Data-Driven Surface Traversability Analysis for Mars 2020 Landing Site Selection

Masahiro Ono, Brandon Rothrock, Eduardo Almeida, Adnan Ansar, Richard Otero, Andres Huertas, and Matthew Heverly
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109
{ono, brothroc, ealmeida, aiansar, otero, mheverly }@jpl.nasa.gov

***Abstract*—The objective of this paper is three-fold: 1) to describe the engineering challenges in the surface mobility of the Mars 2020 Rover mission that are considered in the landing site selection processs, 2) to introduce new automated traversability analysis capabilities, and 3) to present the preliminary analysis results for top candidate landing sites. The analysis capabilities presented in this paper include automated terrain classification, automated rock detection, digital elevation model (DEM) generation, and multi-ROI (region of interest) route planning. These analysis capabilities enable to fully utilize the vast volume of high-resolution orbiter imagery, quantitatively evaluate surface mobility requirements for each candidate site, and reject subjectivity in the comparison between sites in terms of engineering considerations. The analysis results supported the discussion in the Second Landing Site Workshop held in August 2015, which resulted in selecting eight candidate sites that will be considered in the third workshop.**

TABLE OF CONTENTS

1 INTRODUCTION 1
2 PROBLEM DESCRIPTION 2
3 OVERVIEW OF ANALYSIS METHOD 3
4 AUTOMATED TERRAIN CLASSIFICATION 4
5 AUTOMATED ROCK DETECTION 5
6 DEM GENERATION 5
7 MULTI-ROI ROUTE PLANNING 7
8 ANALYSIS RESULTS 8
ACKNOWLEDGMENTS 10
REFERENCES 10
BIOGRAPHY 11

1. INTRODUCTION

The success of planetary surface exploration missions is dependent on the ability of a rover to traverse the terrain in order to accomplish the mission objectives. The Mars 2020 Rover (M2020) mission and a potential Sample Retrieval and Launch (SRL) rover mission are even more contingent on efficient rover traverse performance than the Mars Science Laboratory (MSL) mission. MSL's science goal is incremental, meaning that the more the rover drives the greater science return you get. In contrast, the science goal of M2020 is somewhat binary. It involves the collection of rock and regolith samples, which could be returned to Earth by a SRL and a future sample return orbiter [1]. As a result, the science objectives of M2020 will not be fully met unless the rover successfully drives to the prespecified regions of interest (ROIs) and completes the sample collection.

(C) 2015 California Institute of Technology. Government sponsorship acknowledged.

The M2020 mission is currently in the process of narrowing down candidate landing sites through a series of four workshops held in 2014-2018. The candidates are very diverse in terms of science content, the distribution of ROIs, and terrain characteristics. From an engineering standpoint, for each candidate site, we need to identify 1) whether the rover can land safely and 2) whether the rover can visit the required number of ROIs during the duration of the surface mission allocated to driving. These analyses are performed using the HiRISE images taken by the Mars Reconnaissance Orbiter, which has a nominal 0.3-meter/pixel resolution. While HiRISE imagery enables landing site analysis with an unprecedented level of detail, in practice, manually performing detailed analysis for all candidate sites is not possible due to the significant volume of data.

We address the challenge by developing a suite of automated analysis capabilities called Mars 2020 Traversability Tools (MTTTT), which include terrain classification, rock detection, stereo processing, and optimal route planning. Terrain type, rock abundance, and slope are translated to an estimated driving speed using a mobility model of the rover.

The newly-developed *sequential Dijkstra algorithm* finds distance-optimal and time-optimal routes from any location of a map to satisfy ROI requirements. Running the route planner everywhere in the map results in a cost map, where the cost is the required driving distance/time. The cost map is used for statistical evaluation of landing sites. For a given center point of the landing ellipse, the probability distribution function (PDF) of landing location is specified. By integrating the cost map with the landing PDF, a cumulative distribution function (CDF) of the required driving distance/time is obtained. The CDFs are used to compare between sites quantitatively in terms of driving requirement. Furthermore, the cost map can be used for entry, descent, and landing (EDL) planning. More specifically, we perform a multi-objective optimization of the landing ellipse placement, where the objective functions involve landing safety and the expected driving distance/time. The concept of such a combined EDL and mobility analysis was initially explored by [2], [3], which developed the combined EDL-mobility analysis tool (CEMAT). The approach in this paper is different in that [2], [3] formulated the problem as a chance-constrained optimization where the cost (distance/time) is minimized with an upper bound on the probability of landing failure, while we perform multi-objective optimization.

The rest of the paper is organized as follows. Section 2 defines the objective of analysis as well as puts readers in the context of M2020 landing site selection. Section 3 provides the overview of the MTTTT capabilities, followed by Sections 4-7 that summarizes the technical approach of each capability included in MTTTT. Finally, Section 8 presents preliminary results of the M2020 landing site analysis.

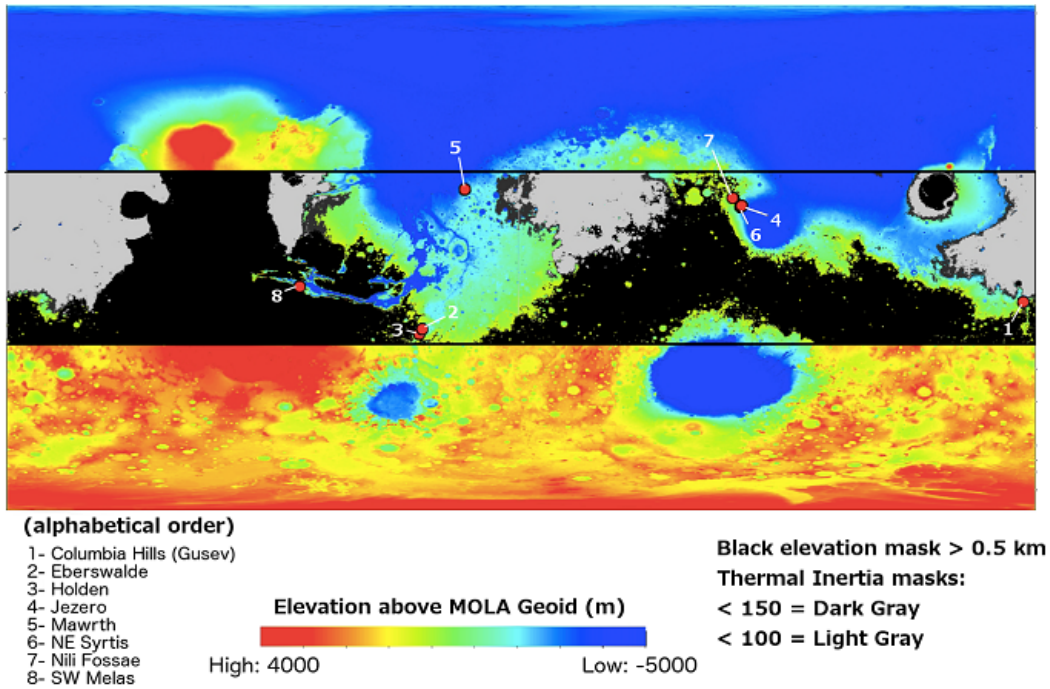


Figure 1: Eight candidate landing sites for the Mars 2020 Rover mission, as of the writing of this paper. Courtesy NASA/JPL-Caltech.

2. Problem Description

The Mars 2020 mission is part of NASAs Mars Exploration Program, which is a long-term effort to explore and better understand the red planet. Specifically, the Mars 2020 rover will look for signs of ancient life, as well as prepare and characterize Martian samples for return to Earth by a potential subsequent mission. The rover will explore two scientifically diverse ROIs, allowing the science team to characterize multiple ancient environments.

The mission is designed to accomplish its objectives in 1.25 Mars years, which is 836 sols or Martian days. The specific landing site for the mission, however, will not be selected until just prior to launch to allow the maximum amount of time for the science community to select the best sites. At the time of the writing of this paper, there are eight candidate landing sites that are being evaluated for both the science value and engineering vaibility: Columbia Hills (Gusev Crater), Eberswalde, Holden, Jezero Crater, Mawrth, Northeast Syrtis, Nili Fossae, and Southwest Melas. The geographical distribution of the eight sites are shown in Figure 1.

To allow the engineering team to design the mission and rover capabilities prior to a detailed analysis of each landing site, a baseline reference scenario has been created. This presents a single representative mission scenario to drive the design of the system capability. This reference scenario has the rover traversing 6 km from the landing site to the first ROI. Once at the ROI the rover will traverse 1.5 km within the ROI to characterize the geology and collect 10 samples for return to Earth. The rover will then traverse another 6 km to reach the second ROI, where it will again traverse 1.5 km within the ROI and collect an additional 10 samples. This notional scenario is illustrated in Figure 2.

The objective of the work described in this paper is to allow

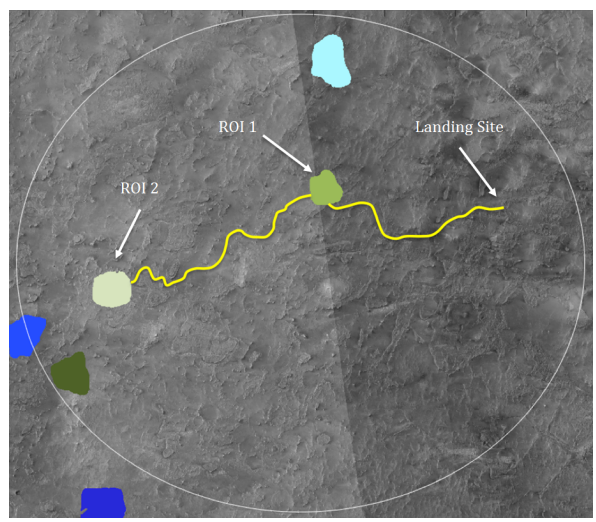


Figure 2: A baseline reference scenario, which represents mission scenario to drive the design of the system capability.

for more site-specific analysis of each proposed landing site. This will allow the landing sites to be evaluated on the likelihood of achieving mission success, and it will also allow the project to better understand if the baseline reference scenario is appropriately bounding for the mission design.

Example of ROI Requirement

Each candidate site has a unique set of ROIs and priorities among them, which are determined to satisfy the top-level mission objectives. Figure 3 shows an example of the ROIs for NE Syrtis. Out of the four types of ROIs shown in the

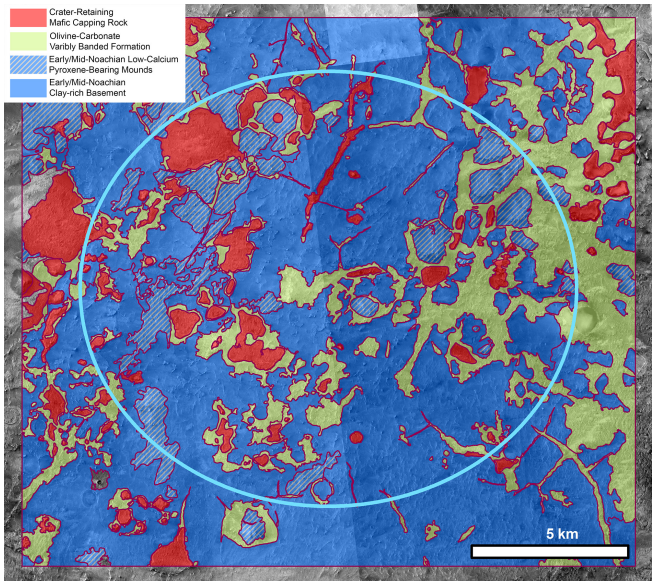


Figure 3: Example of ROI requirement for NR Syrtis. At least two ROIs must be visited, and at least one from ROIs 3-6 must be included. Note that the ROI requirements are updated throughout the landing site selection process and hence this example is not the final ROI requirement.

figure, the rover must visit at any of the olivine-carbonate variably banded formation (shown in green) *and* any of the crater-retaining capping mafic rock (shown in red). The blue ellipse is the landing ellipse, which is the 3-sigma ellipse of the probability distribution of the landing point. The size of the ellipse is 16 km x 14 km. The best combination of ROIs must be chosen to minimize the driving distance or time to visit them from a given landing point. In addition, routes must be planned to avoid the hazards, such as rock fields, deep sands, and steep slopes.

M2020 Rover Capability

The Mars 2020 mission is a high heritage mission based on the Mars Science Laboratory (MSL), which successfully landed on Mars on August 6th, 2012. The M2020 rover will be very similar to the MSL Curiosity rover with the exception of new science instruments and a new sample coring and caching system. Because the mobility system will have very few changes from MSL, the traverse capabilities of the vehicle are well characterized [4].

In bedrock or cohesive regolith terrain, the vehicle is able to climb slopes up to approximately 20 degrees. On large-scale sand dunes, however, the rover is only able to traverse up slopes of approximately 10 degrees. The vehicle also has difficulty traversing in ripple fields where the amplitude of the sand ripples is 10 cm or greater since these ripples cause motion resistance resulting in excessive wheel sinkage and high vehicle slip, and therefore ripple fields at any slope should be avoided [5].

For Mars 2020 it is anticipated that much of the traverse distance between ROIs will be done autonomously to maximize the time the science team has to explore within the ROI. The performance, in terms of traverse rate, of the autonomous navigation (AutoNav) algorithm is highly dependent on the number of obstacles that the vehicle must avoid. This rock

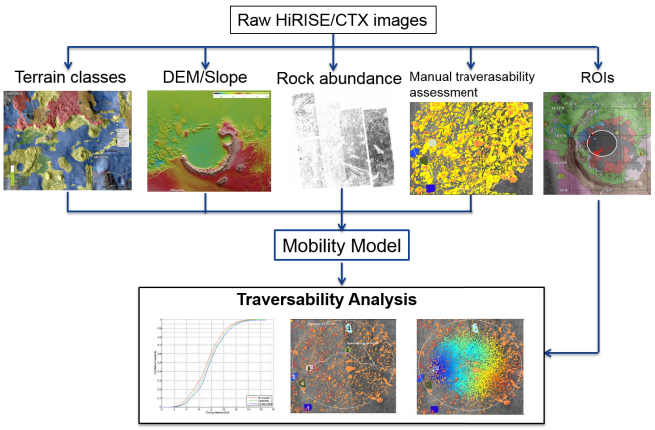


Figure 4:

abundance is characterized from orbit by a rock Cumulative Fractional Area (CFA) [6]. It is believed that for areas with a CFA value below 7% that the rover will be able to drive at its high speed AutoNav rate of approximately 80 m/hr. When the rock abundance is between 7% and 15% it is believed that the AutoNav algorithm will be able to find a path through the rock field, but that it will be necessary to acquire and process more information in order to find a safe path and will therefore drive at the reduced rate of approximately 60 m/hr. If the rock abundance is greater than 15% it is believed that the AutoNav algorithm will frequently be unable to find a safe path, and therefore these rocky regions should be avoided. In Section 8, we specify the mobility model assumed in the analysis, which maps terrain type, CFA, and slope into driving speed.

3. Overview of Analysis Method

Figure 4 shows the overall flow of the analysis performed by MTTTT, which includes the following capabilities:

- **Automated terrain classifier** identifies a terrain type for each pixel of the image, given a full-resolution HiRISE image. Examples of terrain types include smooth regolith, outcrop, or sand dunes. The output from the terrain classifier is a terrain type map.
- **DEM generation algorithm** performs stereo processing to obtain a digital elevation model (DEM), from which a slope map is derived.
- **Automated rock detection** visually identifies the location and size of rocks by finding their shadows. The local rock abundance level is identified by fitting the local size distribution of rocks to a theoretically predicted distribution[6]. The final output from the rock detection tool is a CFA map.
- **SILT (SILT Is a Labeling Tool)** collects manually generated information on ROIs and hazards. SILT is a web-based annotation and visualization tool, which allows users to continuously zoom from the entire ellipse to the full HiRISE resolution like Google Map. Annotations of hazards and ROIs are turned to a hazard map and an ROI map for each site.
- **Mobility model** is essentially a cost function for the optimal route planning, which takes as inputs terrain type map, slope map, CFA map, and hazard map, and outputs a cost map. In an abstract sense, it represents the rovers capability on a given terrain condition. More specifically, we use two types of mobility model. One is a binary mobility model

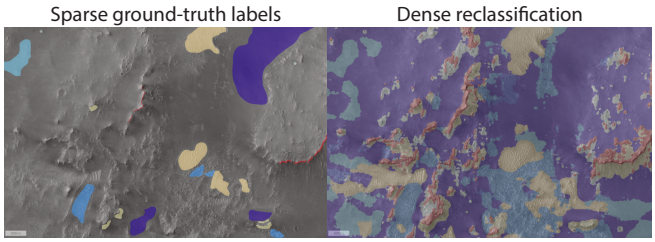


Figure 5: Terrain classification example on NE Syrtis. Ground-truth terrain labels are potentially very sparse, requiring the classifier to infill unlabeled terrain.

that tells whether each pixel on the map is traversable or untraversable. We set thresholds for CFA and slope, and if either of the thresholds are exceeded we identify the pixel as untraversable. The pixels covered by untraversable terrain classes or hazards are also identified as untraversable. The binary mobility model is used for distance-minimal route planning. The other model is a continuous-valued model that assigns estimated traverse speed to each pixel. See the description of M2020 rover capability in Section 2 for the details of the model. The continuous-valued model is used for time-optimal route planning.

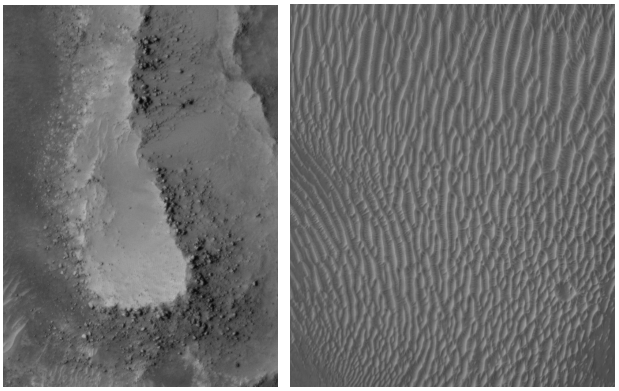
- **Multi-ROI Route Planning** computes a route to visit ROIs with a minimum distance or time.

Each of the analysis capabilities is describe in detail in the following sections.

4. Automated Terrain classification

Terrain classification is the process of labeling each pixel of the orbital image with a terrain type. The resulting terrain map is used by the path planner to estimate both risk and traversal time of a given path. We have identified 11 different terrain types corresponding to different traversability regimes. Terrain maps have traditionally been annotated by hand, which is extremely time-intensive and significantly limits the number of sites that can be analyzed. For this work, we develop an algorithm for terrain classification to enable comprehensive analysis of all candidate landing sites, and at full HiRISE resolution.

The terrain classification problem is commonly referred to as image labeling or semantic segmentation in the computer vision literature. Given pairs of terrain images and ground-truth label masks, the objective is to learn a pixel-to-pixel mapping between terrain and label. Our method is derived from the recent state-of-art work of [7], which utilizes a convolutional neural network architecture to learn a deep hierarchy of image filters. Each label pixel is treated as a multi-way softmax classifier that covers a receptive field of 128×128 pixels in the terrain image. The network also utilizes a "fully convolutional" structure, which converts all inner product layers into equivalent convolutional layers allowing the network to be independent of the input and output image dimensions. Furthermore, the network compensates for the implicit downsampling from strided convolution and pooling with an upsampling mechanism that allows the network to be trained end-to-end without the need for additional post-processing such as using super-pixels or conditional random fields. This method is also very efficient, producing a full image classification in a single forward pass of the network.



(a) Rock field (b) Polygonal ripple field

Figure 6: Example of (a) a rock field and (b) a polygonal ripple field observed from HiRISE orbital images at NE Syrtis.

One of the challenges for training the network is the limited volume of annotated terrain labels, which can easily lead to memorization and poor generalization. This is addressed using supervised pre-training to initialize the network weights using a model trained on a much larger dataset. For our experiments, we use the VGG16 network [8] trained for the ILSVRC14 challenge. The full network is then refined during training with standard backpropagation and SGD from [9]. The loss function sums cross-entropy for each spatial position of the network.

The per-pixel classification performance of the network is currently 94.9%. As shown in Figure 5, our ground-truth labels are typically very sparse due to the time-intensive nature of generating them. This sparsity is accommodated by incurring no loss penalty for any predictions in unlabeled regions. The map is then reclassified by the network, producing a dense map that infills all the previously unlabeled regions. Our ongoing work utilizes expert terrain labelers to provide corrections to the classification results to characterize true classification performance in these previously unlabeled regions.

Training Data Collection

From HiRise orbital images, the user can identify various types of terrain. At the NE Syrtis landing site, 11 unique terrain classes were identified:

- Smooth regoligh
- Smooth outcrop
- Fractured outcrop
- Sparse linear sand ripples
- Rough outcrop
- Craters
- Rock fields
- Dense linear sand ripples
- Polygonal sand ripples
- Deep sand accumulations
- Scarps.

Examples are shown in Figure 6. A small number of examples (ranging from 1 to 10) for each of these terrains was provided as training data for the terrain classifier.

5. AUTOMATED ROCK DETECTION

Landing or driving over rocks can be very hazardous to a rover. The candidate sites on Mars present diverse terrain characteristics in terms of the distributions of rock sizes and rock densities. Thus, a rock detection algorithm is important for determining safe regions based on rock populations. The method of Golombek *et al.* [10] is used to estimate rocks from their shadows and quantify the landing and traversability hazards based on rock abundance, *i.e.*, how much of the surface area is locally covered by rocks. The original rock detection algorithm is described in [11]. Formally, rock abundance is evaluated through a theoretical model of the cumulative fractional area covered by rocks of diameters equal to or greater than some value D [6]. These models are based in fracture and fragmentation theory [12] and are estimated for a HiRISE image based on detected rocks.

Rock detection is performed by segmenting rocks' shadows by an image processing technique that enhances the contrast of the shadow regions [11]. The algorithm then fits an ellipse to each shadow segment and uses the sun incidence angle direction to derive rock height from shadow length and rock diameter from shadow width, according to illumination direction and estimated slope from elevation models (see section 6). Results of shadow detections on Martian terrain are shown in Figure 7. Prior to shadow segmentation, blind image deconvolution is used to sharpen HiRISE images in order to resolve shadows of size ∼5 pixels (1.5 meters at 0.3m resolution) or larger.

In this paper, the size-frequency distribution of rocks is estimated from detected rocks in each image and it is fit to predicted statistical exponential models [10], [13]. From the fits, which are performed in bins of $30m \times 30m$ or $150m \times 150m$, one can infer the amount of missing small rocks that are not resolved and the amount of large rocks that are incorrectly detected (see Figure 8). The rocks used for fitting have estimated diameter between 1.5m and 2.5m, since smaller rocks are not reliably resolved and large boulders are isolated, providing little statistical significance for the fit. In fact, large detections are potentially non-rock objects that cast shadows, *e.g.*, small hills. This fitting technique has been validated in HiRISE images by comparing distributions of rocks estimated from orbit with the ones observed from the ground. This distribution fit significantly improves the detection results. The general size-frequency distribution power law is given by

$$F_k(D) = ke^{-q(k)D}, \tag{1}$$

where D represents rock diameter, $F_k(D)$ is the cumulative fractional area covered by rocks of diameter equal to or greater than D, k is the fraction of the total area covered by all rocks (the local rock abundance) and $q(k)$ governs the decay speed with increasing diameter [6]. Note, $F_k(D)$ is a function of a rock diameter D and $k \in [0, 1]$ is the value of the function evaluated at diameter zero. The value k is traditionally referred to as *rock abundance*, or *CFA* (cumulative fractional area) value, or simply CFA. As illustrated in Figure 8, the rock abundance k is ultimately defined by the curve that parallels the empirical distribution of estimated rocks. Here this is achieved using rocks with diameter $D \in [1.5m, 2.5m]$, which means that the total fractional rock coverage k is inferred only from an statistically significant subset of rocks.

A common alternative way of representing rock size-frequency distributions is based on the cumulative *number* of rocks of certain diameter or larger per unit area, which also follows a power law analogous to (1). The distributions based on fractional area or density, for a given rock abundance, are equivalent and can be derived from each other [10].

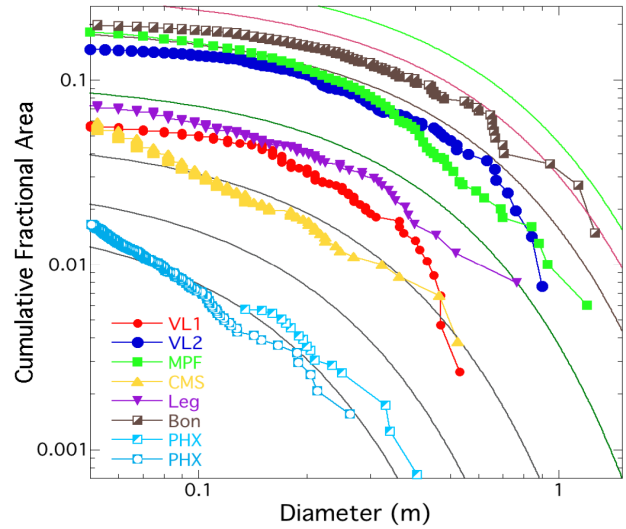


Figure 8: Cumulative fractional area covered by rocks versus measured rock diameter from different landing sites (image from Golombek *et al.* [10]). Each marker represents measurements from a site (see [10] for data sources). Exponential models of size-frequency distributions for 2%, 3%, 5%, 10%, 20%, 30% and 40% rock abundance are also shown. The empirical distributions fit nicely with the theoretical models.

6. DEM GENERATION

Digital Elevation Maps (DEMs) of the Martian surface are generated using stereo imagery taken by the HiRISE camera aboard the MRO spacecraft. The processing chain consists of:

1. Stereo correlation between HiRISE image pairs
2. Triangulation of correlated pixels to generate a point cloud in a Mars-centric coordinate frame
3. Ortho-projection of the point cloud into a DEM and generation of a pixel co-registered ortho-projected image

We describe each step of the process in greater detail.

Stereo correlation

The first step in generating 3D is to match corresponding pixels between two HiRISE images with sub-pixel accuracy. Given the large format (on the order of 20k x 100k pixels) and correspondingly large disparity between matched pixels, we begin this process by computing a best linear transformation between the image pair and warping one image, typically taken off-nadir, to approximately co-register with the other, typically taken near nadir and considered the stereo reference image. The resulting warped image has disparities on the order of tens or hundreds of pixels with respect to the reference image rather than thousands or tens of thousands. This initialization reduces stereo match times by up to two orders of magnitude. From this point, pixel matching is performed using a pseudo-normalized correlation scheme

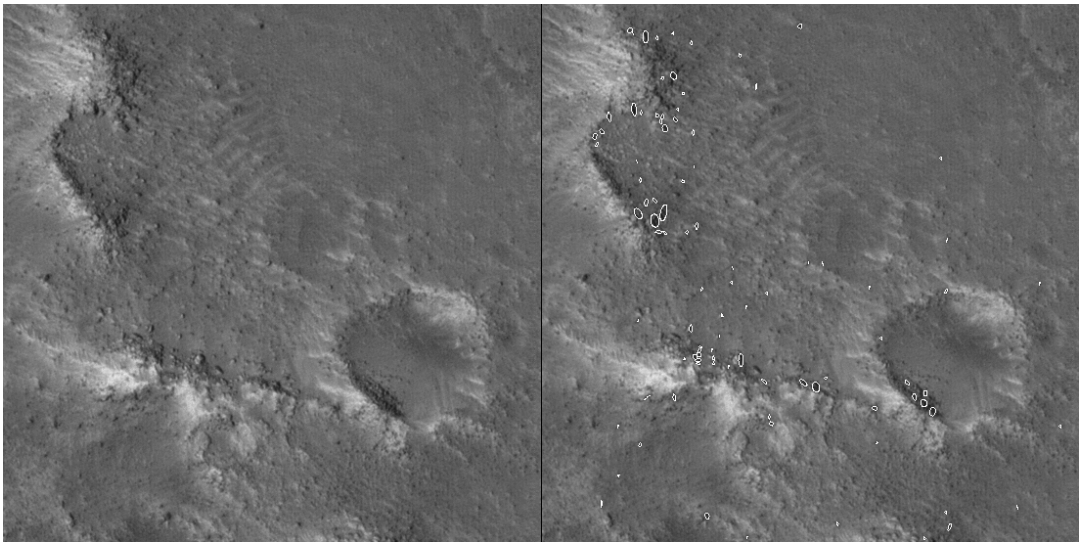


Figure 7: Shadow segments, shown with white outline, estimated from HiRISE orbital images at McLaughlin using the method described in [10].

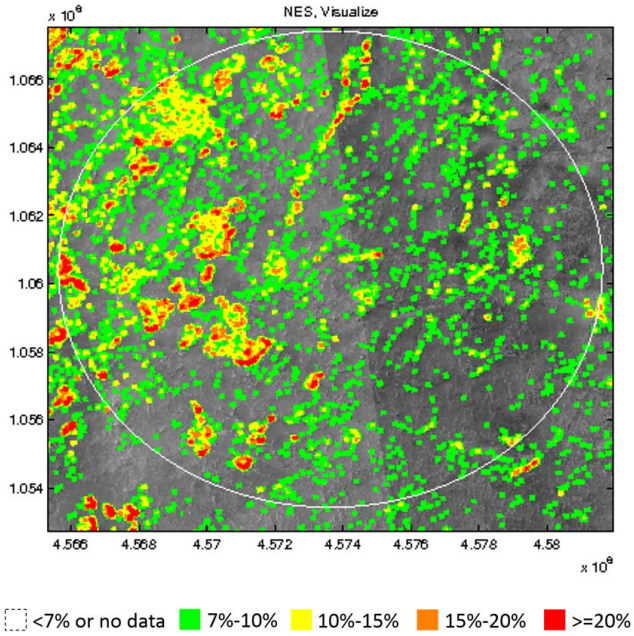


Figure 9: CFA (rock abundance) map on NE Syrtis.

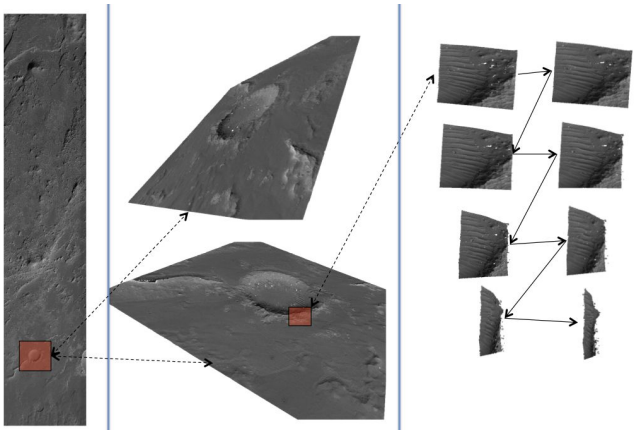


Figure 10: Reconstruction using the DEM generation software. Left: HiRISE image with crater highlighted in red. Center: 2 views of 3D reconstruction of crater. Right: Detail on crater rim slowly rotated out of plane to show structure on edge.

that operates on image sub-blocks and with a coarse-to-fine scheme that further speeds up the process. Correlation on full HiRISE image on a standard Desktop processor using a Matlab implementation and without explicit parallelization can be performed in under 5 hours. Note that this processing takes place on the full resolution reference image and retains the native spatial resolution of that image.

Formation of point cloud

Given sub-pixel correspondence between two HiRISE images, we can generate for each pixel in the reference image 3D point. Using the SPICE framework [14], we can compute for every point on either image a ray in a Mars centric, inertial coordinate frame. The point simultaneously minimizing the distance between rays corresponding to a pixel in the reference image and its sub-pixel match location in the second image is considered to be the 3D point associated with that pixel. Thus for each pixel in the reference image with a match in the second image, we produce a 3D point in the chosen inertial frame. Note that HiRISE is a multi-focal plane integrating pushbroom instrument. Thus, camera position and pointing are a function of the row number in each image. It is impossible to approximate this by a simple model. Therefore the time dependent SPICE ephemeris and attitude must be fully exploited.

Recovery of Digital Elevation Maps from Orbital Imagery

Given a point cloud in an inertial frame, the process to transform this into a DEM makes further use of the SPICE

framework. Mars-relative (x,y,z) coordinates are transformed into (lat., lon., height) coordinates relative to the Mars geoid. These are uniformly resampled at a resolution that can be as fine as the ground sample distance of the reference image. Since the underlying point-cloud is pixel co-registered with the reference image, the resulting DEM will have attached to each posting an index into the underlying reference image. This forms the basis for generating the ortho-projected image corresponding to the DEM.

7. MULTI-ROI ROUTE PLANNING

Finding an optimal route in our problem setting is essentially to solve a variant of the traveling salesman problem (TSP) where goals are regions instead of points. The problem involves two technical challenges:

1. Optimize the combination and the order of ROIs to visit, and
2. Optimize the point in each of the ROIs to visit.

The first challenge turned out not to be difficult. In our analysis we only need to visit just two ROIs, hence the complexity of TSP is at most $N(N-1)$ where N is the number of ROIs. With the polynomial complexity, it is computationally feasible to solve TPS by simply enumerating all the feasible orderings and find the one with the least cost.

The second one is more significant, hence given more emphasis in this paper. To intuitively communicate what the challenge is, see Figure 11. Assume that K ROIs must be visited out of N labeled as $\mathcal{R}_1, \mathcal{R}_2, \cdots \mathcal{R}_K$, which must be visited in this order. We assume that $\mathcal{R}_i$ is a closed compact set in a Cartesian space. For each ROI, a *sampling point*, $x_i \in \mathcal{R}_i$, must be chosen, and then for each pair of adjacent ROIs, a route that connects x_{i-1} and x_i, denoted by p_i, must be planned. This is critically different from regular route planning problems where goals are given as points, in which there is no need to optimize the sampling points. As a result, the following nested optimization problem must be solved:

$$\min_{x_1 \in \mathcal{R}_1 \cdots x_K \in \mathcal{R}_K} \left(\min_{p_1 \in \mathcal{P}_1 \cdots p_K \in \mathcal{P}_K} \sum_{i=1}^{K} c(p_i) \right), \qquad (2)$$

where $\mathcal{P}_i$ is the set of all the feasible routes between x_{i-1} and x_i, x_0 is the landing point, and $c(\cdot)$ is the cost function. In a practical problem, solving this nested optimization problem is usually computationally intractable, even in our special case where $K = 2$. For example, in a typical problem with $N = 2$ where each ROI consists of $\sim$100-by-100 pixels on a map, a naive solution of (2) requires solving the optimal route planning problem (i.e., the inner optimization of (2)) 10^8 times. Furthermore, in order to obtain CDF for statistical analysis, we need to run the optimal route planning from every points in the map; a typical map consists of a few thousand by a few thousand pixels, resulting in $\sim 10^{14}$ route planning problems to be solved.

Sequential Dijkstra Algorithm

To solve this problem efficiently, we developed the *sequential Dijkstra algorithm,* which only requires to run the Dijkstra algorithm K times. The Dijkstra algorithm obtains the cost to go to the goal from *all* nodes in a graph . In our case, each pixel in a map corresponds to a node, where each node is connected to 8 neighboring nodes (8-connected graph). The shortest path from a given node to the goal is obtained by recursively moving to the node that has the least cost among the neighbors of the current node. Note that a route on a 8-connected graph results in 5.4% overestimation of distance on average, Therefore we decrease the cost by 5.4% to obtain a better estimation.

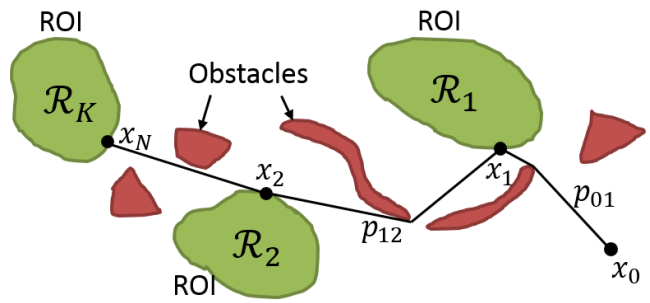


Figure 11: Multi-ROI route planning is challenging because we need to optimize not only the routes between ROIs $(p_{01}, p_{12}, \cdots)$ but also the points in ROIs to go through (i.e., $x_i \in \mathcal{R}_\rangle$.)

The sequential Dijkstra algorithm propagates cost backwards in time, just as the regular Dijkstra for multi-source, single-goal shortest path planning. Figure 12 illustrates the algorithm. First, the algorithm initialize the cost map by assigning 0 to the pixels in $\mathcal{R}_K$ and ∞ to all the other pixels. Then it runs the regular Dijkstra algorithm. The resulting costmap, shown in Figure 12-(a), represents a cost to go from a given start point to a point in $\mathcal{R}_K$ that can be reached with the minimum cost. We denote this cost map by C_K. Then, in the next iteration, the cost values in $\mathcal{R}_{K-1}$ are preserved while all the other pixels are reset to ∞ as in Figure 12-(b), and the regular Dijkstra algorithm is run again. The resulting cost map, shown in Figure 12-(c), represents the minimum cost to go from a given start point to a point in $\mathcal{R}_K$ through a point in in $\mathcal{R}_{K_1}$. We denote the cost map by C_{N-1}. By repeating this process K times, the cost map is derived that represents the cost of the shortest path that visits the K ROIs sequentially starting from anywhere in the map. The final product is a cost map, as shown in FIgure 12-(c). For example, if the cost is distance, it is a distance map where the value at each pixel represents the driving distance to go through all the ROIs in a specified order. Finally, the algorithm stores all the intermediate cost maps since it is used later for shortest path planning.

Generalized Traveling Salesman Problem

Route Reconstruction

In order to construct the shortest path from a given start point x_0, the stored cost maps are used forward in time. Starting from x_0, as shown in Figure 13-(a), a route is extended to the node that has the lowest cost among the neighbors in C_1 until reaching $\mathcal{R}_1$. The point in the route that first hits $\mathcal{R}_1$ is x_1. Then, we switch to C_2, and extend the path in the same manner until reaching $\mathcal{R}_2$, as shown in Figure 13-(b). We repeat the process until the path finally reaches $\mathcal{R}_K$. It is guaranteed by construction that the resulting path goes through $x_1 \cdots x_K$ that minimize the total route cost.

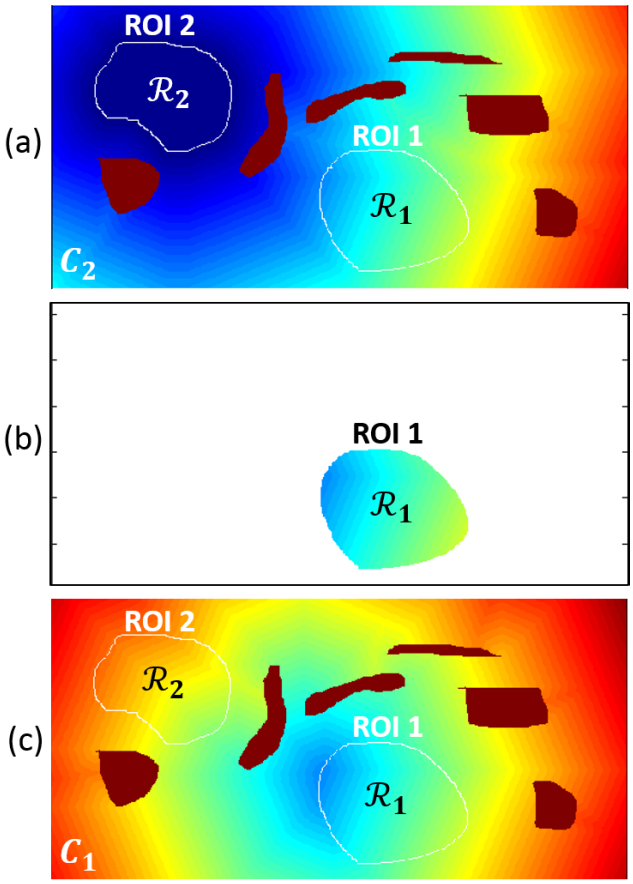


Figure 12: Graphical presentation of the sequential Dijkstra algorithm. At the ith iteration, the algorithm computes the cost to sequentially visit ROIs $K - i + 1 \cdots K$ from all the nodes (pixels) in the map. The color in the cost map corresponds to the cost. This figure shows the case with $K = 2$. (a) is the cost map after the first iteration, (b) is the initialization of the cost map before the second iteration, and (c) is the cost map after the second iteration.

8. Analysis Results

We note that the results presented in this paper is preliminary and not final because 1) ROIs for each site are updated throughout the process of landing site selection, 2) landing ellipse placement is not final, and 3) data (i.e., terrain classification, rock map, and slope map) is not complete. We performed the evaluation of landing sites based on time-optimal route planning using the mobility model described below.

Mobility Model

Based on the discussion on the M2020 rover capability in Section 2, we created a mobility model that maps terrain type, slope, and CFA to the expected driving speed. Figure 14 shows the mobility model used for the analysis. The eleven terrain classes are categorized into four groups. For each group, the expected driving speed per sol is given as a function of CFA and slope. When terrain class is not available, we use the one for smooth regolith, smooth outcrop, and fractured outcrop.

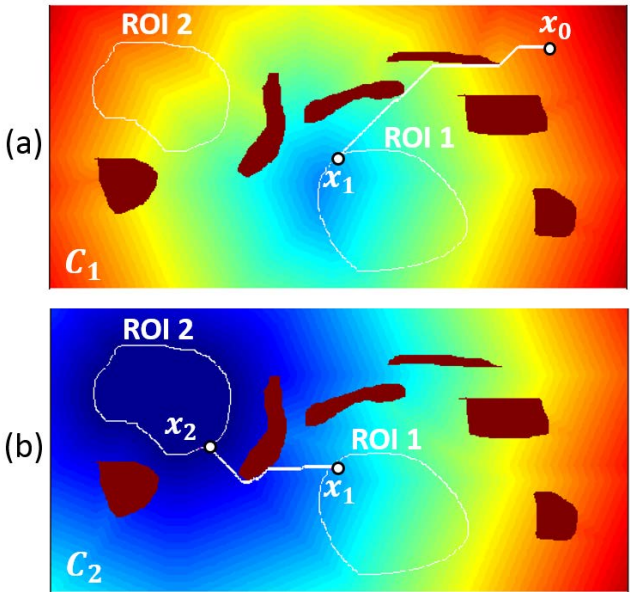


Figure 13: Reconstruction of the optimal path. The optimal route from x_{i-1} to x_i is obtained by using the cost map obtained by the $K-i+1$th iteration of the sequential Dijkstra algorithm.

Analysis Process

For each candidate site, we ran the sequential Dijkstra algorithm to obtain optimal routes. The cost of each pixel of the map was given as:

$$\frac{1}{\text{Driving speed}[\text{m/Sol}]},$$

where the driving speed was given by the mobility model described above. Using this cost function results in the minimization of driving time. The length of the resulting route was also computed.

We were given the probability distribution of landing point computed by a meso-scale wind model, given as a point cloud. The distance map was used to evaluate the driving distance from each of the 16,000 points in the cloud. An example on Jezero Crater is shown in Figure 15, where landing points are colored by the required driving distance. Points on the southeast (bottom right) tend to have longer driving distance because a must-visit ROI (a delta fan) is located on the northeast (top left) of the landing ellipse. Figure 16 shows an example of a route starting from an eastern landing point.

From the distribution of the distance and time of the point cloud, we constructed a cumulative distribution functions (CDFs). Figure 17 shows an example of CDFs. The CDF provides rich information about the correlation between driving distance/time and mission success probability. For example, in Figure 17, if a rover is capable of driving 8 km on surface, there is a 52% chance that the ROI requirement is satisfied. To put it in the other way, in order to be 90 % certain that the ROI requirement is met, the rover must be capable of driving at least 11.2 km. Likewise, we can obtain the CDF for driving time. We used percentile distance and time as the metrics to quantitatively compare between candidate sites. The percentile distance and time were adjusted in two ways to obtain a more realistic estimate of driving distance.

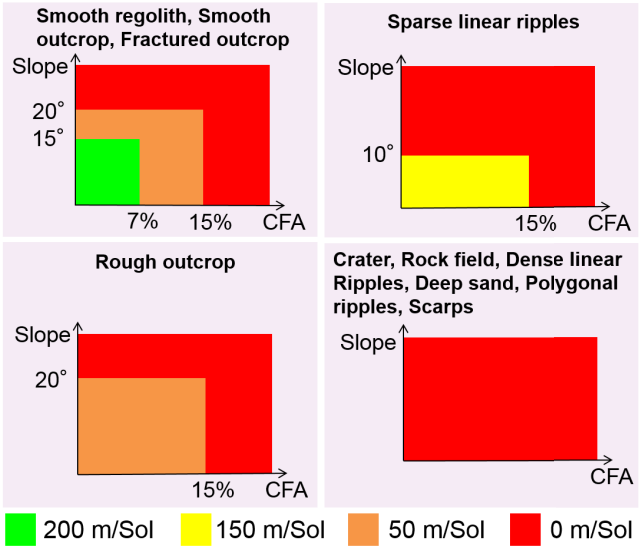


Figure 14: The mobility model assumed in the analysis in this paper. The eleven terrain classes are categorized into four groups. For each group, the expected driving speed per sol is given as a function of CFA and slope.

First, the value was decreased by 5.4% to account for the overestimation of distance by the 8-connected graph. Second, the value was increased by 30% to account for the deviation from the shortest-distance route due to obstacle avoidance and opportunistic science observation. We estimated that the difference in distance is ∼30% by comparing the distance obtained by the distance-optimal route planner and the actual odometry of Curiosity on the Gale crater.

Analysis Results of All Candidates

We performed analysis on the eight candidate landing sites shown in Table 1, which were selected as the top candidates in the Second Landing Site Workshop. Terrain characterization had not been complete yet, and the availability of data differs between candidates. For example, as shown in Table 1, while NE Syrtis and SW Melas had hazard, rock, and slope maps, Nili Fossae and Holden Crater only had a slope map. Therefore, the results on the paper are the current best estimate based on the available data for each site, but comparison between candidates is not completely fair. Nonetheless, we argue that the result is a reasonable approximation of the actual distance because 1) primary hazard types are mostly covered on all sites (e.g., a site known for rock abundance is covered by a rock map) 2) the most influential factor is the distribution of ROIs and the ellipse placement.

Figures 18 and 19 shows the CDFs of driving distance and time for the eight candidate sites. The right four columns of Table 1 shows the 50th and 90th percentile distance and time. Although the size of landing ellipse is almost the same for all sites, the analysis resulted in significantly different driving distance from site two site. The difference is mainly due to two factors. The first factor is whether ROIs are within the ellipse (such sites is called "land-on sites") or outside of it ("go-to site"). For example, Eberswalde has the longest driving distance and time among the eight sites because its landing ellipse is far from the ROIs. The second factor is the distribution of ROIs. For example, in NE Syrtis, the must-visit ROIs are distributed all over the ellipse. Therefore the

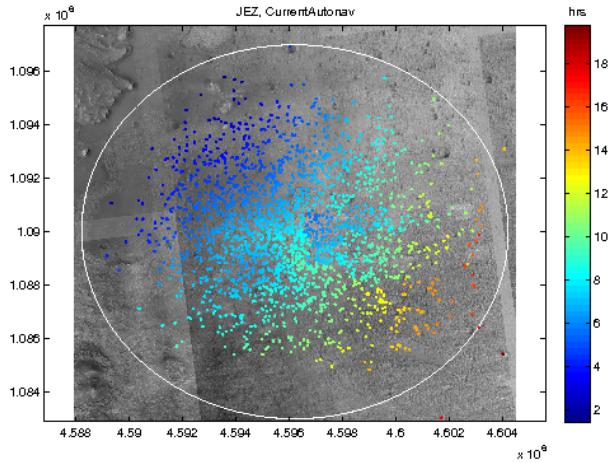


Figure 15: Landing point cloud on Jezero Crater, colored by the time to drive to required ROIs.

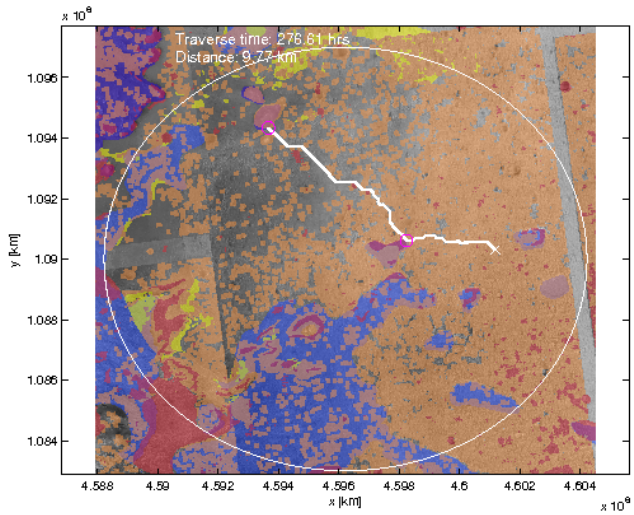


Figure 16: A sample path on Jezero Crater.

ROI requirement is met just by driving to the nearest ROIs from the landing site, hence the resulting distance is small.

The analysis results are being updated continuously. In the Second Landing Site Workshop, a previous version of the analysis results were used to support the discussion. As a result of the workshop, a few sites which were previously ranked highly, such as East Margaritifer and McLaughlin, were dropped. Instead, Columbia Hills and Eberswalde were added to the list of top-ranked sites. After the workshop, additional ROIs were identified for NE Syrtis partially to mitigate the driving distance requirement. Also, the ROI requirement was changed in Holden to convert it from a go-to site to a land-on site, which will result in a significant reduction in the driving distance.

We are still in the process of refining ROIs and landing ellipses, we well as improving data coverage. Accordingly, the analysis will be continuously refined in the future.

Table 1: The eight top candidates selected by the Second Landing Site Workshop and the preliminary analysis results. Also shown are the assumption on the location of the center of the landing ellipse, as well as the availability of data (hazard, slope, and rock maps). The right two columns show the results of the analysis. 50% distance means the distance that the M2020 rover needs to be able to drive in order to meet the ROI requirements.

		Inputs to analysis					Outputs			
Landing Site	Acronym	Latitude	Longitude	Data availability			Distance [km]		Time [sols]	
				Terrain	Slope	Rock	50%	90%	50%	90%
Columbia Hills	CLH	14.590°S	175.534°E	No	No	No	3.0	5.0	20.5	33.7
Eberswalde	EBW	23.858°S	33.185°W	No	No	No	16.6	19.7	112.1	132.8
Holden Crater	HOL	26.417°S	34.799°W	No	Yes	Yes	5.4	8.0	36.4	53.9
Jezero Crater	JEZ	18.389°N	77.541°E	No	No	Yes	7.4	10.5	57.8	99.6
Mawrth Valles	MAW	23.955°N	19.060°W	No	Yes	Yes	1.3	2.2	8.9	15.3
NE Syrtis Major	NES	17.890°N	77.160°E	Yes	Yes	Yes	1.0	1.9	8.4	13.5
Nili Fossae Trough	NIL	21.023°N	74.358°E	No	Yes	No	8.9	11.3	59.9	76.3
SW Melas Chasma	SWM	9.805°S	76.416°W	No	Yes	Yes	1.4	2.7	10.6	22.2

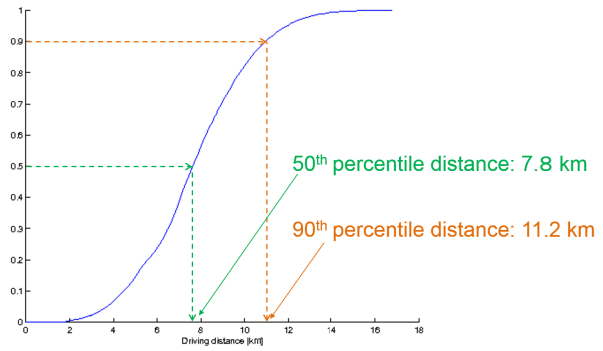


Figure 17: An example of the cumulative distribution function of driving distance.

Conclusion

Three main contributions presented by this paper were:

1. The problem formulation of the landing site selection for the Mars 2020 Rover mission from the surface traversability perspective,
2. The quantitative traversability analysis capabilities including automated terrain classification, automated rock detection, DEM generation, and multi-ROI route planning, and
3. The preliminary analysis result on eight candidate sites.

The analysis results supported the selection of eight candidate sites considered in the Third Landing Site Workshop, as well as guided the modification of ROI requirements and ellipse placement. The analysis will be refined for the third workshop with updated ROI requirements and ellipse location, improved data coverage, and improved mobility model based on terrain classification results.

Acknowledgments

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Government sponsorship acknowledged.

References

[1] J. Mustard, M. Adler, A. Allwood, D. Bass, D. Beaty, J. B. III, W. Brinckerhoff, M. Carr, D. D. Marais, B. Drake, K. Edgett, J. Eigenbrode, L. Elkins-Tanton, J. Grant, S. M. Milkovich, D. Ming, C. Moore, S. Murchie, T. Onstott, S. Ruff, M. Sephton, and A. Steele, "Report of the mars 2020 science definition team," Published online at http://mars.jpl.nasa.gov/mars2020/files/mars2020/SDT-Report%20Finalv6.pdf, 2013.

[2] M. Ono, M. Pavone, Y. Kuwata, and J. B. Balaram, "Chance-constrained dynamic programming with application to risk-aware robotic space exploration," *Autonomous Robots*, 2015.

[3] Y. Kuwata, M. Pavone, and J. B. Balaram, "A risk-constrained multi-stage decision making approach to the architectural analysis of planetary missions," in *Proceedings of the IEEE Conference on Decision and Control*, 2012.

[4] M. Heverly, J. Matthews, J. Lin, D. Fuller, M. Maimone, J. Biesiadecki, and J. Leichty, "Traverse performance characterization for the mars science laboratory rover," *Journal of Field Robotics*, vol. 30, no. 6, pp. 835–846, 2013. [Online]. Available: http://dx.doi.org/10.1002/rob.21481

[5] R. E. Arvidson, F. Zhou, N. T. Stein, M. Heverly, P. Bellutta, M. Maimone, J. P. Grotzinger, K. Iagnemma, A. Fraeman, and D. Rubin, "Mars science laboratory curiosity rover megaripple crossings up to sol 710 in gale crater," *Journal of Field Robotics, publication pending*, 2015.

[6] M. Golombek and D. Rapp, "Size-frequency distributions of rocks on mars and earth analog sites: Implications for future landed missions," *Journal of Geophysical Research: Planets*, vol. 102, no. E2, pp. 4117–4129, 1997. [Online]. Available: http://dx.doi.org/10.1029/96JE03319

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Ima-

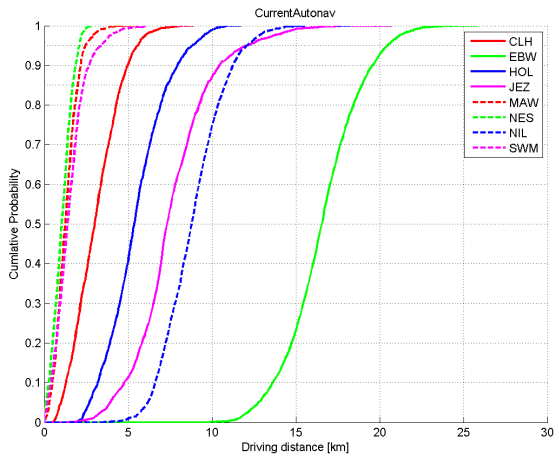


Figure 18: Cumulative distribution function of the required driving distance for the eight candidate sites.

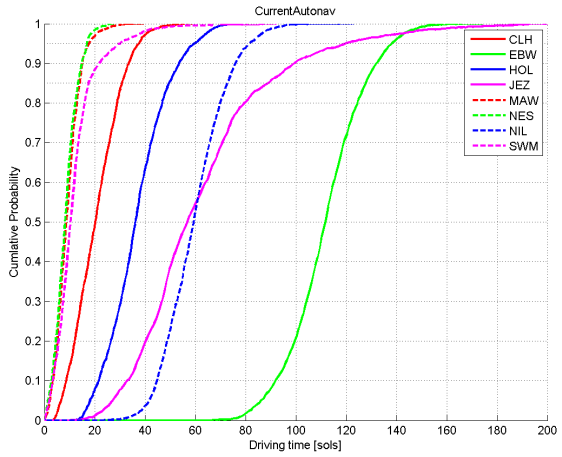


Figure 19: Cumulative distribution function of the required driving time for the eight candidate sites.

genet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[10] M. Golombek, A. Huertas, D. Kipp, and F. Calef, "Detection and characterization of rocks and rock size-frequency distributions at the final four mars science laboratory landing sites," *Mars*, vol. 7, pp. 1–22, 2012.

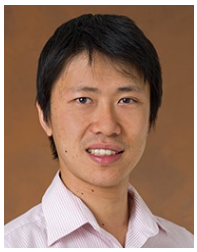
[11] A. Huertas, Y. Cheng, and R. Madison, "Passive imaging based multi-cue hazard detection for spacecraft safe landing," in *Aerospace Conference, 2006 IEEE*. IEEE, 2006, pp. 14–pp.

[12] P. Rosin, "The laws governing the fineness of powdered coal," *J. Inst. Fuel.*, vol. 7, pp. 29–36, 1933.

[13] M. P. Golombek, A. Huertas, J. Marlow, B. McGrane, C. Klein, M. Martinez, R. E. Arvidson, T. Heet, L. Barry, K. Seelos, D. Adams, W. Li, J. R. Matijevic, T. Parker, H. G. Sizemore, M. Mellon, A. S. McEwen, L. K. Tamppari, and Y. Cheng, "Size-frequency distributions of rocks on the northern plains of mars with special reference to phoenix landing surfaces," *Journal of Geophysical Research: Planets*, vol. 113, no. E3, pp. n/a–n/a, 2008, e00A09. [Online]. Available: http://dx.doi.org/10.1029/2007JE003065

[14] C. H. Acton, "Ancillary data services of nasa's navigation and ancillary information facility," *Planetary and Space Science*, vol. 44, no. 1, pp. 65–70, 1996.

BIOGRAPHY



Masahiro Ono *is a Research Technologist in the Robotic Controls and Estimation Group. He is particularly interested in risk-sensitive planning/control that enables unmanned probes to reliably operate in highly uncertain environments. His technical expertise includes optimization, path planning, robust and optimal control, state estimation, and automated planning and scheduling. Before joining JPL in 2013, he was an assistant professor at Keio University. He earned Ph.D. and S.M. degrees in Aeronautics and Astronautics as well as an S.M. degree in Technology and Policy from MIT, and a B.S. degree in Aeronautics and Astronautics from the University of Tokyo.*

Brandon Rothrock *is a Research Technologist in the Computer Vision Group. His research focus includes machine learning, compositional models, and semantic perception. He obtained his Ph.D. in Computer Science from UCLA, and a B.S. in Aeronautics and Astronautics from the University of Washington.*

Eduardo Almeida *is a Research Technologist in the Maritime and Aerial Perception Systems Group at the Jet Propulsion Laboratory. The focus of his research includes image-based 3-D reconstruction, structure from motion, probabilistic models and pattern recognition. He obtained his Ph.D. degree in Engineering as well as an M. Sc. degree in Applied Mathematics and an M. Sc. degree in Engineering, all from Brown University. He also holds a B.S. degree in Electrical Engineering from Federal University of Ceará (Brazil).*

Adnan Ansar *Adnan Ansar is a Member of Technical Staff in the Computer Vision group. He received his Ph.D. in Computer Science from the GRASP Laboratory at the University of Pennsylvania in 2001. His research resulted in a novel linear solution of the n-point/n-line pose estimation problem with several advantages over traditional methods and with direct application to use in augmented reality tasks. He also worked on stability issues for uncalibrated structure from motion in the context of critical surfaces. Prior to his switch to Computer Science, he was a Ph.D. candidate in Mathematics specializing in Differential Geometry. Since starting at JPL in 2002, he has worked on calibration (including development of an unsurveyed calibration routine), stereo, landmark detection and recognition, and the pose estimation aspect of various pinpoint landing tasks.*

Richard Otero

Andres Huertas *Andres Huertas was born in Colombia. In the 70s he studied Systems and Computer Engineering where he developed an interest in Artificial Intelligence. In 1976 he became a student at USC in Los Angeles, where he studied Computer Science and Electrical Engineering. As a student at USC he worked at the Image Processing Institute and upon graduation, at the Institute for Robotics and Intelligent Systems. His work since 1978 dealt with computer vision research, sponsored by the 20-year DARPA Image Understanding Program. The systems he developed in LISP to map cultural features from large single, pairs, and multiple aerial images were ported to several national labs and industrial sites for testing and evaluation. In 2002 he joined the Computer Vision Group at the Jet Propulsion Laboratory to work on computer vision techniques for autonomous off-road navigation, and on techniques for hazard detection for safe landing of spacecraft on planetary surfaces.*

Matthew Heverly *Matt Heverly is is the end to end mobility systems engineer for the Mars 2020 program. He holds a B.S. In Mechanical Engineering from Cal Poly San Luis Obispo as well as a M.S. In Mechanical Engineering from Boston University.*