Landing Site Selection with a Variable-Resolution SLAM-Refined Map

Corey L. Marcus
The University of Texas at Austin
Department of Aerospace Engineering
2617 Wichita St, Austin, TX 78712
cmarcus@utexas.edu

Timothy P. Setterfield,
Robert A. Hewitt,
Po-Ting Chen
NASA Jet Propulsion Laboratory
California Intitute of Technology
4800 Oak Grove Dr, Pasadena, CA 91109
Timothy.P.Setterfield@jpl.nasa.gov

***Abstract*—In many scenarios it is desirable for planetary landers to select or modify their landing sites autonomously during descent. We present a landing site selection algorithm which is optimized to work in conjunction with a Simultaneous Localization and Mapping system. Our algorithm selects landing sites based on site slope, roughness, and operator-defined interest. In addition, we generate guidance commands and approximate fuel consumption for the highest ranked sites. We validate our algorithm with LiDAR and inertial data gathered by a vertical take-off and landing vehicle.**

TABLE OF CONTENTS

1. INTRODUCTION

Many of the solar system's most scientifically interesting landing sites feature hazards which are poorly characterized or which are dynamic on the time-scale of a typical interplanetary campaign. Environmental factors such as the clouds of Venus, shadows of the Lunar south pole, and potentially dynamic crevasses of Europa and Enceladus are coupled with the high monetary cost and relatively low resolution of orbital imagery to make *a priori* hazard map generation difficult. For these reasons, it is desirable for planetary landers to select their landing site in real-time during descent.

Such a site would be selected from a map generated through a process such as Simultaneous Localization and Mapping (SLAM). SLAM can be used to produce dynamic maps of variable resolution, which evolve as more measurements are taken. Existing landing site selection technologies select sites from static maps which are typically of fixed resolution. In this work we demonstrate a method for picking landing sites from a dynamic and variable resolution map with an emphasis on limiting computational complexity and creating a path towards real-time performance.

We primarily consider a lander using a Light Detection and Ranging (LiDAR) measurement system to map the surface during descent. We assume this LiDAR system takes range measurements in a fixed field of view at constant angular increments. As altitude decreases, the lateral distance between points measured on the terrain shrinks. Thus, it is natural for portions of the terrain mapped at lower altitudes to be represented by a higher resolution map. The quadtree data structure is a variable resolution grid which is well suited to handling this variable resolution mapping problem. We leverage the quadtree heavily in this work.

We use a LiDAR inertial SLAM system to produce a localization and mapping solution. Our landing site selection (LSS) algorithm selects landing sites given inputs of the estimated vehicle pose, LiDAR measurements, and estimated digital elevation map. The SLAM map is stored in a quadtree due to the ease with which it represents map areas with disparate measurement densities. The landing site selected is a function of the terrain's slope, roughness, and *a priori* determined interest. Fuel consumption requirements for top performing sites are found. Our system is validated with real LiDAR and inertial measurements gathered by a rocket powered vertical take-off and landing system.

The remainder of this paper is organized as follows, Section 2 discusses related work, Section 3 provides an overview of the SLAM system used, Section 4 documents the specifics of our algorithm, Section 5 shows how our system performs with real data, and Section 6 offers some conclusions.

2. RELATED WORK

LSS has a relatively rich treatment in the literature. Johnson *et al* document a landing site selection system based on a scanning LiDAR [1]. Their system uses a plane fitting technique to estimate surface slope and roughness. They work directly with the LiDAR-created point cloud and assume that vehicle position is known. Once a site has been chosen, they use a guidance module to generate a trajectory towards the site. Their algorithm is validated in simulation.

Serrano provides another treatment of autonomous landing site selection [2]. He is able to probabilistically model the problem using a Bayesian Network given radar, camera, and LiDAR measurements. Terrain slope and roughness are characterized by fitting planes to small subsets of the map. Craters are located in camera images via edge and shadow detection and matched with an *a priori* map. Rocks are modeled from shadow detection and knowledge of the sun angle. Knowledge of the spacecraft's performance capabilities and current terrain is used to bound reachable and marginally reachable terrain. Finally, he also considers a scientific interest map which has been defined before flight.

978-1-6654-3760-8/22/$31.00 ©2022 IEEE

This map allows mission designers to bias the algorithm towards or away from certain portions of the map according to their scientific value. These factors are all combined to provide a landing site value to each map point.

This process is computationally intensive, so Serrano uses a supervised learning approach to learn to assign landing site values given a digital elevation map (DEM), hazard map (craters and rocks), and scientifically interesting map. Later, Serrano presents a follow-up work which presents essentially identical results while using fuzzy logic in place of supervised learning [3].

Some of NASA's most recent investigations into LSS fall under the Autonomous Landing Hazard Avoidance Technology (ALHAT) and Safe and Precise Landing – Integrated Capabilities Evolution (SPLICE) programs. Cohanim and Collins describe the ALHAT approach to LSS [4]. Multiple potential landing sites are selected according to their distance to the nearest hazard, approximate fuel consumption, and distance to the nearest point of scientific interest. These metrics are provided as inputs to the selection algorithm and their creation is not explicitly explored. One interesting feature of the algorithm is the ability to control the distance between the returned landing sites. The authors model the vehicle's footprint as a circle as opposed to modeling footpads and engine bells explicitly. They argue that such modeling may result in dangerous spacecraft maneuvers and that attempting to straddle a hazard is not a worthwhile endeavor.

Another investigation under the ALHAT program by Ivanov and Carson led to a high fidelity probabilistic approach to LSS [5]. They probabilistically model the likelihood of safe landings under many different landing orientations while evaluating sites. The expected navigation uncertainty is also used to account for the fact that the spacecraft may inadvertently land on nearby and unsafe terrain.

Cui *et al* consider many safety factors while selecting the optimal landing site [6]. They use the classic plane fitting techniques to quantify slope and roughness. They use a polynomial guidance law to find fuel consumption to different points on the map. They also explicitly consider touchdown performance through simulation with execution error and a constant wind disturbance. They conduct a traversal search with a bubble sort to locate the lowest cost landing site.

Luna *et al* demonstrate the effectiveness of the Simple Safe Site Selection (S4) algorithm developed by Johnson and Mandalia [7], [8]. They show results similar in quality to the algorithms developed through ALHAT. S4's design is motivated by computational simplicity. Using a single flash LiDAR image, it identifies the slope, roughness, and rock locations. These are used to output a landing safety map. No other parameters, such as fuel consumption, are considered.

Mango, Opromolla, and Schmitt demonstrate another LiDAR-based algorithm [9]. They use a much larger DEM which has been generated from multiple LiDAR images. They also develop a technique for searching first at a course resolution to narrow the space for fine resolution searching.

In a relatively unique approach, Jung *et al* fit constant elevation contour lines to a triangular Digital Terrain Map (DTM) [10]. These lines are used to locate rocks and slope. The authors identify numerous difficulties which must be overcome with their method. Most stem from the choice in step between contours. It is possible for hazards to become lost between the contours. To avoid this and other issues, many conditionals must be evaluated, and the algorithm's decision tree becomes relatively complex.

Practically, many of the maps which are inputs for landing site selection would come from a combination of offline and online mapping. Yang *et al* consider this explicitly by bundling LSS with Simultaneous Localization and Mapping (SLAM) [11]. They take a 3D point cloud and convert it to a 2D DEM. A landing site is selected as a region of the DEM which is sufficiently flat. This procedure is all performed online on a UAV. The difference in platform means that many needs of spacecraft such as fuel consumption are not considered.

Tomita *et al* use Semantic Segmentation to locate safe landing sites [12]. This is a convolutional neural network based approach. A model is trained given DEM inputs and hazard map outputs. They show that their trained model can outperform ALHAT's system when the DEM is sufficiently noisy.

Schoppmann *et al* detail a multi resolution landing site detection algorithm for unmanned aerial vehicles [13]. This system selects landing sites from a multi-resolution map generated from an on-board structure-from-motion program. The map is organized into a Laplacian pyramid which divides terrain features into layers based on their frequency content. Their landing site selection system considers slope, roughness, and mapping quality.

The LSS implementation of Trawny *et al* demonstrates real-time performance on a flight vehicle [14]. As part of the ALHAT program, they fly a rocket powered vertical take-off and landing vehicle known as Morpheus. Morpheus is flown over a man-made terrain containing craters, boulders, and other features which are reminiscent of a Lunar landscape. Morpheus uses a gimballed flash LiDAR system to scan the terrain. After scanning, a landing site is selected and the vehicle autonomously touches down. We use the same LiDAR measurements to validate our system in this work.

The Chang'E 3 mission to the moon employed a hazard detection and avoidance scheme using both camera images and LiDAR measurements [15]. Camera images and LiDAR measurements provide independent assessments of hazard locations. To detect hazards with LiDAR, a least squares method is used to fit a ground plane to their LiDAR measurements. They then search the ground plane in a spiral pattern, evaluating sites as a function of their fuel requirements and the safety indicated by the LiDAR measurements.

Many Martian landers have employed some level of hazard detection and avoidance. Most recently, the Mars 2020 (Perseverance) rover used terrain relative navigation to select the best landing site in a predefined hazard map [16]. Camera images allowed the lander to localize itself with a predefined hazard map. Then the safest site could be chosen during descent given the spacecraft's estimated position and maneuvering constraints. The Tianwen-1 lander used a combination of camera and LiDAR images to define hazards and select a safe landing site during a hover mode [17]. After hover-mode, the lander entered a hazard relative navigation phase where it avoided hazards and maneuvered to the safe landing site.

3. SIMULTANEOUS LOCALIZATION AND MAPPING

The landing site selection algorithm presented in this work receives its map and localization data from a SLAM system. For this we use the LiDAR-inertial SLAM system developed by Setterfield *et al* [18]. One critical component of this system is its map representation, a quadtree. The quadtree can be thought of as a variable resolution DEM. The system is initialized with some *a priori* DEM, and given new information through LiDAR measurements the estimated elevation of each cell may be refined. As the measurements in a given cell become denser, the cell may be divided into four subdivisions, each with their own elevation estimates. This process can be repeated until the cells achieve some predefined maximum resolution. An example quadtree is shown in Figure 1.

When the SLAM system receives a new LiDAR measurement of the surface, it associates that measurement with a quadtree cell. The cell's elevation estimate is found as the average elevation of all measurements it contains. When the number of measurements within a cell reaches a predefined threshold, it is divided into four children. During division, the measurements are assigned to one of the four children. The divided cell inherits an elevation estimate which is the average of its four children's elevation estimates. This methodology allows cells at any level in the tree to contain elevation estimates which are the average of all measurements the cells below them contain.

While the option exists to form the SLAM map from a fusion between the *a priori* DEM and LiDAR measurements, we form the map from LiDAR measurements alone. This choice was made for several reasons: it allows us to ensure that the LSS map is uncorrelated from the truth map; some terrain modifications were made after the measurement of the truth map; and it demonstrates that the system is not reliant on the *a priori* map for landing site selection (apart for its role in map relative localization).

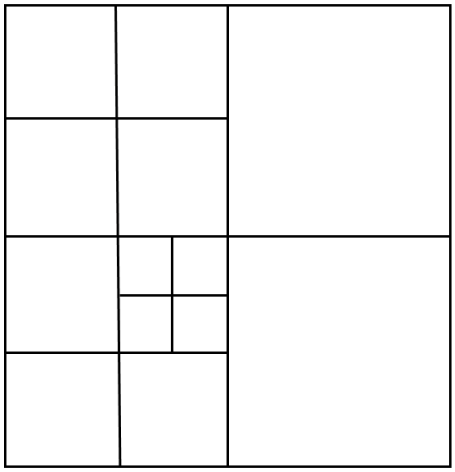


Figure 1: An example quadtree structure demonstrating three levels of depth.

Another critical feature of the SLAM system is its use of map relative localization (MRL). Most SLAM systems create maps which are defined with respect to the sensor as they lack pose measurements with respect to an inertial coordinate system. This SLAM system uses MRL to match LiDAR measurements with an *a priori* defined map of the terrain. Because the *a priori* map is defined with respect to the planet-fixed coordinate system, MRL measurements provide measurements of the vehicle's pose with respect to that same planet-fixed coordinate system. This is important for LSS because mission operators often have some knowledge of potential landing sites before flight. They may wish to bias the selection system towards known scientific interests or away from known hazards which are specified in the planet-fixed coordinate system. The SLAM system's pose estimates with respect to the planet-fixed coordinate system enable this biasing.

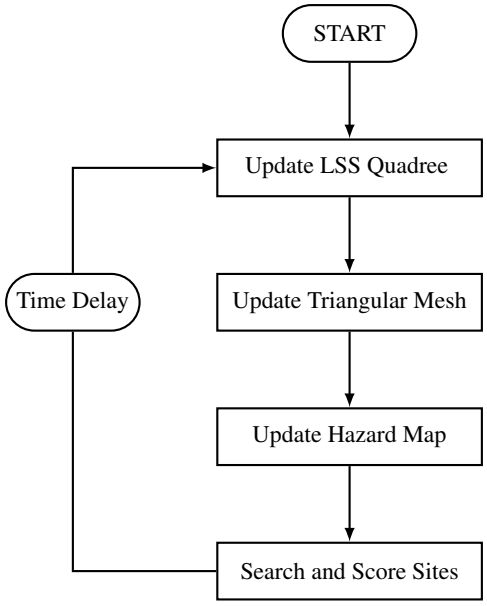


Figure 2: A block diagram of the major LSS operations.

4. APPROACH

Figure 2 provides an overview of the entire LSS algorithm. The system is divided into four primary operations: quadtree update, triangular mesh update, hazard map update, and site searching/scoring.

Our system relies heavily on the quadtree map structure for efficient map searches and data queries. The quadtree update operation maintains a quadtree structure in memory which mirrors the structure of the SLAM map. Independence is maintained between the two structures in order to ease future parallelization efforts. The LSS system runs periodically during flight at a user-defined interval and operates on segments of the map which are new or have been updated since the last LSS iteration. This reduces computational complexity and allows for an incremental LSS solution as the SLAM map is refined with new measurements.

Many LSS operations require searching the quadtree for the smallest available cells within a bandwidth tolerance. An example of cells fitting this criteria is shown in Figure 3. This search operation provides the system additional flexibility when searching maps with varying levels of resolution.

LSS operations discussed in this paper occur at different

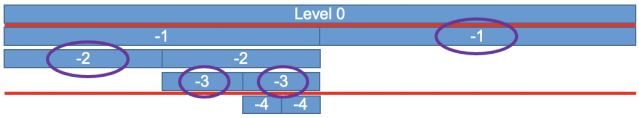


Figure 3: An example bandwidth tolerance. A 2D cross-section of a quadtree with levels 0 to -4 is shown. Red horizontal lines indicate the upper and lower bounds of the bandwidth tolerance. The smallest available cells within bandwidth tolerances are circled and represent the output of a bandwidth search.

levels within the quadtree. Some example LSS operations are shown at representative levels in Figure 4. The specifics of the operations shown will be discussed in a following section. These operations confined to one level in the Figure, but in actuality the levels at which they can occur are defined by bandwidth searches.

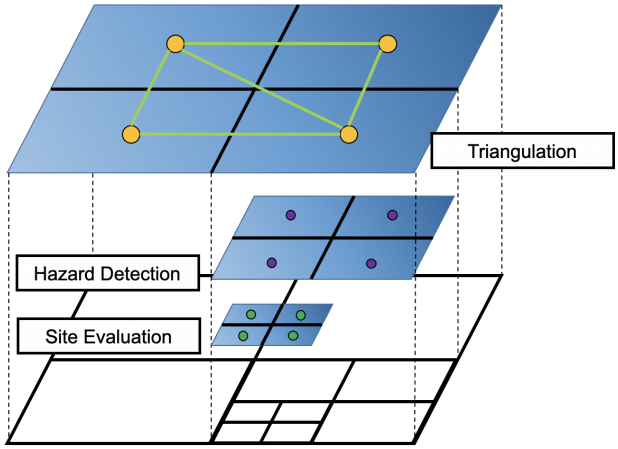


Figure 4: An example of the relative levels at which different operations are performed. In this figure, the triangular mesh is populated at the highest level, followed by the hazard detection and site evaluation operations.

Two physical parameters drive landing site evaluation, slope and roughness. We assume that mission designers have provided slope, λ_θ, and roughness, λ_d, tolerances which a landing zone must meet. A landing zone radius, $l_\mathcal{Z}$, is also provided which accounts for the lander's physical size and any expected differences between the targeted and actual landing site due to guidance navigation and control uncertainties. The site is considered unsafe when the angle, θ, between the local terrain's surface normal, $\hat{n}$, and the local gravity vector, g, exceeds λ_θ. A site is also considered unsafe if the distance between any portion of the terrain and its planar approximation exceeds λ_d within a square landing zone, $\mathcal{Z}$. The zone's half-side length, $l_\mathcal{Z}$, is chosen as a function of the vehicle's size and expected guidance, navigation, and control uncertainty. Larger vehicles, or those with less precise landing site targeting, require larger landing zones. A square zone is chosen over the traditional ellipse because it is more efficient to search a quadtree in a rectangular domain than a circular one. We assert that $l_\mathcal{Z}$ can always be chosen such that the entire ellipsoidal landing zone is contained within $\mathcal{Z}$. These slope and roughness tolerances are diagrammed in Figure 5.

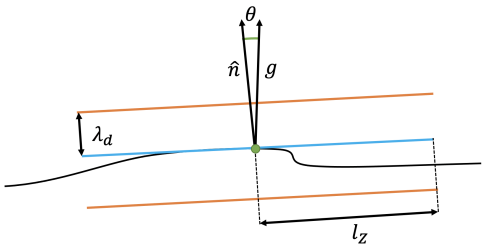


Figure 5: A diagram of the slope, θ, and roughness parameters in 2D. A landing site is represented with the green dot on the black terrain. The blue line represents a planar estimate of the local terrain. The landing zone half width, $\lambda_\mathcal{Z}$, defines the plane's domain. The roughness tolerance, λ_d has been used to define two orange planes. The angle between the local surface normal, $\hat{n}$, and the gravity vector, g, defines slope. If $\theta > \lambda_\theta$ or the truth terrain crosses either orange plane the site would be considered unsafe.

Triangular Mesh

One operation utilizing the bandwidth searches, shown in Figure 3, is the triangular mesh update. Cells within the bandwidth constraints become nodes within a 2.5D triangular mesh. The x and y locations of these nodes are deterministically controlled by the quadtree division structure but the z location is the cell's elevation estimate. Figure 6 shows an example of quadtree cells which have become nodes in a triangular mesh. The Delaunay triangulation method is used to generate triangles from the nodes as it minimizes the occurrence of thin triangles which span large distances in the x-y plane.

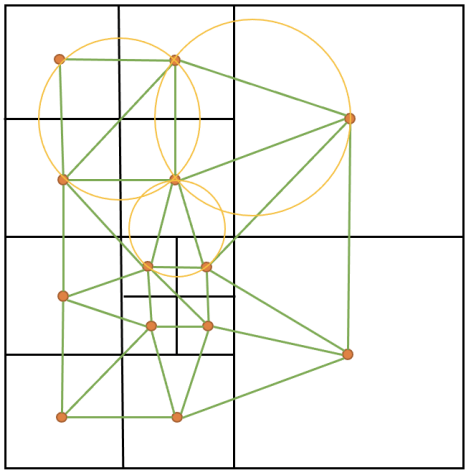


Figure 6: An example quadtree with cells as vertices in a triangular mesh. Several circles illustrate the primary constraint of a Delaunay Triangulation. Any circle defined by the vertices of one triangle may not encompass another vertex; however, intersection is permitted.

The mesh is formed from cells at a low enough resolution that the triangle might contain several potential landing sites. Because the triangles cover a large area relative to the size of the lander, they form good approximations of the local surface absent small-scale features. The vertices of each triangle define a plane which is used as an estimate of the local ground plane. This plane and its surface normal are used in several downstream LSS operations.

Planes, $\mathcal{P}$, in this work are defined by their surface normal, $\hat{n}$, and orthogonal distance from the origin, c. Any point, $p \in \hat{P}$ satisfies Equation (1).

$$p^T \hat{n} - c = 0 \tag{1}$$

We often desire the surface normal at the center of a quadtree cell for downstream slope and roughness calculations. This cell may or may not form a vertex in the triangular mesh. Its center is the 3D point $r = [x, y, \hat{z}]^T$ composed of a deterministic x-y position and estimated elevation, $\hat{z}$. The mesh is queried to find the triangle which contains the x-y coordinates of r. Each triangle has three vertices b, q, $s \in \mathbb{R}^3$ which correspond to cells in the quadtree. The z component of these vertices is the estimated elevation of each cell. The local surface normal, $\hat{n}$ is found through Equation (2) which uses the cross-product operator, $\times$.

$$\hat{n} = \frac{(q-b) \times (s-b)}{\| (q-b) \times (s-b) \|} \tag{2}$$

The orthogonal distance to the origin, c, is found with Equation (3).

$$c = r^T \hat{n} \tag{3}$$

The special case in which the surface normal is desired at boundary or vertex between multiple triangles is handled by finding the surface normal for each triangle. The surface normal which has the greatest angle with the local gravity vector is saved and the others are discarded. Accepting the worst-case surface serves as a conservative approach to handling this special case.

Hazard Identification

The LSS system also searches the quadtree to identify cells which contain hazardous terrain. These target cells are identified with another bandwidth search. The LiDAR measurements, $p_i \in \mathbb{R}^3$, corresponding to the the cell are queried from the SLAM quadtree. Note that the SLAM system has already transformed range and bearing LiDAR measurements into points in $\mathbb{R}^3$. We require that the measurement density (in measurements per m^2) exceed a predefined tolerance, λ_ρ. When the measurement density is insufficient, we choose the conservative option of marking the cell hazardous.

If measurement density tolerances are met, the cell is checked for the presence of hazards. We begin by forming the blue plane shown in Figure 5 as an approximation of the terrain at the center of the cell. The triangular mesh is used to produce a surface normal, $\hat{n}$, and distance from the origin, d, as shown in Equations (2) and (3). When the cell lies on the intersection of two triangles, the surface normal which makes the greater angle with the local gravity vector is taken.

Two planes are formed using the vehicle's roughness tolerance, λ_d, as offsets in either direction orthogonal to the plane defined by $\hat{n}$ and d. These are defined as $\mathcal{P}_1$ and $\mathcal{P}_2$ in Equations (4) and (5). We term points which are not between the two planes *exceptions*. The fraction of measurements which are exceptions is compared to the exception tolerance, λ_e. If the fraction exceeds the tolerance, we mark the cell hazardous. We find λ_e heuristically to minimize missed hazard detections while keeping false positives to a reasonable number.

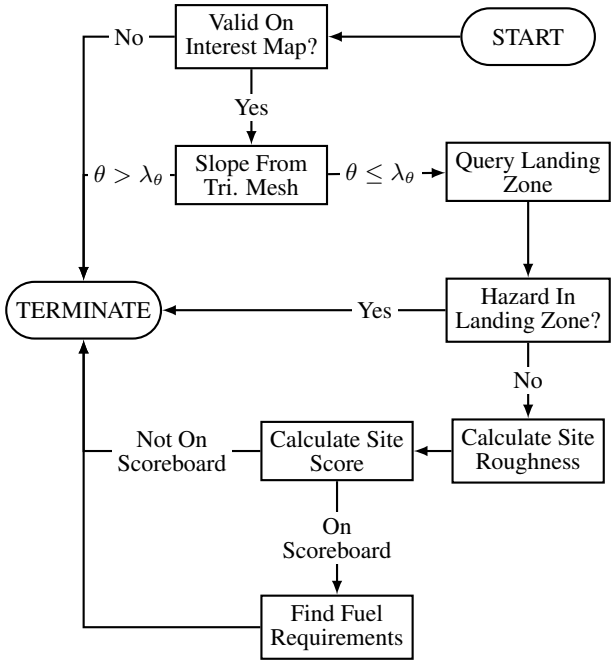


Figure 7: A block diagram of the operations used for scoring landing sites.

$$\mathcal{P}_1 = \left\{ r \in \mathbb{R}^3 \,|\, r^T \hat{n} - c - \lambda_d = 0 \right\} \tag{4}$$

$$\mathcal{P}_2 = \left\{ r \in \mathbb{R}^3 \,|\, r^T \hat{n} - c + \lambda_d = 0 \right\} \tag{5}$$

This hazard criteria considers only the number of exceptions, and not the amount of deviation from the plane fit to the center of the cell. The roughness metric discussed later in this work considers explicitly the magnitude of deviations from the plane fit of all measurements. This hazard identification process serves as a check to quickly eliminate portions of the terrain which appear hazardous.

Site Evaluation

The final major operation is to find and score potential landing sites. We note that landing sites which are very close to each other tend to have similar attributes. In addition, guidance, navigation, and control uncertainties may preclude targeting sites with high precision. For these reasons, a bandwidth search is used to find quadtree cells which will be considered for landing. The lower bound on the bandwidth has the dual effect of placing a lower bound on the distance between potential sites and limiting computational complexity. The scoring operation is diagrammed in Figure 7. The scoring elements are computed roughly in order of computational complexity. A variety of logic checks are used to terminate the scoring process should any intermediate metric fall outside predetermined tolerances. These checks avoid unnecessary computational expense.

Interest Score

The first check of the scoring algorithm is to reference the interest map. The interest map is a fixed resolution discretization of the x-y map domain where each discretized cell contains an interest score, $I(x, y) \in [0, 1]$. These interest scores have been determined prior to flight according to factors such as scientific interest or known hazards. They could even reflect more abstract requirements such as expected sky visibility for orbital communications in mountainous terrain. A higher score indicates a more desirable location. A score of zero indicates a cell which cannot be selected for landing. In this case, the score evaluation is terminated. We label the interest score of the j^{th} landing site as I^j.

Slope Score

The site slope is using the site's unit normal, $\hat{n}^j(x, y)$, provided by the triangular mesh. In case the site is found on the boundary between two or more triangular elements, the triangle with maximum slope is used. Slope of the j^{th} site, θ^j, is defined as the cosine of the angle between $\hat{n}^j(x, y)$ and the normalized gravity vector $\hat{g}$, Equation (6). To create a slope metric, the inner product between $\hat{n}^j(x, y)$ and the normalized gravity field vector is found. We assume a flat earth gravity model and θ^j reduces to the z component of $\hat{n}^j(x, y)$. We note that $\theta^j \in [0, 1]$ and when $\theta^j = 1$ the terrain has no incline. If the slope exceeds the vehicle slope tolerances, score calculation is terminated and the site is not considered for landing.

$$\theta^j = (\hat{n}^j)^T \hat{g} \tag{6}$$

Roughness Score

Roughness score involves consideration of the landing zone of the j^{th} site, $\mathcal{Z}^j$. We search the quadtree to determine if any of the cells within $\mathcal{Z}^j$ have been marked hazardous for insufficient measurement density or fraction of exceptions greater than λ_e. If so, site scoring is terminated and the site is not considered for landing.

If $\mathcal{Z}^j$ does not contain any hazards, we search the SLAM quadtree to locate all LiDAR measurements, $p_i \in \mathcal{Z}^j$. We form a landing plane $\mathcal{P}^j$ by using the triangular mesh to find $\hat{n}^j$ and c^j as described previously. We find the deviation, d_i, for each of the N_m measurements from $\mathcal{P}$ according to Equation (7). These deviations are used to form a roughness score, R^j, as shown in Equation (8).

$$d_i = p_i^T \hat{n}^j - c^j \tag{7}$$

$$R^j = \sum_{i=1}^{N_m} \sqrt{\frac{d_i^2}{N_m}} \tag{8}$$

Total Score

At this point, a score, s^j, is assigned to the j^{th} potential landing site as a function of the site's predefined interest score, slope, and roughness along with weighting parameters (α_I, α_θ, α_R) which control the relative importance of each consideration. These parameters are chosen according to desired vehicle performance. For example, a high α_I leads to a selection which biases strongly towards scientifically valuable sites, even if they have higher slope or rougher terrain than their alternatives. A score of zero is considered optimal, so sites with the lowest scores represent the best landing sites. The formulation of s^j is shown in Equation (9).

$$s^j = \alpha_I \left(1 - I^j\right) + \alpha_\theta \left(1 - \theta^j\right) + \alpha_R R^j \tag{9}$$

Scoreboard

The LSS system maintains a scoreboard of the top N_s scoring sites. As each site has its score computed, it is referenced with the scoreboard to determine if it should be ranked. Fuel consumption is the final major consideration for the LSS algorithm. To limit computational complexity, fuel consumption is found only for sites which are ranked on the scoreboard. We use the modified Apollo Guidance algorithm [19] to produce an estimate of fuel consumption.

The Apollo Guidance algorithm represents guidance as a two-point boundary value problem where the first point is the vehicle's current state and the second is the targeted landing site. The algorithm fits a polynomial trajectory in time, t, between these two points as a function of the touchdown time, T. As with Apollo, T is chosen to produce a specified maximum value of jerk and snap. The LSS system approximates fuel consumption, F, by integrating required acceleration, $a(t)$, and gravity compensation over the polynomial trajectory as shown in Equation (10).

$$F = \int_0^T |a(t) + g| dt \tag{10}$$

5. Results and Discussion

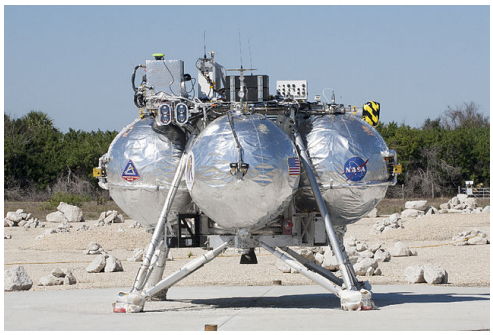


Figure 8: The Morpheus Vertical Take-off and Landing Vehicle. Image Credit: NASA.

We evaluate our algorithm with the data gathered on a rocket-powered vertical take-off and landing vehicle shown in Figure 8. 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 flight 14 (FF 14) [14].

Figure 9: The man-made terrain which was scanned by Morpheus. Image Credit: NASA.

Table 1: Parameters used for validating the LSS system

Parameter	Value
λ_θ	5.0 deg
λ_d	0.3 m
λ_ρ	5.0 meas/m^2
λ_e	0.015
$\lambda_\mathcal{Z}$	2.5 m
α_I	1.0
α_R	100.0
α_θ	1000.0

FF 14 lasted approximately 96 seconds and included multiple phases of flight. During a 20 second scanning phase, the gimbal-mounted LiDAR swept the terrain shown in Figure 9. We use LiDAR-Inertial measurements during 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. Our SLAM system produces localization and map estimates which are fed into the LSS system. 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.

Figure 10 shows a truth DEM of the man-made terrain. Morpheus did not scan the entire man-made terrain. Instead, it scanned only the segment of terrain outlined in red. Notably, some of the grassy terrain shown in Figure 9 was scanned and is free of rocks or major hazards. The data-sets obtained for this work were expressed in the East-North-Up (ENU) coordinate system. To remain consistent, our results throughout this paper are also presented in the ENU coordinate system. The man-made terrain was not constructed with this system in mind, and thus appears rotated in Figure 10 and subsequent figures.

Table 1 documents the values of all tuning parameters and tolerances used in the LSS system.

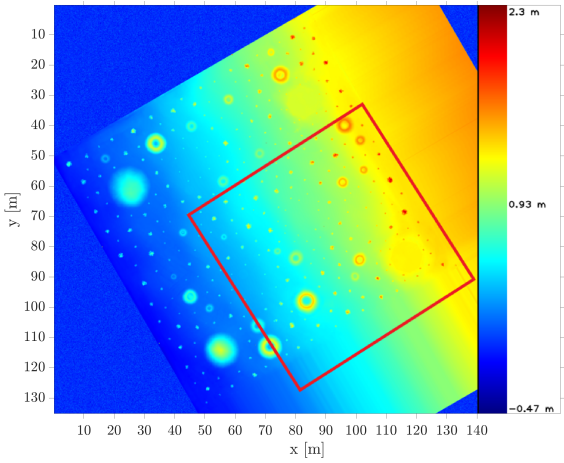


Figure 10: A DEM of the man-made terrain which was scanned by Morpheus. The red box approximately outlines the section of the terrain which is scanned with the LiDAR.

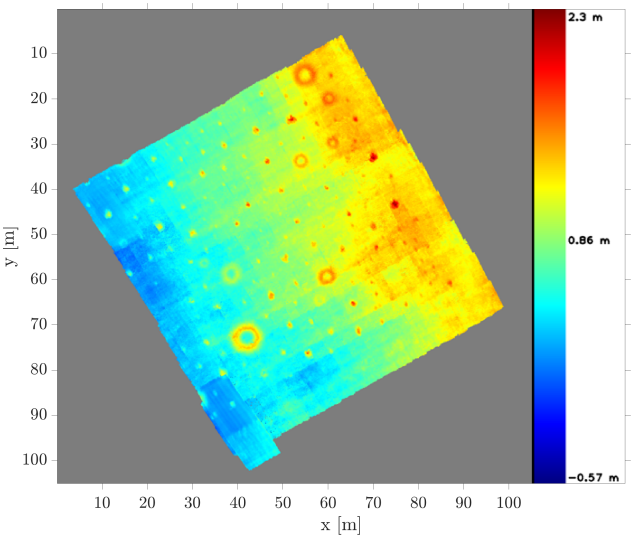


Figure 11: Mapping elevation estimates after the scanning portion of the Morpheus flight.

A detailed discussion of the SLAM system performance is available in the work presented by Setterfield *et al* [20]. Figure 11 shows the map estimates and Figure 12 shows the estimate error. In this and all remaining figures, grey portions of the image because they have not been scanned or have insufficient measurement density. In general, mapping errors are less than 0.5m.

Figure 13 shows LSS slope estimates and Figure 14 shows estimate error when compared with the truth slope. Slope truth is generated by forming a triangular mesh with every cell from a down-sampled the truth DEM with ground sample distance of $2\lambda_\mathcal{Z}$. Truth surface normals are taken as the surface normals of this truth mesh. Error is found as the angle between the truth surface normal and the estimated surface normal. Across the board we achieve slope errors which are less than 6 degrees. The highest slope errors occur in the neighborhood of a relatively large crater.

Figure 15 shows the exception ratio produced by the LSS

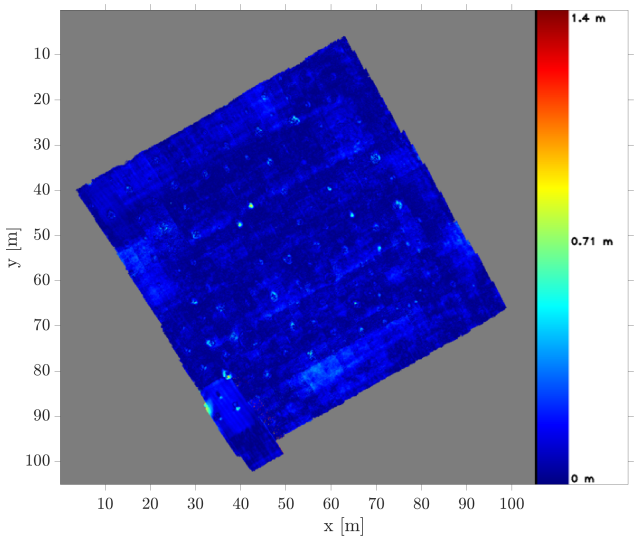


Figure 12: Elevation estimate error after the scanning portion of the Morpheus flight.

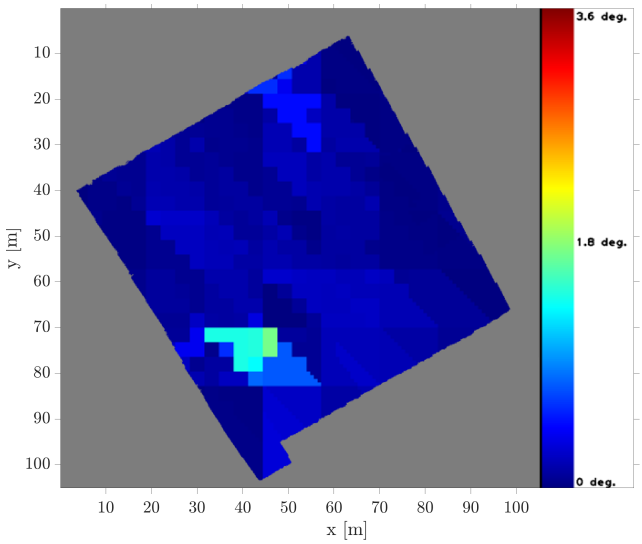


Figure 13: Slope estimates produced by the LSS system.

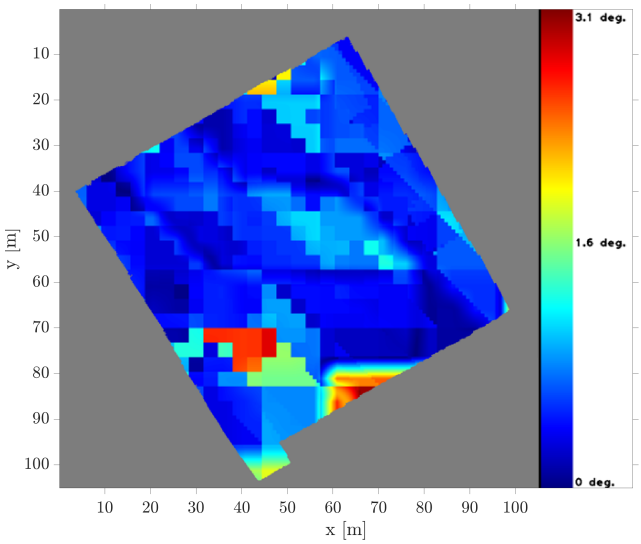


Figure 14: The slope estimate error. High slope errors are seen in the presence of a crater.

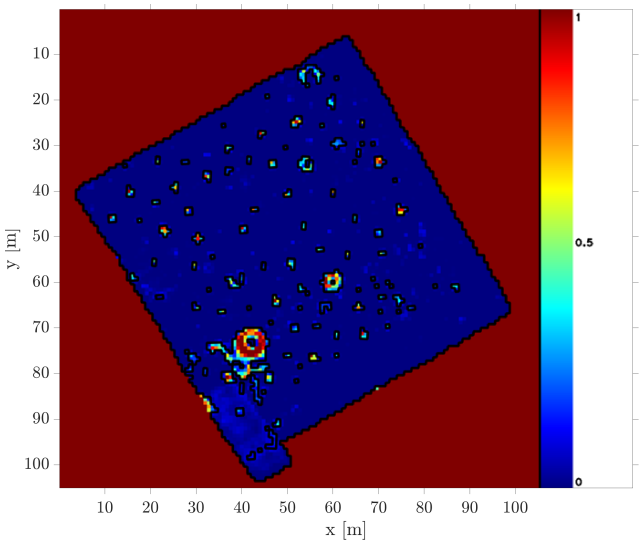


Figure 15: The exception ratio for the scanned terrain. Cells with an exception ratio $> \lambda_e$ are outlined in black.

system. We overlay the hazard on the truth elevation in Figure 16. The hazard boundaries occasionally lie in areas without measurements due to the discretization size of the hazard map. Almost all rocks and craters are successfully identified. However, the lower left region shows a large safe area which was erroneously declared hazardous. This area was scanned before MRL lock occurred. Since no locks were established concurrently with these measurements, their re-projection is subject to inertial measurement unit (IMU) drift. This results in many points lying outside of the roughness tolerances and the cells being marked hazardous. No portion of any site's landing zone, $\hat{\mathcal{Z}}$, is allowed to overlap with any terrain marked hazardous.

To further validate our results, we generate a truth hazard map using the previously discussed truth mesh. We find the orthogonal distance between each element of the full resolution truth DEM and the truth mesh. If this distance exceeds roughness tolerances, we mark that cell as unsafe on the truth hazard map. We compare our hazard map with the truth hazard map in Figure 17. We note that only a couple decimeter scale hazardous areas have been marked safe by our system. These areas could possibly be categorized correctly with denser LiDAR scans. Relatively large amounts of safe terrain has been incorrectly marked hazardous. One potential pitfall would be that the system is too conservative in marking territory hazardous and there is no remaining area large enough for the lander. Our algorithm accurately classifies enough safe terrain that this is not an issue.

Figure 18 shows the landing sites overlaid on the truth hazard map at the end of the scanning phase of flight. None of the landing zones contain true hazards, indicating that our algorithm successfully avoids them.

Figure 19 shows the LSS quadtree after 13 seconds of flight. Portions of the map with a fine resolution have been scanned by the LiDAR. Portions without scans remain at a course

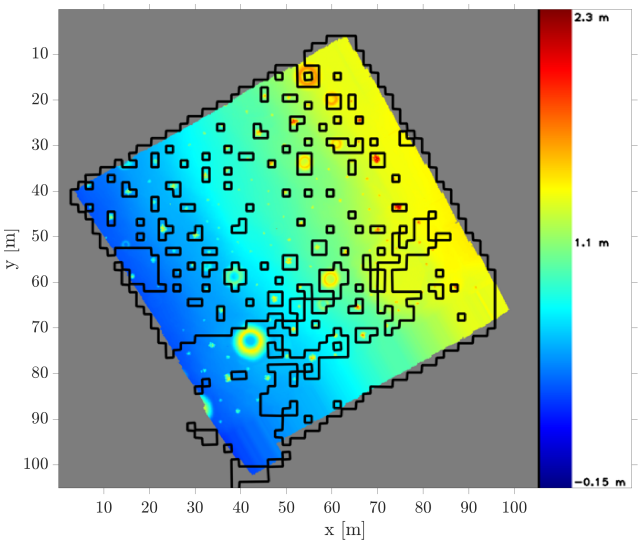


Figure 16: The estimated hazards overlaid on truth elevation. Black boxes outline cells which have been identified as hazardous.

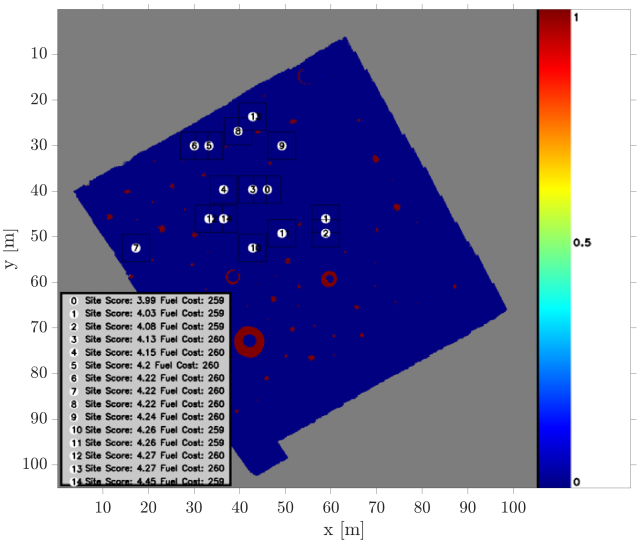


Figure 18: The truth hazard map with landing sites after scanning has been completed. Blue regions are safe while red ones are hazardous.

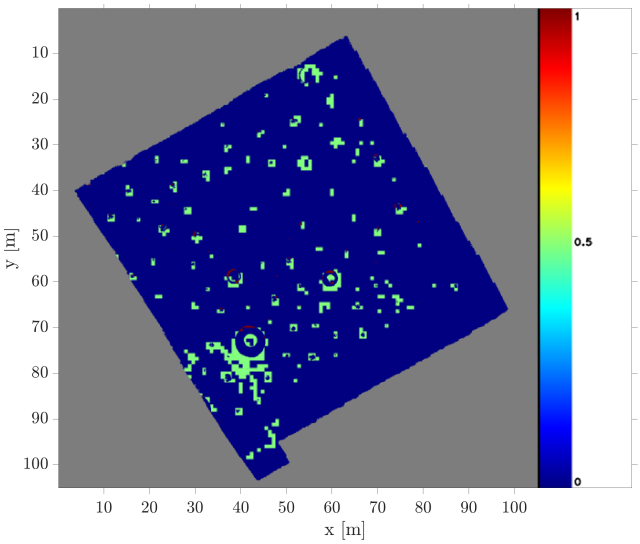


Figure 17: The hazard identification error. Blue regions have correctly been identified. Green regions are estimated hazardous but in truth safe. Red regions are estimated safe but in truth hazardous.

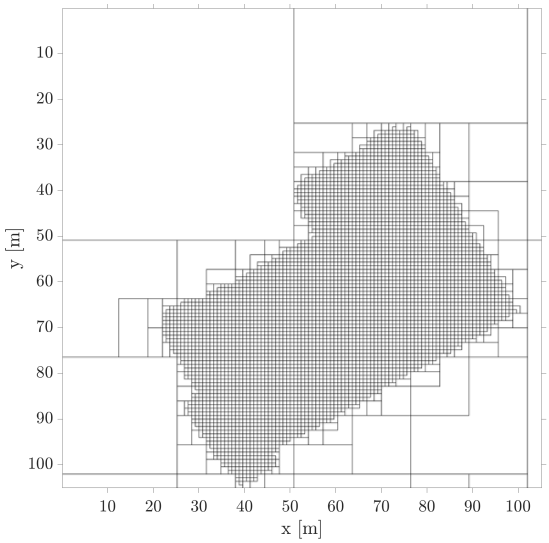


Figure 19: LSS quadtree after 13 seconds. Lines indicated the division between quadtree cells. Divisions after the 8th are not illustrated due to sizing limitations. Note the high grid resolution in the portions of terrain which were scanned by the LiDAR.

resolution to reflect their uncertainty. Figure 20 shows the quadtree at 20 seconds of flight. At this point, Morpheus has completed its scan.

Figure 21 shows the LSS outputs at 13 seconds overlaid onto the truth elevation terrain. Each of these sites has also had its fuel consumption estimated. Note that Morpheus is still scanning the terrain at this time. The sites produced are the current best estimated sites given the current data. Looking ahead to 20 seconds, Morpheus has finished its scanning and is now in the feature lock segment of flight. Figure 22 shows the LSS output overlaid onto the truth elevation terrain at 20 seconds. Some of the landing sites remain in the same place but the optimal one is located in an area of the map which had not yet been scanned at 13 seconds.

Fuel consumption is relatively constant among the sites. This is because the distances between the vehicle and sites is much greater than the distances between the sites themselves.

The final consideration for site selection is the predefined interest metric. A map of this metric was created and shown in Figure 23. All sites are located on higher interest regions of the map. None are able to select the highest interest regions because of the presence of hazards.

6. Conclusion

We have presented a landing site selection algorithm which is optimized for searching a SLAM generated quadtree map.

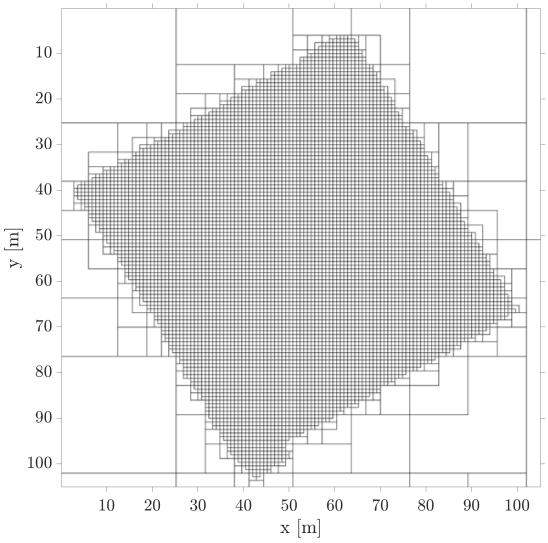


Figure 20: LSS quadtree after 20 seconds

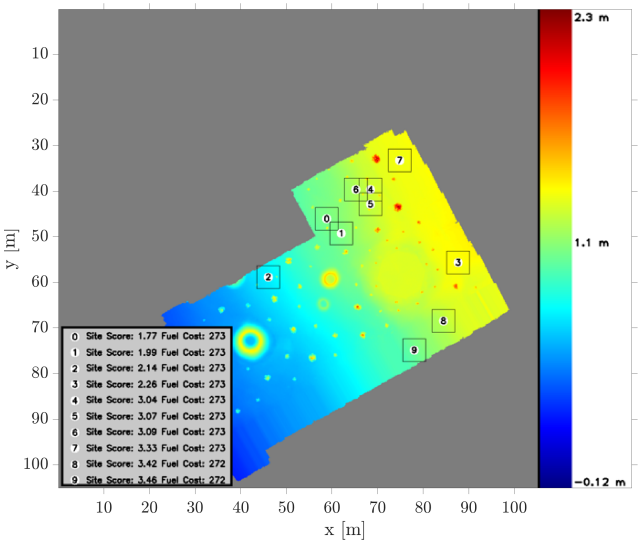


Figure 21: The truth elevation map with selected landing sites after 13 seconds. Black boxes around each site denote the landing zone, $\mathcal{Z}$. Dark black lines indicate regions which are estimated to be hazardous. The legend details the score of each site and its estimated fuel cost.

The algorithm considers slope, roughness, and previously defined interest in weighting sites. The most promising candidates have a rudimentary fuel consumption metric calculated. We show that our system performs with real data gathered from a rocket powered lander.

ACKNOWLEDGMENTS

We would like to acknowledge Nikolas Trawny and the ALHAT team for providing flight data from Morpheus. Additional thanks to Nikolas and Andrew Johnson for their assistance in reviewing this work. This work was supported by a NASA Space Technology Graduate Research Opportunity, NASA grant number: 80NSSC20K1195. The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. ©2021. California Institute of Technology. Government sponsorship acknowledged.

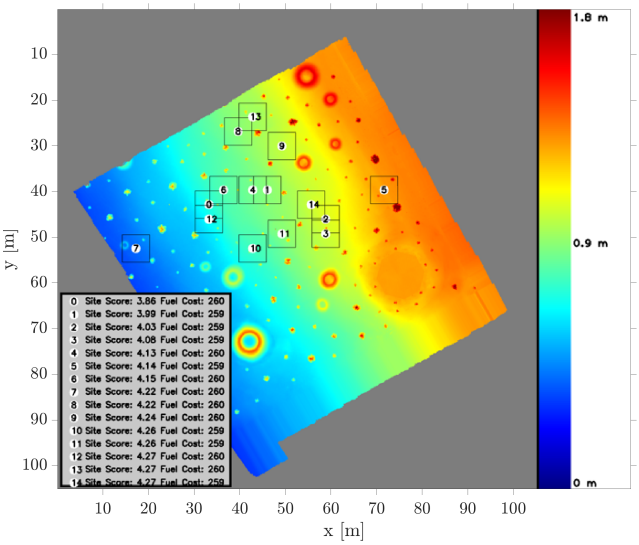


Figure 22: The truth elevation map with selected landing sites after 20 seconds.

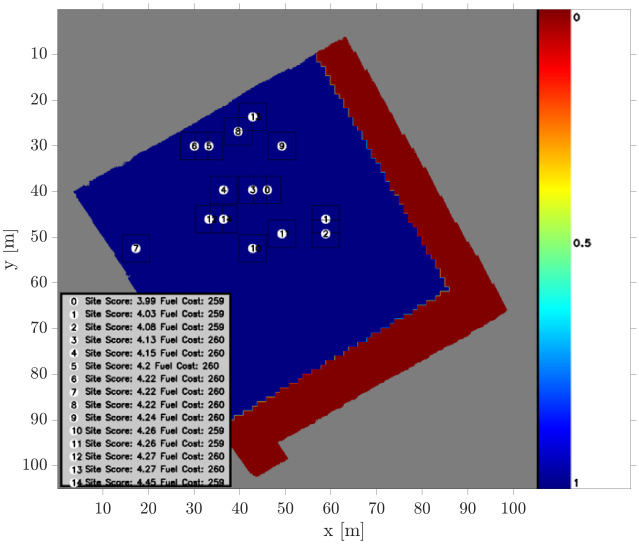


Figure 23: The predefined interest metric. Blue sites ($I = 1$) are more desirable while red sites ($I = 0$) are avoided by the algorithm. Selected sites are also plotted. Black boxes around each site denote the landing zone, $\mathcal{Z}$.

REFERENCES

[1] A. E. Johnson, A. R. Klumpp, J. B. Collier, and A. A. Wolf, "Lidar-based hazard avoidance for safe landing on mars," *Journal of guidance, control, and dynamics*, vol. 25, no. 6, pp. 1091–1099, 2002.

[2] N. Serrano, "A Bayesian framework for landing site selection during autonomous spacecraft descent," *IEEE International Conference on Intelligent Robots and Systems*, pp. 5112–5117, 2006.

[3] N. Serrano and H. Seraji, "Landing site selection using fuzzy rule-based reasoning," *Proceedings - IEEE International Conference on Robotics and Automation*, no. April, pp. 4899–4904, 2007.

[4] B. E. Cohanim and B. K. Collins, "Landing point designation algorithm for lunar landing," *Journal of Spacecraft and Rockets*, vol. 46, no. 4, pp. 858–864, 2009.

[5] T. Ivanov, A. Huertas, and J. M. Carson, "Probabilistic hazard detection for autonomous safe landing," *AIAA Guidance, Navigation, and Control (GNC) Conference*, pp. 1–13, 2013.

[6] P. Cui, D. Ge, and A. Gao, "Optimal landing site selection based on safety index during planetary descent," *Acta Astronautica*, vol. 132, no. October 2016, pp. 326–336, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.actaastro.2016.10.040

[7] M. E. Luna, E. Almeida, G. Spiers, C. Y. Villalpando, A. E. Johnson, and N. Trawny, "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, pp. 1–14, 2017.

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

[9] D. Mango, R. Opromolla, and C. Schmitt, "Hazard detection and landing site selection for planetary exploration using LIDAR," *2020 IEEE International Workshop on Metrology for AeroSpace, MetroAeroSpace 2020 - Proceedings*, pp. 392–397, 2020.

[10] Y. Jung, S. Lee, and H. Bang, "Digital Terrain Map Based Safe Landing Site Selection for Planetary Landing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 368–380, 2020.

[11] T. Yang, P. Li, H. Zhang, J. Li, and Z. Li, "Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments," *Electronics (Switzerland)*, vol. 7, no. 5, 2018.

[12] K. Tomita, K. Skinner, K. Iiyama, B. Jagatia, T. Nakagawa, and K. Ho, "Hazard Detection Algorithm for Planetary Landing Using Semantic Segmentation," pp. 1–11, 2020.

[13] P. Schoppmann, P. Proença, J. Delaune, M. Pantic, T. Hinzmann, L. Matthies, R. Siegwart, and R. Brockers, "Multi-resolution elevation mapping and safe landing site detection with applications to planetary rotorcraft," 10 2021.

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

[15] S. Li, X. Jiang, and T. Tao, "Guidance summary and assessment of the chang'e-3 powered descent and landing," *Journal of Spacecraft and Rockets*, vol. 53, no. 2, pp. 258–277, 2016.

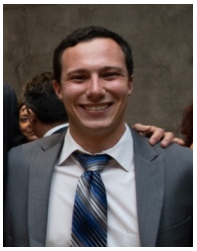
[16] A. Nelessen, C. Sackier, I. Clark, P. Brugarolas, G. Villar, A. Chen, A. Stehura, R. Otero, E. Stilley, D. Way *et al.*, "Mars 2020 entry, descent, and landing system overview," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–20.

[17] B. B. Time, E. E. Interface, M. M. O. A. Sensor, O. O.-B. D. Handling, O. O. O. A. Sensor, and P. P. A. Sensor, "The tianwen-1 guidance, navigation, and control for mars entry, descent, and landing."

[18] T. P. Setterfield, R. A. Hewitt, P.-T. Chen, A. T. Espinoza, N. Trawny, and A. Katake, "Lidar-inertial based navigation and mapping for precision landing," in *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–19.

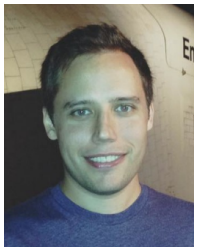
[19] A. R. Klumpp, "Apollo lunar descent guidance," *Automatica*, vol. 10, no. 2, pp. 133–146, 1974.

[20] T. P. Setterfield, R. A. Hewitt, P.-T. Chen, N. Trawny, and A. Katake, "Real world testing of lidar-inertial based navigation and mapping for precision landing," in *2022 IEEE Aerospace Conference*, 2022, work forthcoming.

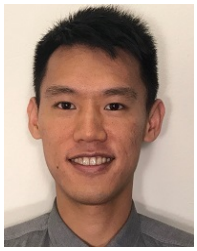


Corey L. Marcus *is a PhD student and NSTGRO Fellow at the University of Texas at Austin. He obtained his BS in Mechanical Engineering from the Ohio State University. He received his MS in Aerospace Engineering from the University of Texas at Austin. His research focuses on guidance and navigation for spacecraft during landing.*

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 Queen's 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.*