat{\boldsymbol{P}}$, with estimated covariance $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{P}}$. A predicted measurement in inertial, Cartesian, space, $\bar{\boldsymbol{r}}^{\mathcal{I}}$, is found using the spacecraft's estimated pose, the measured azimuth and elevation, and $\hat{\boldsymbol{P}}$. After all lidar measurements have been processed, a postfit residual is found $\tilde{\boldsymbol{r}} = \hat{\boldsymbol{r}}^{\mathcal{I}} - \bar{\boldsymbol{r}}^{\mathcal{I}}$. A linearized approximation for the covariance of $\tilde{\boldsymbol{r}}$ is given by the following.

$$\boldsymbol{\Sigma}_{\tilde{\boldsymbol{r}}} = \frac{\partial \tilde{\boldsymbol{r}}}{\partial \boldsymbol{P}} \hat{\boldsymbol{\Sigma}}_{\boldsymbol{P}} \frac{\partial \tilde{\boldsymbol{r}}}{\partial \boldsymbol{P}}^{T} + \boldsymbol{\Sigma}_{\boldsymbol{r}^{\mathcal{I}}} \tag{61}$$

We find the Mahalanobis distance, (46), between each $\tilde{\boldsymbol{r}}$ and associated distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{r}}})$. The Mahalanobis distance is Chi distributed with 3 degrees of freedom when $\tilde{\boldsymbol{r}}$ is truly sampled from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{r}}})$. We analyze the consistency of our mapping scheme by investigating the degree to which the Mahalanobis distance of our measurement residuals meets the predicted Chi distribution. We specify a confidence level $\alpha = 0.05$ from which an acceptance region can be constructed. The region's upper and lower boundaries, r_1 and r_2 can be found with the inverse Chi cumulative distribution function, c_{χ}^{-1}.

$$r_1 = c_{\chi}^{-1}(\alpha/2) \tag{62}$$

$$r_2 = c_{\chi}^{-1}(1 - \alpha/2) \tag{63}$$

For each test flight we find the fraction of measurements above, ρ_{r_2}, and below, ρ_{r_1}, the acceptance interval. An estimator is called inefficient if its variance is larger than the Cramer-Rao Lower Bound [37]. Large fractions of distances below r_1 indicate an inefficient estimate. Large fractions above r_2 indicate inconsistency. If our estimates are consistent, we expect to see $\rho_{r_1} = \rho_{r_2} = \alpha/2$. Investigations revealed most estimates were generally inefficient. We believe this is due to the lidar performing better than the 8cm 1σ design specification. We reduce the predicted lidar uncertainty to 4cm 1σ to achieve more efficient estimates. The results for each test are outlined in Table 2. We see that in general, our estimates remain conservative, with a large portion of residuals being lower than expected. This is likely due to conservatively large spacecraft pose covariances as reported by the localization system.

D. Landing Site Selection Results

A critical step in the LSS process is hazard identification. Figure 21 shows the true safety map. This was found through a wholly independent process by simulating points of contact between vehicle landing pads and the truth DEM. This is in contrast to the simple circular landing sites produced by the LSS system. This discrepancy is caused by the desire for a high fidelity truth safety map and the anticipated computing constraints of an onboard LSS system. A point is marked unsafe if slope or roughness tolerances are exceeded as the vehicle rests on the truth DEM. In reality, a point's safety may depend on vehicle orientation. It is possible for a vehicle to come to a safe landing while its legs straddle a rock. We ignore these circumstances and mark a point as unsafe if there exists any single orientation which makes it such. As previously mentioned, hazards from Table 1 are not contained within the truth DEM. These hazards are approximately one meter in diameter and we assume they would violated vehicle safety constraints. We manually mark regions within one lander radius of these hazards as unsafe.

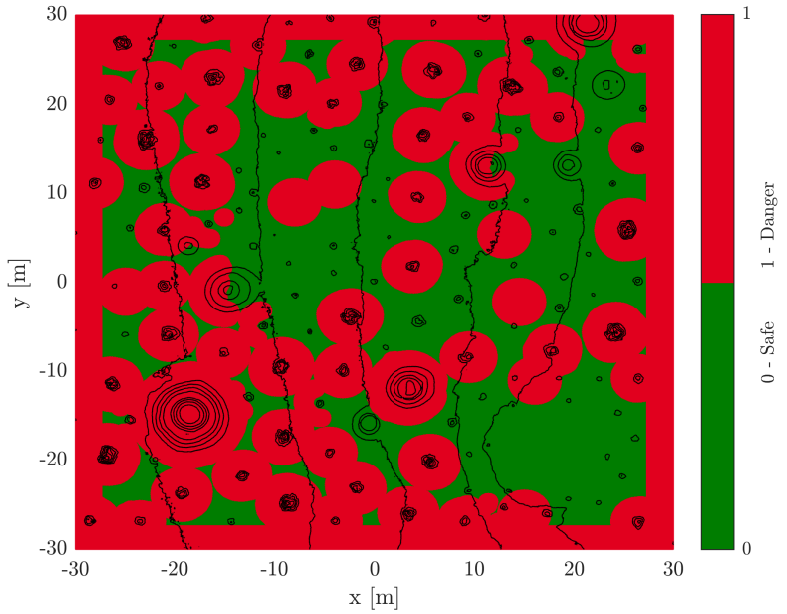


Fig. 21 The true safety map.

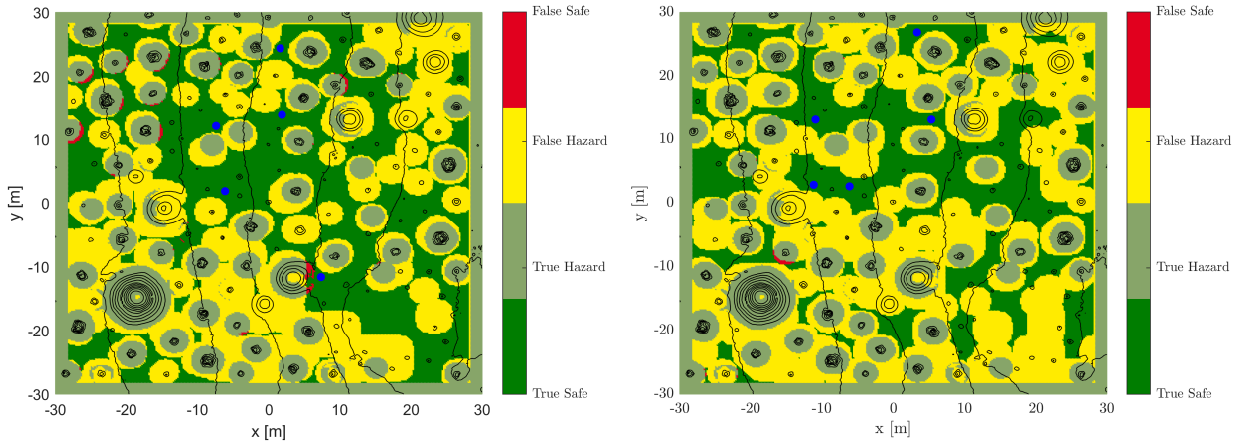
To quantify site selection performance we investigate the rate of successful hazard and safety detection. A perfect system would identify 100% of hazards and safe areas. In reality false positives are likely. We make the conservative choice of tuning our system to detect almost all hazardous terrain which comes at the expense of falsely classifying significant portions of safe terrain. Table 3 shows the detection accuracy for each flight test. We are able to detect the large majority of hazardous terrain while also providing ample true safe terrain detection for numerous landing sites to be selected.

Figure 22 shows the safety classification performance for FF13 and FF14. All hazards have been correctly identified

Table 3 The fraction of hazardous and safe terrain correctly identified for each flight.

Flight	Hazardous Terrain	Safe Terrain
FF13	99.09%	44.36%
FF14	99.84%	36.06%
FF13 + FF14	99.20%	35.11%

in FF13 but some hazards have been sized too small in resulting in false safe terrain at their edges. The vehicle is discouraged from landing in these areas by the hazard distance penalty in site scoring. All hazards are detected in FF14 and the system does a much better job at correctly estimating hazard extent. Also displayed are the top five scoring landing sites as blue dots. The dots mark the lander's center of mass at touchdown and do not reflect the vehicle's true footprint.



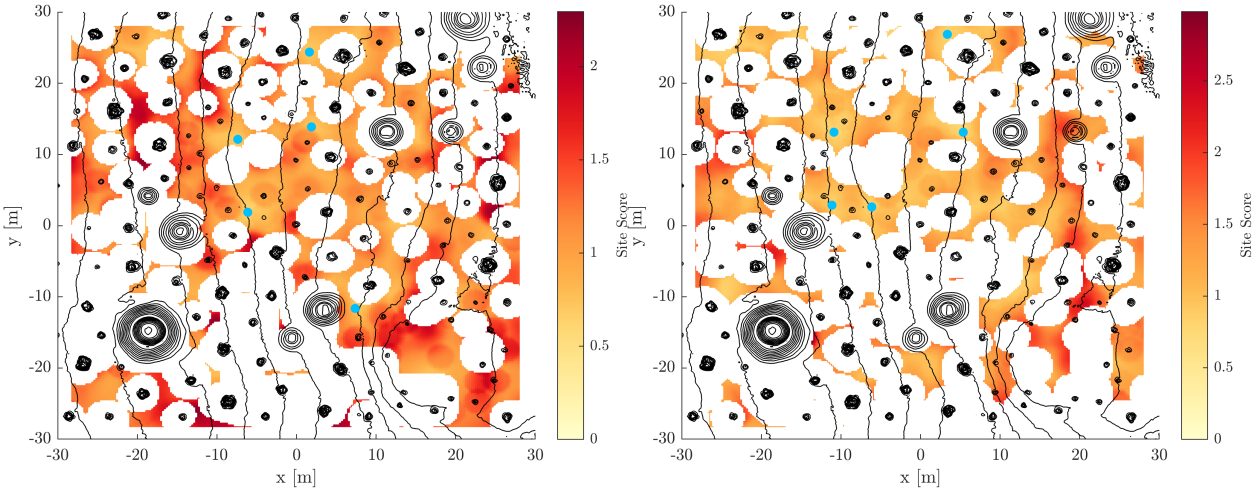
(a) **(b)**

Fig. 22 Site safety classification performance for FF13 (a) and FF14 (b).

Figure 23 shows the site score for terrain classified as safe in FF13 and FF14. The heightened site score around many small rocks shows that they have been identified as subscale features and are avoided by the site selection system. This behavior is highlighted in Figure 24. The top five scoring landing sites after lidar scans have been processed are shown as cyan dots. The dots mark the lander's center of mass at touchdown and do not reflect the vehicle's true footprint.

Figure 24 shows the site score for terrain classified as safe in FF13 + FF14. Several examples of subscale features are annotated with blue arrows. Successful subscale feature detection results in a heightened site score, allowing the LSS system to avoid the hazard. Note these subscale features are within acceptable landing site tolerances, but we prefer the flattest site possible. The top five scoring landing sites after lidar scans have been processed are shown as cyan dots. The dots mark the lander's center of mass at touchdown and do not reflect the vehicle's true footprint.

Figure 25 shows the selected sites for all flights. They are overlaid upon the binned and averaged measurement data



(a) FF13 **(b) FF14**

Fig. 23 Site scoring for FF13 (a) and FF14 (b). Contours represent the truth terrain.

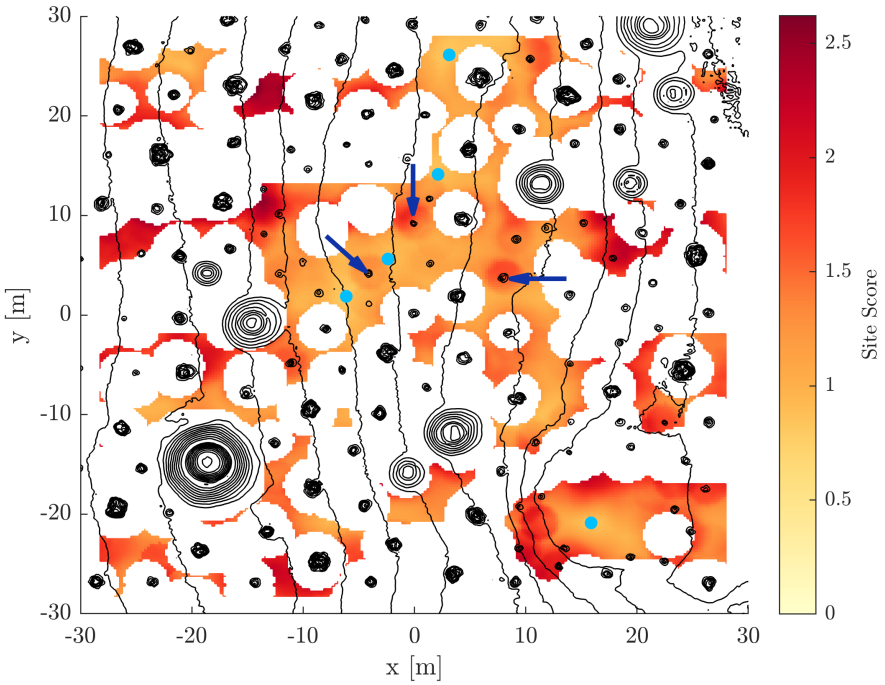


Fig. 24 Site scoring for FF13 + FF14. Contours represent the truth terrain.

for FF13 + FF14 to show that our system consistently returns site selections which avoids all major terrain features, even those which are not necessarily mission ending. Circles surrounding each site represent the vehicle's footprint and expected GNC uncertainty upon touchdown.

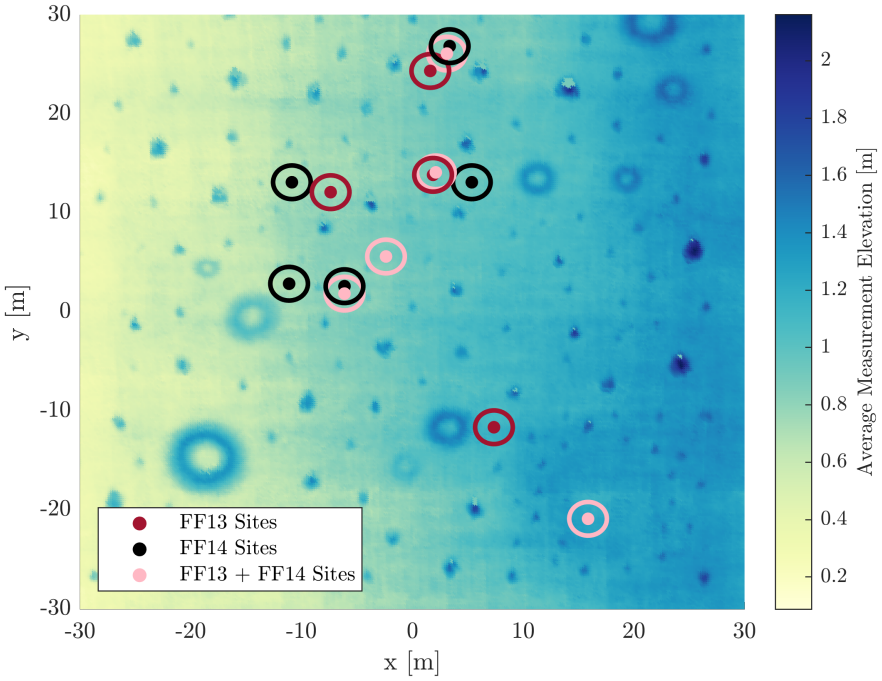


Fig. 25 The top five scoring landing sites from all flights with circles to represent the vehicle's footprint.

VIII. Conclusion

This work presents a novel combined mapping and landing site selection technique. The map is represented as a variable resolution quadtree in which each element contains a planar estimate of the terrain. Terrain slope is a vital consideration for landing site selection and can be easily accessed with the estimated surface normal vectors. Estimating slope directly simplifies and streamlines the landing site selection process. Landing sites are selected as a function of estimated slope, roughness, distance from existing hazards, and presence of subscale features. Our mapping and subscale feature estimation techniques are filter-based and completely recursive, allowing them to be updated quickly upon the arrival of new information.

Future work might refine our system further by including additional sources of information such as cameras or radar. It might also seek to validate our system with additional flight-test data or for real-time operation onboard an airborne vehicle.

Acknowledgments

Special thanks to Nikolas Trawny and the NASA Jet Propulsion Laboratory team for assistance with the Morpheus flight test data. This work was supported by a NASA Space Technology Graduate Research Opportunity, NASA grant number: 80NSSC20K1195.

References

[1] Nelessen, A., Sackier, C., Clark, I., Brugarolas, P., Villar, G., Chen, A., Stehura, A., Otero, R., Stilley, E., Way, D., Edquist, K., Mohan, S., Giovingo, C., and Lefland, M., "Mars 2020 Entry, Descent, and Landing System Overview," *2019 IEEE Aerospace Conference*, 2019, pp. 1–20. https://doi.org/10.1109/AERO.2019.8742167.

[2] "Autonomous Landing Hazard Avoidance Technology (ALHAT)," https://techport.nasa.gov/view/94147, 2016. [Online; accessed: October 6th, 2019].

[3] "Safe and Precise Landing – Integrated Capabilities Evolution (SPLICE)," https://www.nasa.gov/directorates/spacetech/game_changing_development/projects/SPLICE/about, 2020. [Online; accessed: June 16th, 2023].

[4] Shankar, U. J., Shyong, W.-J., Criss, T. B., and Adams, D., "Lunar terrain surface modeling for the ALHAT program," *2008 IEEE Aerospace Conference*, IEEE, 2008, pp. 1–10. https://doi.org/10.1109/AERO.2008.4526300.

[5] Trawny, N., Huertas, A., Luna, M. E., Villalpando, C. Y., Martin, K., Carson, J. M., Johnson, A. E., Restrepo, C., and Roback, V. E., "Flight testing a real-time hazard detection system for safe lunar landing on the rocket-powered morpheus vehicle," *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 0326. https://doi.org/10.2514/6.2015-0326.

[6] Finkel, R., and Bentley, J., "Quad Trees: A Data Structure for Retrieval on Composite Keys." *Acta Inf.*, Vol. 4, 1974, pp. 1–9. https://doi.org/10.1007/BF00288933.

[7] Serrano, N., "A Bayesian framework for landing site selection during autonomous spacecraft descent," *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 5112–5117. https://doi.org/10.1109/IROS.2006.282603.

[8] Cohanim, B. E., and Collins, B. K., "Landing point designation algorithm for lunar landing," *Journal of Spacecraft and Rockets*, Vol. 46, No. 4, 2009, pp. 858–864. https://doi.org/10.2514/1.42002.

[9] Cui, P., Ge, D., and Gao, A., "Optimal landing site selection based on safety index during planetary descent," *Acta Astronautica*, Vol. 132, No. October 2016, 2017, pp. 326–336. https://doi.org/10.1016/j.actaastro.2016.10.040.

[10] Luna, M. E., Almeida, E., Spiers, G., Villalpando, C. Y., Johnson, A. E., and Trawny, N., "Evaluation of the simple safe site selection (S4) hazard detection algorithm using helicopter field test data," *AIAA Guidance, Navigation, and Control Conference, 2017*, , No. January, 2017, pp. 1–14. https://doi.org/10.2514/6.2017-1497.

[11] Mango, D., Opromolla, R., and Schmitt, C., "Hazard detection and landing site selection for planetary exploration using LIDAR," *2020 IEEE International Workshop on Metrology for AeroSpace, MetroAeroSpace 2020 - Proceedings*, 2020, pp. 392–397. https://doi.org/10.1109/MetroAeroSpace48742.2020.9160010.

[12] Marcus, C. L., Setterfield, T. P., Hewitt, R. A., and Chen, P.-T., "Landing Site Selection with a Variable-Resolution SLAM-Refined Map," *2022 IEEE Aerospace Conference (AERO)*, IEEE, 2022, pp. 1–12. https://doi.org/10.1109/AERO53065.2022.9843602.

[13] Hertzberg, C., Wagner, R., Frese, U., and Schröder, L., "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, Vol. 14, No. 1, 2013, pp. 57–77. https://doi.org/10.1016/j.inffus.2011.08.003.

[14] Marcus, C., Setterfield, T., and Zanetti, R., "Variable Resolution Quadtree Mapping for Planetary Landing Using Planar Elements," *Rocky Mountain AAS GN&C Conference*, 2022.

[15] Setterfield, T. P., Hewitt, R. A., Chen, P.-T., Espinoza, A. T., Trawny, N., and Katake, A., "LiDAR-Inertial Based Navigation and Mapping for Precision Landing," *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–19. https://doi.org/10.1109/AERO50100.2021.9438153.

[16] Viejo, D., and Cazorla, M., "3D plane-based egomotion for SLAM on semi-struetured environment," *IEEE International Conference on Intelligent Robots and Systems*, 2007, pp. 2761–2766. https://doi.org/10.1109/IROS.2007.4399138.

[17] Rusinkiewicz, S., and Levoy, M., "Efficient variants of the ICP algorithm," *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*, 2001, pp. 145–152. https://doi.org/10.1109/IM.2001.924423.

[18] Zureiki, A., and Devy, M., *SLAM and data fusion from visual landmarks and 3D planes*, Vol. 41, IFAC, 2008. https://doi.org/10.3182/20080706-5-kr-1001.02481, URL http://dx.doi.org/10.3182/20080706-5-KR-1001.02481.

[19] Servant, F., Houlier, P., and Marchand, E., "Improving monocular plane-based SLAM with inertial measures," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 3810–3815. https://doi.org/10.1109/IROS.2010.5648977.

[20] Grant, W. S., Voorhies, R. C., and Itti, L., "Efficient Velodyne SLAM with point and plane features," *Autonomous Robots*, Vol. 43, No. 5, 2019, pp. 1207–1224. https://doi.org/10.1007/s10514-018-9794-6.

[21] Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F., "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research*, Vol. 31, No. 2, 2012, pp. 216–235. https://doi.org/10.1177/0278364911430419.

[22] Gallant, M. J., and Marshall, J. A., "The LiDAR compass: Extremely lightweight heading estimation with axis maps," *Robotics and Autonomous Systems*, Vol. 82, 2016, pp. 35–45. https://doi.org/10.1016/j.robot.2016.04.005.

[23] Johnson, A. E., Klumpp, A. R., Collier, J. B., and Wolf, A. A., "Lidar-based hazard avoidance for safe landing on Mars," *Journal of guidance, control, and dynamics*, Vol. 25, No. 6, 2002, pp. 1091–1099. https://doi.org/10.2514/2.4988.

[24] Serrano, N., and Seraji, H., "Landing site selection using fuzzy rule-based reasoning," *Proceedings - IEEE International Conference on Robotics and Automation*, , No. April, 2007, pp. 4899–4904. https://doi.org/10.1109/ROBOT.2007.364234.

[25] Moghe, R., and Zanetti, R., "On-line hazard detection algorithm for precision lunar landing using semantic segmentation," *AIAA Scitech 2020 Forum*, 2020, p. 0462. https://doi.org/10.2514/6.2020-0462.

[26] Ivanov, T., Huertas, A., and Carson, J. M., "Probabilistic hazard detection for autonomous safe landing," *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, pp. 1–13. https://doi.org/10.2514/6.2013-5019.

[27] Johnson, A., and Mandalia, A., "Simple safe site selection: Hazard avoidance algorithm performance at mars," *Proc. 37th AAS Rocky Mountain Guidance and Control Conference. AAS*, 2014.

[28] Jung, Y., Lee, S., and Bang, H., "Digital Terrain Map Based Safe Landing Site Selection for Planetary Landing," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 1, 2020, pp. 368–380. https://doi.org/10.1109/TAES.2019.2913600.

[29] Yang, T., Li, P., Zhang, H., Li, J., and Li, Z., "Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments," *Electronics (Switzerland)*, Vol. 7, No. 5, 2018. https://doi.org/10.3390/electronics7050073.

[30] Schoppmann, P., Proença, P. F., Delaune, J., Pantic, M., Hinzmann, T., Matthies, L., Siegwart, R., and Brockers, R., "Multi-resolution elevation mapping and safe landing site detection with applications to planetary rotorcraft," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 1990–1997. https://doi.org/10.1109/IROS51168.2021.9636507.

[31] Li, S., Jiang, X., and Tao, T., "Guidance summary and assessment of the Chang'e-3 powered descent and landing," *Journal of Spacecraft and Rockets*, Vol. 53, No. 2, 2016, pp. 258–277. https://doi.org/10.2514/1.A33208.

[32] Huang, X., Li, M., Wang, X., Hu, J., Zhao, Y., Guo, M., Xu, C., Liu, W., Wang, Y., Hao, C., et al., "The Tianwen-1 guidance, navigation, and control for Mars entry, descent, and landing," *Space: Science & Technology*, Vol. 2021, 2021. https://doi.org/10.34133/2021/9846185.

[33] Gallant, M. J., "Axis mapping: the estimation of surface orientations and its applications in vehicle localization and structural geology," Ph.D. thesis, Queen's University (Canada), 2016.

[34] Hewitt, R. A., "Intense navigation: Using active sensor intensity observations to improve localization and mapping," Ph.D. thesis, Queen's University (Canada), 2018.

[35] Sola, J., Deray, J., and Atchuthan, D., "A micro Lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018. https://doi.org/10.48550/arXiv.1812.01537.

[36] Dellaert, F., and Contributors, G., "borglab/gtsam," , May 2022. https://doi.org/10.5281/zenodo.5794541, URL https://github.com/borglab/gtsam).

[37] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T., *Estimation with applications to tracking and navigation: theory algorithms and software*, John Wiley & Sons, 2004. https://doi.org/10.1002/0471221279.