Masahiro Ono, Brandon Rothrock, Kyohei Otsu, Shoya Higa, Yumi Iwashita, Annie Didier, Tanvir Islam, Christopher Laporte, Vivian Sun, Kathryn Stack, Jacek Sawoniewicz, Shreyansh Daftry, Virisha Timmaraju, Sami Sahnoune, Chris A. Mattmann
Jet Propulsion Laboratory, California Institute of Technology
ono@jpl.nasa.gov

Olivier Lamarre
University of Toronto

Sourish Ghosh, Dicong Qiu
Carnegie Mellon University

Shunichiro Nomura, Hiya Roy
University of Tokyo

Hemanth Sarabu
Georgia Tech

Gabrielle Hedrick
West Virginia University

Larkin Folsom, Sean Suehr, Hyoshin Park
NC A&T State University

***Abstract*— MAARS (Machine leaning-based Analytics for Automated Rover Systems) is an ongoing JPL effort to bring the latest self-driving technologies to Mars, Moon, and beyond. The ongoing AI revolution here on Earth is finally propagating to the red planet as the High Performance Spaceflight Computing (HPSC) and commercial off-the-shelf (COTS) system-on-a-chip (SoC), such as Qualcomm's Snapdragon, become available to rovers. In this three year project, we are developing, implementing, and benchmarking a wide range of autonomy algorithms that would significantly enhance the productivity and safety of planetary rover missions. This paper is to provide the latest snapshot of the project with broad and high-level description of every capability that we are developing, including scientific scene interpretation, vision-based traversability assessment, resource-aware path planning, information-theoretic path planning, on-board strategic path planning, and on-board optimal kinematic settling for accurate collision checking. All of the onboard software capabilities will be integrated into JPL's Athena test rover using ROS (Robot Operating System).**

TABLE OF CONTENTS

1. INTRODUCTION 1
2. MAARS VISION AND OVERVIEW 2
3. DRIVE-BY-SCIENCE CAPABILITY 3
4. RESOURCE-AWARE LOCAL PATH PLANNING CAPABILITY 5
5. RESOURCE-AWARE STRATEGIC PLANNING AND SCHEDULING CAPABILITY 8
6. DBS EXPERIMENT WITH SCIENTISTS 10
7. INTEGRATION WITH ATHENA ROVER 12
8. HPSC/SNAPDRAGON DEPLOYMENT 12
9. CONCLUSIONS 13
ACKNOWLEDGMENTS 13
REFERENCES 13
BIOGRAPHY 15

1. INTRODUCTION

The main roadblock to a Mars exploration rollout is that the best computers are on Earth, but the best data is on Mars. High-Performance Spaceflight Computing (HPSC) - a new generation of radiation-hardened (RAD-hard) multicore processor qualified for space - is currently being developed by NASA and the Air Force, and would enable a vast suite of new mission concepts [1]. The HPSC chiplet integrates three subsystems: the High Performance Processing Subsystem (HPPS) consisting of two clusters of Quad Cortex A53 ARM CPUs, the Real Time Processing Subsystem (RTPS) consisting of dual lockstep R52 ARM, and the Timing, Reset, Health Controller (TRCH), consisting of Triple Modular Redundant low power ARM M4F core. The algorithms presented in this paper are intended for the HPPS susbsystem. In the meantime, the Mars Helicopter Scout (Figure 1), the first vehicle to fly on Mars, uses Qualcomm's Snapdragon system-on-a-chip (SoC) for visual navigation[2]. The computation power of such modern commercial off-the-shelf (COTS) SoCs for mobile devices far surpasses the existing spacecraft computers such as the RAD750. For example, the Snapdragon 855 SoC has the ability to run deep neural networks in real-time with the support of its graphics processing unit (GPU), its digital signal processor (DSP), and its AI processor (AIP)[1].

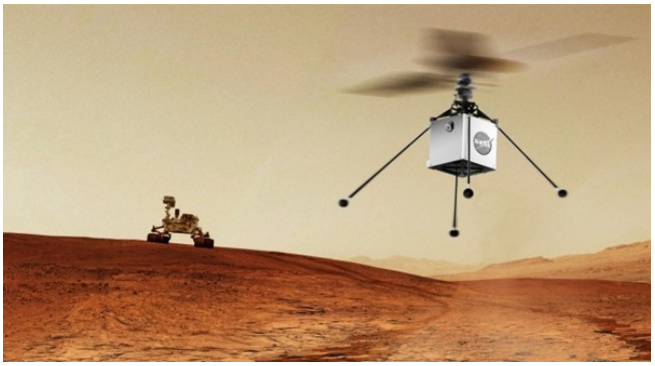


Figure 1: An artist's concept of the Mars Helicopter Scout (MHS), which will piggyback a ride onboard the Mars 2020 Rover to fly in the Martian atmosphere for the first time in history. MHS is pioneering the use of modern system-on-a-chip (SoC) for on-board autonomy.

There is an urgent need for significantly enhancing on-board autonomy of future rover missions. For example, the sample fetch rover of the Sample Retrieval and Lander (SRL) mission concept is expected to drive up to $\sim$ 1 km per sol, more than a ten-fold extension of the average per-sol driving distance of the Curiosity rover. Faster driving generates data at an increased rate (e.g., navigation images taken at a constant

[1]Note the COTS processors are distinct from HPSC, therefore direct comparisons cannot be made. Rather, this paper focuses on the development of autonomy algorithms that exploits the increased computational resources delivered by these processors, as well as the testing of them on multiple processor options.

978-1-7281-2734-7/20/$31.00

Figure 2: Unnoticed green monster problem - a metaphor that a rover with extended driving ability but limited intelligence may miss out on scientific opportunities.

interval), while the capacity of Mars-Earth communication remains limited by the laws of physics as well as the availability of relay orbiters and the Deep Space Network. For example, the downlink capacity from the Curiosity rover to Earth is typically $\sim$ 500 Mbit ($=\sim$ 60MB) per Sol while data-intensive instruments, such as hyperspectral imagers and ground-penetrating radars, can easily produce hundreds of megabytes to gigabytes of data. Such situations may result in the "unnoticed green monster problem[2]" (Figure 2) [3], meaning that science opportunities may be passed up by necessity or missed entirely simply because the data cannot be fully downlinked to Earth. Furthermore, since the majority of the driving distance will need to be covered by AutoNav (autonomous navigation), complex safety assessments that are currently performed on the ground must be performed on-board the rover. Solar-powered rovers, such as the one being considered for a potential Mars Sample Return, have larger uncertainties in their energy budgets compared to RTG-powered rovers. This necessitates proactive energy management as well as on-board prediction of energy generation and consumption, which have previously been performed on the ground during the Mars Exploration Rover missions.

Given these challenges, JPL's Machine learning-based Analytics for Rover Systems (MAARS) project aims at developing autonomy software capabilities that would significantly enhance the safety, productivity, and cost efficiency of future Mars rovers by fully exploiting the computation power of HPSC and modern SoCs such as Snapdragon. In particular, MAARS focuses on the following two capabilities:

1) Drive-By Science (DBS)
DBS refers to a capability to analyze data, in particular the engineering images acquired for autonomous driving, to assist ground scientists to detect interesting science features and selectively download relevant data without interrupting drives. With DBS, we can convert an engineering rover that does not have dedicated science instruments, such as the Sample Fetch Rover, to a scientifically valuable one by allowing scientists to instruct the rover to find geological features through its engineering cameras.

Our approach to realizing DBS is multi-fold. We use machine learning to extract a compact representation of collected data, such as feature vectors or image captions, which can be used as a summary of the data for downlink. Alternatively, the rover can perform data triage with criteria specified by scientists. Furthermore, it allows the rover to autonomously plan a path, select activities, and target the instruments to achieve prescribed science goals with a significantly fewer number of ground-in-the-loop cycles.

2) Risk- and Resource-aware AutoNav
The traversability assessment of the conventional AutoNav solely depends on terrain topography obtained through stereo vision, where the path objective is limited to trips between waypoints. We incorporate additional risk factors, such as terrain type and slip, as well as resource constraints, such as the energy budget, into consideration when planning a path. Combined with on-board data interpretation using machine learning, it will allow future rovers to plan a long-range path with diverse objectives, including driving energy minimization and science gain maximization, while increasing the level of safety.

What is described in this paper is the snapshot of our development effort at the end of the second year of the three-year project.

2. MAARS Vision and Overview

MAARS is not a monolithic system but a collection of multiple autonomy algorithms, as illustrated in Figure 3. Each capability is designed to be modular; that is, although all the algorithms are integrated together through the Robot Operating System (ROS), each individual capability can be easily adapted for different missions. Also, unlike what the name of the project indicates, some of the MAARS algorithms do not involve machine learning. Machine learning however remains a central theme of the project. The fact that MAARS is a homonym of Mars is not an indication of our exclusive interest in the particular planet but simply a reflection of the sense of humor of the team members. In fact, most MAARS algorithms are applicable to rover missions in any world with sufficient gravity for wheeled mobility. Finally, we note that the apparent resemblance of algorithm names to Star Trek characters, such as SCOTI, SPOC, RAND, and VeeGer, is purely coincidental.

MAARS covers nearly all aspects of on-board autonomy that would be needed for future rover missions, including perception, planning, and automated science. In the following three sections, high-level overviews of all the algorithms developed and investigated under this project are provided. Section 3 describes algorithms related to the DBS capability, including automated image captioning (SCOTI), image similarity search, on-board data prioritization, and the ground-based interface for efficiently accessing and searching the on-board database. The risk and resource-aware AutoNav capabilities are described in Sections 4 and 5. The algorithms related to local planning, including the vision-based driving energy estimation (VeeGer) and on-board kinematic settling for collision checking (OBKS) are included in Section 4, while the algorithms related to global planning, including the ground-based energy-aware path planning, information-theoretic path planning, and on-board strategic path planning (RAND), are

[2]This is a terminology coined by us.

introduced in Section 5. Section 6 reports the experiment we performed to test the DBS capability and section 7 describes the overall software architecture and the software integration into JPL's Athena test rover. Finally, Section 8 briefly summarizes our deployment and benchmarking effort on an HPSC emulator and Qualcomm's Snapdragon SoC.

3. DRIVE-BY-SCIENCE CAPABILITY

Overview

The objectives of drive-by-science are to autonomously downselect a subset of images taken by the rover's navigation or engineering cameras over the course of an autonomous drive such that the science value of the selected images is maximized. The volume of downselected images is limited by the rover's downlink bandwidth. The notion of science value is of course difficult to objectively define; our approach characterizes this as a combined relevance score produced by a trained model to select images according to a specification provided by mission scientists for a given planning period. This specification is provided in natural language using relevant geologic vocabulary. We present a model called SCOTI (Scientific Captioning Of Terrain Images) that automatically generates captions of Mars surface imagery, trained on expert annotations from Mars geologists. We also incorporate the ability to specify image similarity using a distance measure on extracted image features, which can also be used as a measure of image diversity. The captioning and similarity models are packaged into a content-based search system that is the primary interface for how scientists will interact with the rover. Scientists can provide a set of text queries, similarity examples, diversity criteria, filters, and more for what imagery they expect to be of scientific interest for the next planning cycle. This specification is then uploaded to the rover, which autonomously downselects a set of images taken during a given drive.

SCOTI: Scientific captioning of terrain images

SCOTI takes images as an input and outputs a English sentence explaining the geological content of the image. It is built upon the Show, Attend, and Tell model [4]; it is an attention-based model that automatically learns to describe the content of images. The model is trained in a deterministic manner using standard backpropagation techniques with stochastic gradient descent using adaptive learning rate algorithms. The encoder part of the model uses a convolutional neural network for extracting a set of feature vectors, also known as annotation vectors. The decoder part of the model uses a long short-term memory (LSTM) network that produces a caption by generating the words sequentially conditioned on context vector as well as the previous hidden state. The VGG19 pre-trained on ImageNet is used as the feature extractor. However, in principal, other encoding functions could also be used. Regularization strategy and model selection are adopted based on BLEU scores. The LSTM network is trained from scratch with 1,000 image captions created by human geologists. It achieved a BLEU-4 score of 0.85 on validation sets. Sample outputs from SCOTI are posted in Figure 4.

SCOTI enables an interplanetary Google image search. The ground scientists express their interest in natural language words, which are uplinked to the rover. Captions from SCOTI in the on-board database are compared against the query words for data prioritization. Alternatively, SCOTI can be used as a data summarization capability, where the ground scientists use the downlinked captions to decide which images should be downlinked at full resolution. There is a ground application, too: currently, scientists needs to manually inspect tens of thousands of rover images to find images with specific geological features such as veins, nodules, crossbedded layers and more. SCOTI allows them to find images of interest through a text-based search like Google search.

Image Similarity Search

We also developed an image similarity search capability, which returns a list of images that are qualitatively similar to a query image provided by users. This ability is enabled by extracting an intermediate (Conv5_3) layer from the VGG19 image encoding portion of SCOTI and compare the feature vector against a feature vector of other images using the cosine distance metric to find the most similar or dissimilar (in the case of novelty search) images.

Like SCOTI, this capability could be used for on-board data prioritization and ground-based data search. For the on-board application, the query image is first converted into a feature vector, which is much smaller in terms of data size. The uplinked feature vector is compared against the feature vectors of on-board images to prioritize them. The ground-based similarity search provides an additional way of quickly finding images of interest in image datasets.

On-board data prioritization— A future rover can easily collect gigabytes to terabytes of data (e.g., high-resolution images, hyperspectral images, ground-penetrating radar observations) over a single operation cycle. However, it may not be able to downlink all the raw data due to the limitation in communication bandwidth. Data prioritization attempts to identify a subset of the data that contain features relevant to the query by scientists on the ground but contain diversity with respect to the interesting feature's representation. Moreover, the total data volume of the subset cannot exceed the available downlink capacity. This is an initial heuristic approach at quantifying the expected scientific value that geologists will receive from downlinking the image set. The goal of the data prioritization problem is to maximize the relevant dissimilarity amongst all pairs of data within the selected set. While this basic idea applies to any kind of data, in this project we mainly focused on images. We used NAVCAM image data from the Curiosity rover as the dataset.

The image prioritization problem was formulated as a quadratic knapsack problem as we are maximizing the value of the image set (knapsack) through selecting images based on pairwise relevant dissimilarity (quadratic decisions) and the set is subjected to the downlink capacity. The objective function of the model includes an adjustable hyper-parameter. This hyper-parameter allows geologists to increase the similarity or dissimilarity of images within the set. The intention of this feature is to give geologists the ability to tune their queries or to allow the system to provide recommendations to how similar/dissimilar they prefer the images to be. The feature assisted in providing geologists image sets are generated by the model for qualitative comparison, however, the quadratic knapsack model is difficult to solve.

The quadratic knapsack problem is NP-Hard, which indicates that finding an exact solution is computationally expensive. The initial approach to solving this problem was implemented using Gurobi's Gurobipy package for python along with varying the program's run-time. In addition, a greedy quadratic-knapsack heuristic and a minimum spanning tree heuristic

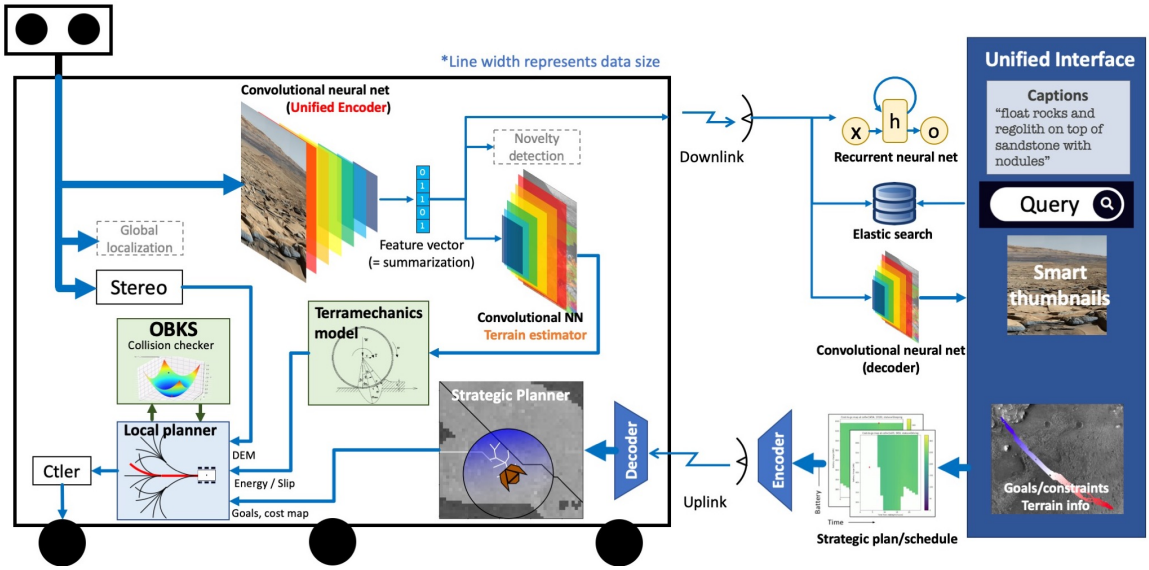


Figure 3: Overview of the algorithms developed under the MAARS project

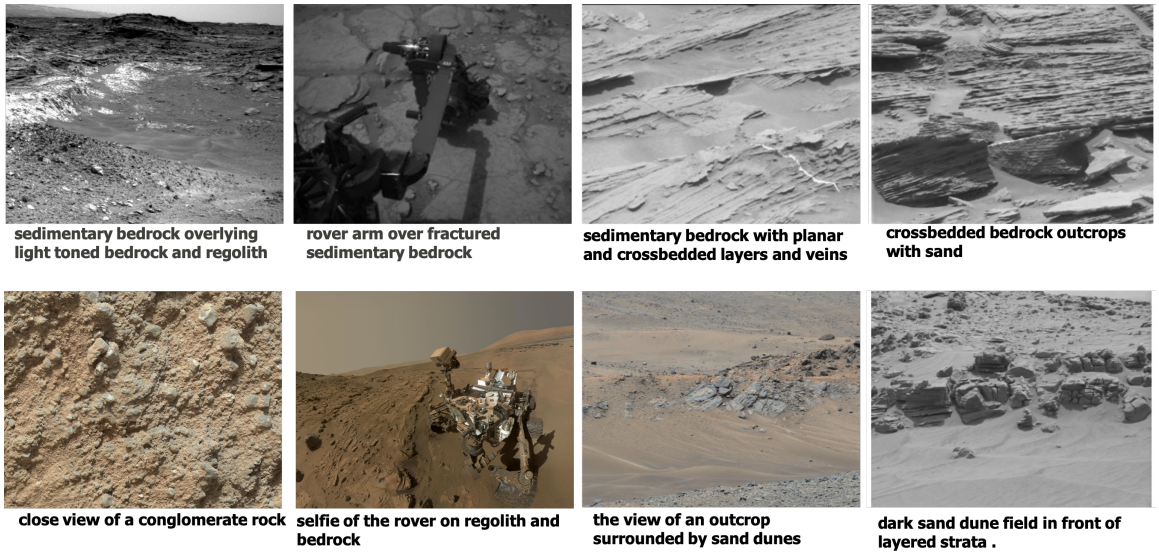


Figure 4: Sample outputs from SCOTI, which generates a natural language sentence describing a given image.

were implemented for comparison of computational time and objective function value. These approaches are initial attempts at solving the computationally intensive problem; other future approaches that are not implemented in this problem are defining the problem with a sub-modular objective function or implementing a dynamic programming heuristic.

PDS pipeline

On the Earth, a geologist must manually search thousands of images in the Planetary Data System Cartography and Imaging Sciences Node (PDS) to find Mars images relevant to their research. To provide a more efficient search experience, we have integrated SCOTI and image similarity search into a content-based search system. SCOTI model is deployed on a GPU-enabled PDS server through docker, with a direct mount to the PDS MSL NavCam images. The resulting model captions and image feature vectors are stored in an ElasticSearch database, along with metadata such as the sol, and latitude and longitude at which the image was taken. ElasticSearch was originally designed for fast text-based searches, but since its inception, it has been expanded to include features that allow users to search by geolocation, time, number, etc. Using a custom plugin that calculates the cosine distance between two vectors, we add the capability to search for similar images as well [5]. Combining these attributes in one search engine enables scientists to quickly find data by geological feature within the desired location or sol range, and to search for "more images like this."

DBS Interface

To enable any scientist or rover operator to interact with all of these DBS capabilities, we have built a web application (Figure 6) that implements a Google Maps-like search interface for the Martian surface explored by MSL. To start

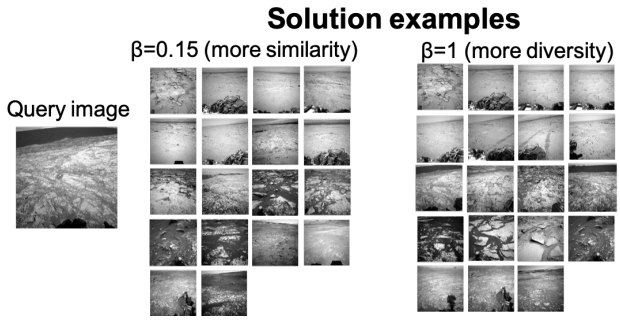


Figure 5: A sample result of the onboard data prioritization algorithm. The algorithm returns a subset of data that balances the relevance between the query and the diversity within the subset. This balance can be tuned by a parameter β.

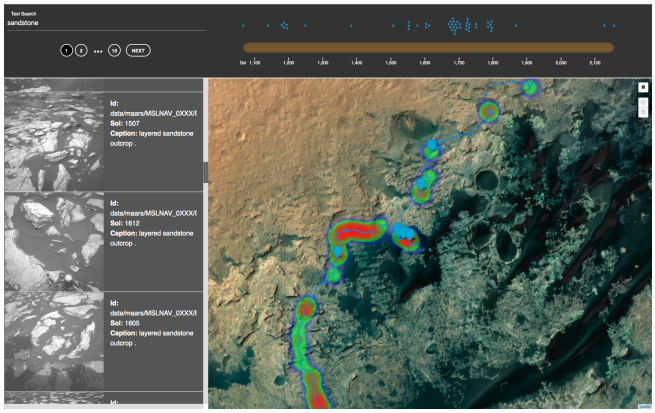


Figure 6: All our DBS capabilities are made available to scientists and operators through a web-application.

with, an user may perform a text search on the SCOTI-generated image captions after which the paginated side-bar displays the query results by showing the picture, the generated caption, and other relevant metadata. Additionally, the geographic locations of results on the current page are marked by pins on orbital satellite imagery, while the geographic distribution of all results is visualized using a dynamic heat map. If the user is interested in specific geographic regions, a drawing tool allows them to specify areas of interest and only search and see results within those regions. We also integrate the image similarly search capability. Once an image is selected, a simple click of a button allows a user to see similar images displayed in the same way as the text query results. Ultimately, making these capabilities available through a centralized user-friendly application greatly helps scientists make the best use of the vast trove of Martian images to make quicker and optimally-informed decisions.

4. Resource-aware Local Path Planning Capability

Overview

The local path planning capability of MAARS is provided by a combination of several new and existing algorithms. On the perception side, a new driving energy prediction algorithm called *VeeGer* is proposed that combines the SPOC terrain classifier [6] and obstacle detection with stereo vision. The resource awareness comes from VeeGer, where the estimated

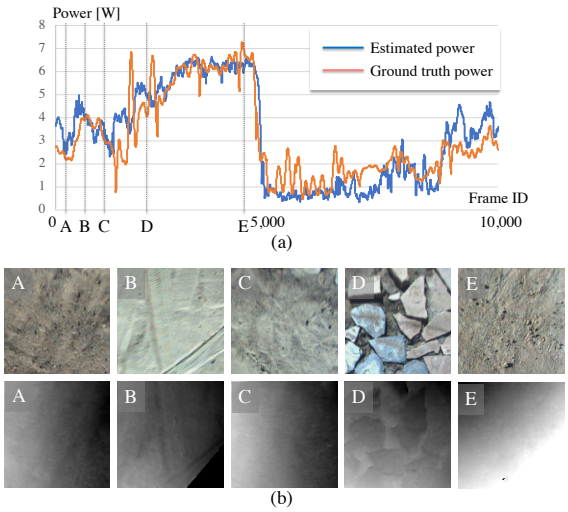


Figure 7: (a) Energy consumption estimated from terramechanics parameters predicted with Veeger-TerramechanicsNet, (b) RGB images (top raw) and depth images (bottom) at time A∼E in (a).

energy usage is used as a part of the cost function of the path planner. On the path planning side, a standard tree planner is combined with collision checking. Users can choose from three collision checking algorithms: ACE (approximate kinematic settling) [7], which will be used by Mars 2020 Rover with a collision checking algorithm, the probabilistic extension of ACE called p-ACE [8], and optimization-based kinematic settling (OBKS). Alternatively, a meta algorithm called CAAPS (Context - Aware Adaptive Policy Selection) is also proposed that can autonomously choose the collision checking algorithm which works best in the given environment. This section only describes new algorithmic components: VeeGer, p-ACE, OBKS, and CAAPS.

Vision-based Driving Energy Prediction (VeeGer)

We proposed vision-based algorithms to remotely predict the driving energy consumption using machine learning [9]. Specifically, we develop and compare two machine-learning models in this paper, namely VeeGer-EnergyNet and Veeger-TerramechanicsNet, respectively. The former is trained directly using recorded power, while the later estimates terrain parameters from the images using a simplified-terramechanics model, which explained in the following section and calculate the power based on the model. The two approaches are fully automated self-supervised learning algorithms.

Frontend: CNN—To combine RGB and depth images efficiently with high accuracy, we propose a new network architecture called Two-PNASNet-5, which is based on PNASNet-5. Figure 7 shows an example result of predicting power by Veeger-TerramechanicsNet. A comparison of the two approaches showed that Veeger-TerramechanicsNet had better performance than VeeGer-EnergyNet.

Backend: Terramechanics model—In VeeGer-TerramechanicsNet, a simplified terramechanics model was used for both to estimate the terramechanics parameters and to calculate the driving energy of the rover. To train the VeeGer-TerramechanicsNet, simplified terramechanics parameters were estimated by optimizing the following model under the condition in which the wheel load is calculated based on the simplified terramechanics model and estimated wheel load

balance: [9]:

$$\begin{aligned} \min_{k_{normal},n,k_{shear},z_s,z_d} \quad & \left[(1-F_z/W)^2+(1-M_y/T)^2\right] \\ \text{s.t.} \quad & 0.0 \le k_{normal} \le 9.0e^5 \\ & 0.1 \le n \le 1.0 \\ & 0.0 \le k_{shear} \le 1.0 \\ & 1e^{-4} \le z_s \le 2(r+h)/3 \\ & 0.0 \le z_d \le 0.0015 \end{aligned} \tag{1}$$

where W, T, F_z, and M_y are an estimated wheel load, measured wheel torque, calculated wheel load, and calculate wheel torque based on the simplified terramechanics model, respectively. Moreover, $k_{normal}, n, k_{shear}, z_s, z_d$ are simplified terramechanics parameters used to calculate F_z and M_y. Preliminary estimated terramechanics parameters were mapped with RGB and depth images based on the timestamps. More details on VeeGer is described in [9].

Probabilistic approximate collision check

Accurate kinematics-based collision detection on Mars rovers requires computationally intensive iterative solvers with geometric constraints and they are typically unstable on rocky terrain. Due to computational constraints on slow spacecraft computers, such as RAD750 (used by the Curiosity rover and the upcoming Mars 2020 rover), a lightweight body-terrain clearance evaluation algorithm (called ACE [10]) has been developed for the automated path planning of the Mars 2020 rover. ACE obtains conservative min-max bounds on vehicle clearance, attitude, and suspension angles *without* iterative computation by estimating the lowest and highest heights that each wheel may reach given the underlying terrain, and calculating the worst-case vehicle configuration associated with those extreme wheel heights. The conservative bounds guarantee safety during autonomous navigation. However, ACE's conservative safety check approach can sometimes lead to over-pessimism: feasible states are often reported as infeasible, thus resulting in frequent false positive collision detection (Figure 8). In order to reduce the over-pessimism, a computationally efficient probabilistic variant of ACE (called p-ACE [8]) has been developed, which aims to relax the hard constraints imposed by ACE.

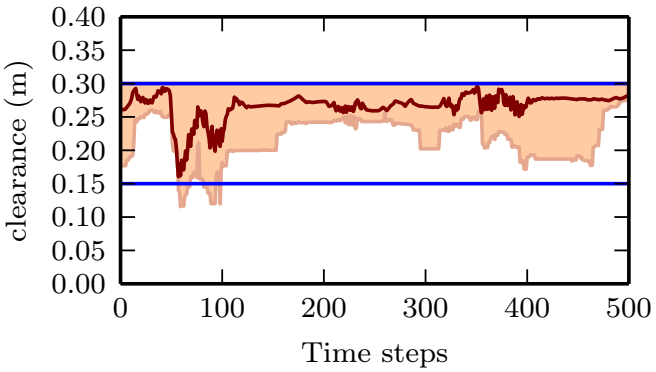


Figure 8: ACE estimation results of body clearance. The red line corresponds to the ground-truth state value and the shaded region represents the intervals between the ACE bounds. The blue lines indicate the allowable state range.

Given a target rover pose $\mathbf{x} = (\mathbf{x}, \mathbf{y}, \psi)$ and a terrain model m, p-ACE estimates the probability of constraint satisfaction of the remaining states $\omega \in \mathbb{R}^7$ (clearance, attitude, and suspension angles) as:

$$P(\omega \in \Omega | \mathbf{x}, \mathbf{m}) = \int_{\omega \in \mathbf{\Omega}} \mathbf{p}(\omega | \mathbf{x}, \mathbf{m}) \mathbf{d}\omega \tag{2}$$

p-ACE estimates the constraint satisfaction probability (2) in response to queries from the path planner. The probability distributions are characterized offline by running Monte Carlo simulations on the ROAMS (Rover Analysis, Modeling, and Simulation) [11], [12] simulator. p-ACE can be used by planners for risk-aware motion planning. The collision risk of a path $X = \{\mathbf{x_0}, \mathbf{x_1}, \cdots, \mathbf{x_k}, \cdots\}$ can be defined as:

$$R(X) \equiv \max_k P(\omega \notin \Omega | \mathbf{x_k}) \tag{3}$$

The path X is collision free if $R(X) < r$, where r is the *risk-factor* threshold. Figure 9 shows the extension of a binary collision map (ACE) to a probabilistic collision map (p-ACE). The advantage of having probabilistic collision maps is that it helps to quantify risk, which can be taken into account by path planners. Figure 10 compares the success of motion planning on simulated Mars terrains with ACE and p-ACE. Success here means the ability for the motion planner to generate a valid path from the start pose to the end pose. CFA $\geq$ 7% deterministic ACE has 0% success rate, whereas p-ACE with a risk-factor of 1e-03 had around 50% success rate. Even with a small relaxation of the worst-case bounds in ACE, the success rate of path planning is drastically increased. This is a clear indication of the reduced pessimism involved with p-ACE.

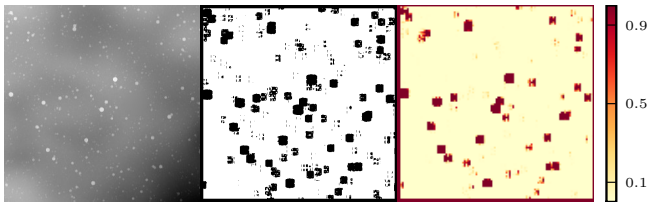


Figure 9: Extension of a binary collision map (ACE) to a probabilistic collision map (p-ACE). **Left**: DEM. **Middle**: ACE collision map in C-space (*black* = inaccessible, *white* = accessible). **Right**: p-ACE collision map drawn as a heatmap in C-space. The rock abundance is set to 5% CFA (cumulative fraction of area). For all rover poses, the yaw angle is fixed to $\psi = 0$.

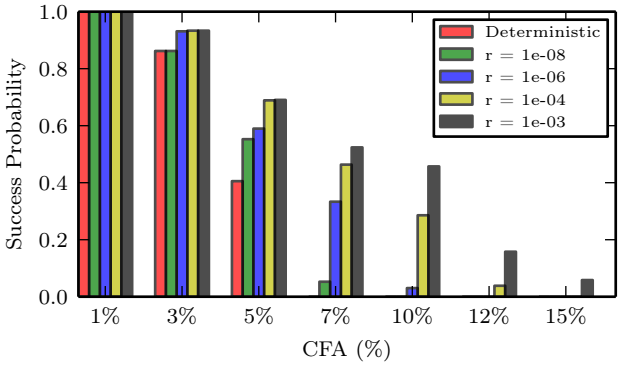


Figure 10: A comparison of the probability of success of risk-aware path planning using ACE and p-ACE at various risk-factors (r) and CFAs. Rocky 8 rover model was used for this comparison.

On-board Optimization-based Kinematic Settling (OBKS)

The use of approximate clearance techniques unsurprisingly introduces varying levels of conservatism into pose prediction and clearance evaluation algorithms [10]. When used in path planning for collision-checking, ACE is seen to perform better than the state-of-the-art algorithm GESTALT [13], [10] that rejects all paths, including feasible ones that allow the rover to straddle small obstacles and traverse over undulating terrain. By determining a tighter interval on possible states that the rover can occupy, ACE can reduce this conservatism. It is important to note that the conservatism manifests as false-rejections of candidate paths when used in a motion-primitive path planner. To reduce the rate of false-rejections, a more sophisticated algorithm is necessary such that the confidence interval may be further reduced.

HPSC and COTS processors open up a variety of avenues to explore, owing to the increased computational capabilities available. We developed an iterative algorithm for onboard computation that is very much in the vein of those used in ground rover simulation software at JPL [11], [12]. The algorithm named Optimization-Based Kinematic Settling (OBKS) solves a local optimization problem to minimize contact between terrain and rover wheels. We modeled it as a least-squares problem subject to pose constraints on joint angles as determined by rover design limits. As we expect the solution yielded by OBKS to be close to the exact pose of the rover for a given location in a heightmap, the *interval* of uncertainty is smaller hence, reducing the intrinsic conservatism of the path planner in comparison with ACE.

Although the time per query of ACE is lower than that of OBKS (seen in Figure 12), the increase in success rate during random placement (seen in Figure 11) allows the time taken to generate a path for a 20m traversal to be significantly lower for OBKS in relatively complex terrain (CFAs greater than 7%). Due to lower conservatism, the OBKS planner is also able to generate paths with lower path inefficiency (seen in Figure 15). We implemented this algorithm on an Nvidia Jetson TX2 board using Ceres solver [14] for Athena Rover (see Section 7) and runs with a mean query time of 13.7 ms.

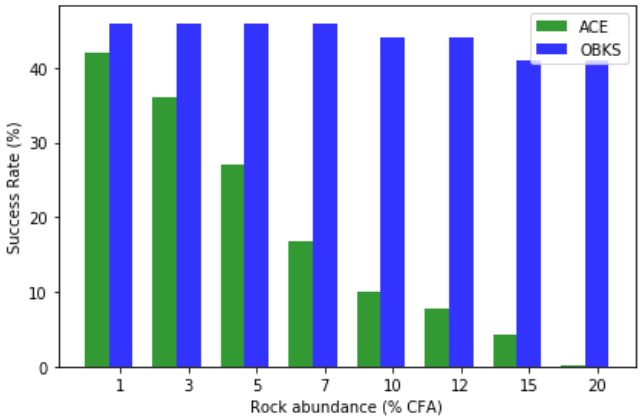


Figure 11: Monte Carlo estimates of success rates of ACE and OBKS for random placement on height maps of varying CFA. A random placement is considered a success if the collision-checking algorithm returns that the location in the map is safe for traversal.

Meta-policy for Adaptive Planner Selection (CAAPS)

We developed a learning-based meta-algorithm to select the *optimal* planner for a given state of the rover and its environment. The learned policy must select a planner such that the energy or computation required to call the planner is mini-

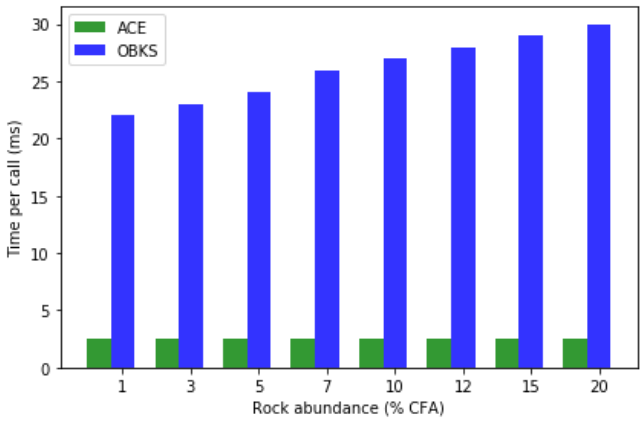


Figure 12: Monte Carlo estimates of query time for ACE and OBKS for random placement on height maps of varying CFA.

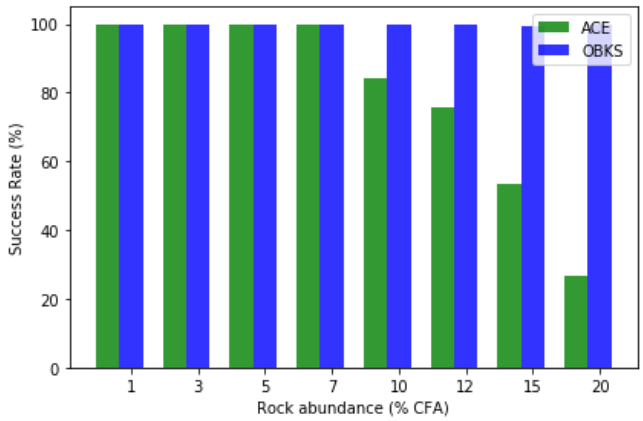


Figure 13: Monte Carlo estimates of success rates for an ACE-based planner and an OBKS-based planner for 20m traversals on height maps of varying CFA. A traversal is considered a success if the planner is able to generate a path from the start location to the goal location in the map.

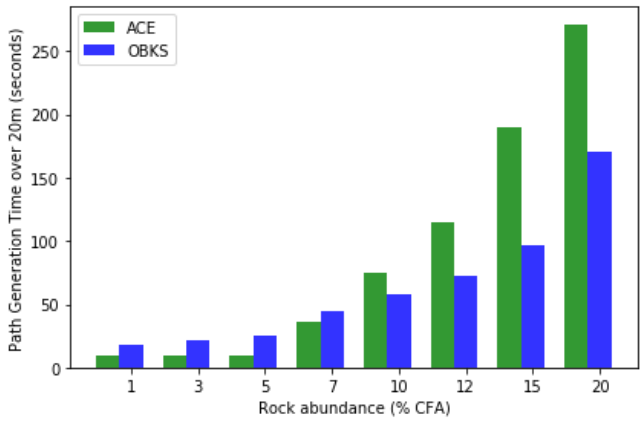


Figure 14: Monte Carlo estimates of path generation times for an ACE-based planner and an OBKS-based planner on height maps of varying CFA.

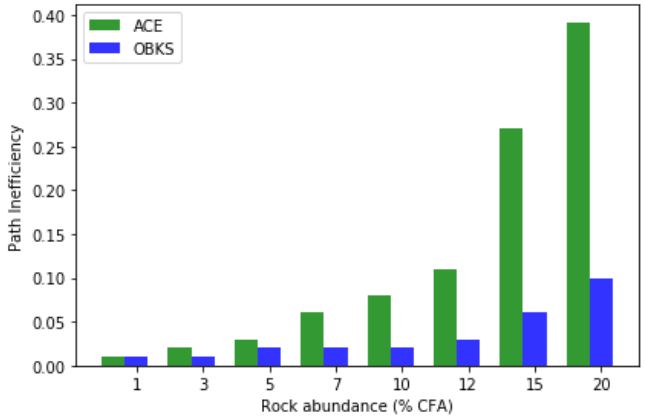


Figure 15: Monte Carlo estimates of path inefficiency for an ACE-based planner and an OBKS-based planner on height maps of varying CFA.

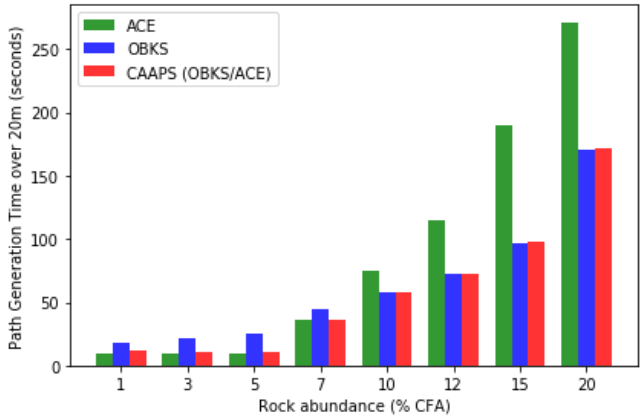


Figure 16: Comparison of path generation time for planner using OBKS, ACE and meta-policy selector that greedily switches between the two

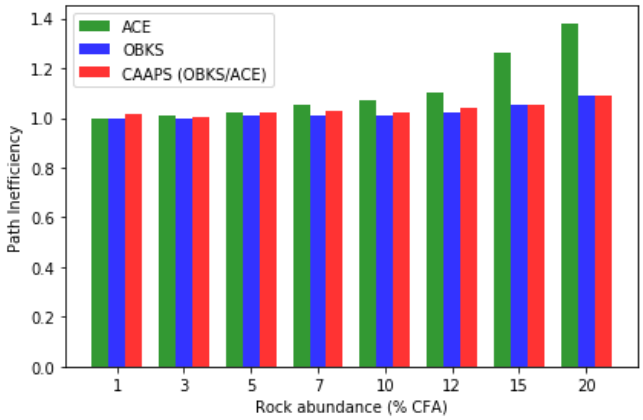


Figure 17: Comparison of path inefficiency for planner using OBKS, ACE and meta-policy selector that greedily switches between the two.

mized with a controlled trade-off in task performance. Development of OBKS (see summary in section 4) for onboard computation on future rover missions provided insight into the algorithm's strengths and weaknesses upon utilization on terrains of varying complexity. While in CFAs greater than 7% the arc-primitive path planner using OBKS outperformed the same planner using ACE for collision-checking, the latter demonstrated an advantage in path generation time with no detriment to path inefficiencies for lower CFAs. We formulate the problem such that the two planners are regarded as experts, their path generation times as computation-related performance measures we mean to minimize whilst keeping the probability of success greater than a user-defined threshold. Formally, given a map M, planners $\Pi = \{\Pi_1, \Pi_2\}$,

$$\min_{\Pi_i \in \Pi} T(\mathcal{M}, \Pi_i) \tag{4}$$

$$\text{s.t } P(s) \geq \eta \tag{5}$$

where $T(\mathcal{M}, \Pi_i)$ is the expected time taken to produce a path by planner Π_i for map $\mathcal{M}$, $P(s)$ is the probability of successful path generation and η is the user-defined lower threshold.

To solve the problem posed above, we designed a simple policy selector and implemented it to predict path generation times and the probability of success for each planner. The selector model is a 4 layer convolutional neural network that operates on Depth Elevation Maps (DEM) maps as inputs. We trained the model using a supervised learning approach on over 7000 DEM maps spanning 30m x 30m and varying CFAs (1%, 3%, 5%, 7%, 10%, 12%, 15%, and 20%) to achieve mean test accuracy of 81% and 92% on success prediction and path time generation respectively. Preliminary results (summarized in Figures 16,17) indicate that the policy selector is able to select the optimal planner to minimize path generation time (and as a result, path inefficiency due to strong correlation between the two quantities).

5. Resource-aware Strategic Planning and Scheduling Capability

The existing strategic path planning process for Mars rovers consists of three subprocesses. First, traversability of the landing site, typically $\sim$ 10 x 10 km in size, is analyzed using the orbital data. In Mars 2020, slope, rock density, and terrain type are estimated for each of the 1 x 1 m grids of the landing site, which is turned to estimated driving speed. More details of this process are described in [15]. Second, a route is planned based on the traversability analysis. The strategic route has been planned mostly manually for the previous and current Mars rovers. For the landing site traversability analysis of the Mars 2020 Rover mission, we used an optimal route planner that plans the shortest-time path from any locations in the landing site to scientific regions of interest (ROIs) [15]. This path planning capability is integrated into Mars 2020's ground operation software. Third, during the day-to-day operation of the rover, the rover planner sets the waypoints and the goal for the next planning cycle based on the strategic route. When AutoNav is utilized, no-go zones and keep-in zones are set if needed. Besides the uplinked goal, waypoints, and no-go/keep-in zones, the on-board software of the rover is not aware of any information regarding the global environment. Hence, the rover is incapable of onboard replanning at the strategic scale.

MAARS proposes updates to all three subprocesses. As for the traversability assessment, we are experimenting with a new approach based on thermal inertia. As for the ground-

based route planning, we developed two new methods: energy-aware path planning and scheduling, which concurrently optimizes a path and activity schedule in a way that respects the energy constraint, and the information-theoretic path planning, which takes into account the knowledge uncertainty in orbiter-based traversability assessment as well as a supporting aerial vehicle (helicopter) that serves as a scout. Finally, as for the on-board plan specification, we developed a new approach called RAND, which compactly compresses the full plan and schedule of the entire state space, uplink them, and decompress on-demand on-board, such that the rover can replan anytime when it has to deviate from the pre-planned route or it experiences unexpected change in the battery level. We introduce each of these new approaches in the following subsections.

Traversability analysis with thermal inertia

Thermal inertia could be used for traversability analysis as it brings information about the surface and subsurface. It is derived from the Mars Odyssey Thermal Emission Imaging System (THEMIS) nighttime temperatures [16] and depends on several factors including particle size, degree of induration, rock abundance and exposure of bedrock at the subsurface (within a few centimeters of the surface). It translates the potential of a material to store heat during the day and release it at night [17] and is defined as follows:

$$I = \sqrt{k\rho c} \tag{6}$$

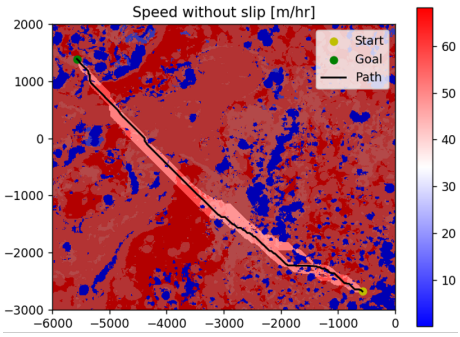
Where I is the thermal inertia in thermal inertia unit (TIU), k is the bulk thermal conductivity, ρ is the bulk density and c is the specific heat of the surface layer (up to a few centimeters below the surface [17]). In general, low thermal inertia values are associated with deep sand, leading to harder conditions for a rover to drive. High thermal inertia, however, translates into indurated material such as bedrock. This could imply that the rover would have less difficulty driving on such terrain. However, a dense rock field could also display high values of thermal inertia (rocks are indeed indurated material), which would be a harder terrain to drive on. It is thus necessary to pair this data set with other ground data (e.g., images) to draw meaningful conclusions and adequately assess traversability.

Energy-aware path planning and scheduling

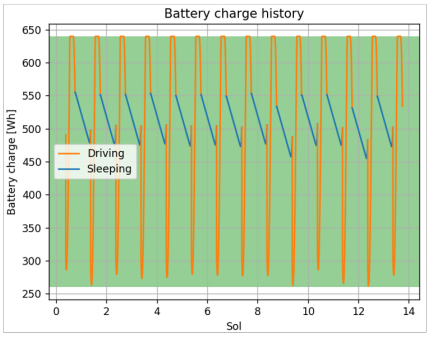
We developed a planning algorithm that enables a solar-powered rover to reach strategic goals in minimum time while respecting energy constraints. The algorithm concurrently optimizes the trajectory and schedule, where activities specified in the schedule include commands such as heater activation, drive start, and sleep. With the ability to explicitly optimize the rover's energy use, this algorithm extends its autonomy capability. For example, a schedule generated by the planner for a given Martian day (sol) might look like the following: Power on avionics and warm up motors in the morning. Traverse the planned trajectory while generating power during the day. Stop traversing and power off avionics in the evening. We consider constraints such as maintaining a battery charge over a given threshold while planning the path and schedule.

The algorithm takes terrain data and the strategic start and goal locations as inputs, and generates a "cost-to-go map." Once the rover receives the cost-to-go map, it can calculate the optimal action by a simple computation.

We formulated the problem as a Markov decision process (MDP). The state is defined by four elements: the mode (the driving mode or the sleeping mode), the cell (the location in the discretized map), the time, and the state of the battery charge. The action is either to change the mode, to drive to an adjacent cell, or to stay at the cell. The cost is defined by the action time, i.e., the travel time. A penalty cost is given if the battery charge lower bound is violated (i.e., battery depletion). Because the state space of the MDP is so large that it cannot solve in a reasonable time, we proposed three methods to reduce the state space: region reduction method, periodic approximation method, and function approximation method. The region reduction method reduces cells using the results of a simpler problem that only considers the location. The periodic approximation method assumes the cost-to-go function is approximately periodic in time dimension with a period of one sol. The function approximation method calculates the cost-to-go function in time and battery dimensions with a fitting function generated from a few sample point evaluations.



(a)



(b)

Figure 18: (a) The simulated route of the rover with the highlighted region generated by the region reduction method. (b) The battery charge history. The range from the minimum allowable charge to the maximum charge is shown in green.

Figure 18 shows the results of the demonstration of our algorithm in Jezero crater on Mars. The rover autonomously scheduled the driving and reached the destination without violating its battery charge lower boundary.

Information-theoretic path planning

We developed an algorithm that computes safe, travel-time-optimized routes for a rover in uncertain terrain using a helicopter scout. The probability distribution of the rover's travel time for each square meter of the surface was generated based on the mobility model developed by Ono et al. [18]. Inputs to the model are terrain type determined using the SPOC terrain classifier [6], slope from HiRISE digital elevation model (DEM) data, and rock abundance in terms of the Cumulative Fractional Area (CFA) covered by rocks. We assume that the uncertainty in the rover's travel time through a particular

location is reduced when the helicopter scout observes that location. Scouting a location in advance enables the rover to make adjustments to its planned route if it finds the conditions inside a nearby highly uncertain region more favorable, or if it finds the conditions along its current planned route are unfavorable.

Traditional uncertainty metrics such as Shannon Entropy have difficulty distinguishing between different bimodal distributions, or introduce a bias toward specific probability distribution shapes. Therefore we represented travel time uncertainty as the standard deviation of the travel time probability distribution. Areas of high standard deviation are efficiently explored through observation with the helicopter scout, reducing the travel time uncertainty for the rover, allowing an updated rover path plan. The helicopter scout is routed using a Markov Decision Process. A Rapidly Exploring Random Tree-Star algorithm (RRT*) [19] with travel-time-based cost and the ability to be "re-wired" enables computationally fast rover path planning and re-planning on a large scale, high-resolution environment.

We tested algorithm performance in two environments; a Monte Carlo simulation on a user-defined region, and a simulation using data from the Jezero region of Mars. In both environments, three scenarios were evaluated; rover alone, rover with helicopter navely scouting ahead on the planned route, and rover with the Information-Theoretic helicopter. The environment for the Monte Carlo simulation consists of two obstacles and three possible passages. Each passage was randomly assigned an expected speed, standard deviation, and "ground truth", which was revealed only after observation by the helicopter. The environment in the Jezero subregion consists of a hill with a central obstruction and two possible passages, each of which has uncertain travel time. Compared with the rover alone, results showed that when the helicopter naively scouts along the rover's planned path, the travel time can on average be improved slightly. More significant travel time improvements were obtained by flying the helicopter scout on the optimized route without increased risk for the rover. Benefits of this information-theoretic path planning are enhanced in environments with larger travel time uncertainty and multiple feasible routes for the rover.

Resource-aware, on-board strategic planning (RAND)

We have extended our efforts in global planning to increase the strategic autonomy of planetary rovers. To achieve this, we are currently developing the Resource-Aware planner for Non-stop Driving (RAND), which aims at deploying large resource-aware global plans onboard rovers. Traditionally, strategic planners (such as the energy-aware path planner and scheduler described previously) are only intended to be used on Earth, where data storage and large computational efforts are not a problem. By adapting them to online autonomous operations, RAND maximizes the driving time of planetary rovers on the surface by reducing their reliance on human-made navigation plans every sol.

RAND compresses raw and heavy strategic plans through Monte-Carlo simulations of kilometer-scale rover drives towards a distant goal location. Each simulation varies the rovers start state by randomly sampling the strategic state-space (in our case, the physical location, time of day, energy level and operational mode) to identify the most likely trajectories the rover might employ in the near-future. Once candidate trajectories are found, they are ordered based on a custom utility metric and the ones with the highest score are retained and uplinked to the rover. During ground operations, the network of trajectories continuously informs autonomous navigation operations at a strategic level. Figure 19 demonstrates preliminary results where a large search space on an orbital map of Jezero Crater was compressed down to three reference trajectories. In this case, the compressed plan required less than 1 megabyte of data as opposed to approximately 1 gigabyte of data in the raw strategic plan.

This algorithm is robust to varying communication bandwidths and explores an interesting trade-off: uplinking more data (i.e. more trajectories) will increase the strategic-level awareness of the rover and lead to optimal behaviors with generally low onboard computational loads. On the other hand, a very small amount of data will lead to larger computational efforts to recover strategic behaviors with similar optimality.

6. DBS EXPERIMENT WITH SCIENTISTS

To test our Drive-By-Science (DBS) capabilities, we conducted a "mock science mission," in which four scientists were asked to perform scientific tasks using the DBS capability. We put the subjects in a scenario simulating the conceptual sample fetch rover mission, where the rover collects significantly more images that can be sent. The subjects used DBS tools to downselect the images in a way to maximize the accuracy of scientific interpretation in the given tasks.

Methods—The experiment was performed in the following steps. We prepared a dataset, which contained ~ 100 MSL NAVCAM images sampled from a range of several sols. It was assumed that only 16 images out of the full dataset can be downlinked due to data capacity.

1. First, each subject was asked to select 16 images to downlink in one of the three following ways: i) random selection (the subject has no choice), ii) selection by on-board DBS, where the subject uplinks a set of keywords and the DBS algorithm returns the 16 most relevant images, and iii) ground-based selection with DBS, where the subject selects 16 images based on the auto-generated captions and thumbnails. The assumption made here is that although the rover cannot send back all its raw images, captions and thumbnails of all images are of sufficient size to be downlinked.
2. Next, each subject was asked to perform a scientific task given in a form of a questionnaire using the 16 downlinked images.
3. Then, the subjects were provided with the full dataset
4. Finally, each subject was asked to perform the same scientific task on the full dataset.

We observed the change in the scientific interpretation before and after seeing the full dataset. If the change is small, that indicates that the downsampled dataset properly represents the full dataset. We repeated this experiment three times with different datasets.

Tasks— We developed a questionnaire that was cognizant of the fact that different scientists have different research focuses and may therefore focus on different features. Scientists may interpret the same data differently depending on if they specialize in this particular field of expertise, and differences in interpretation are compounded by the limitations in the image dataset (i.e. there is more uncertainty in interpretation when images are limited in spatial extent and are of relatively coarse resolution). We therefore asked a set of straightforward science interpretation questions to

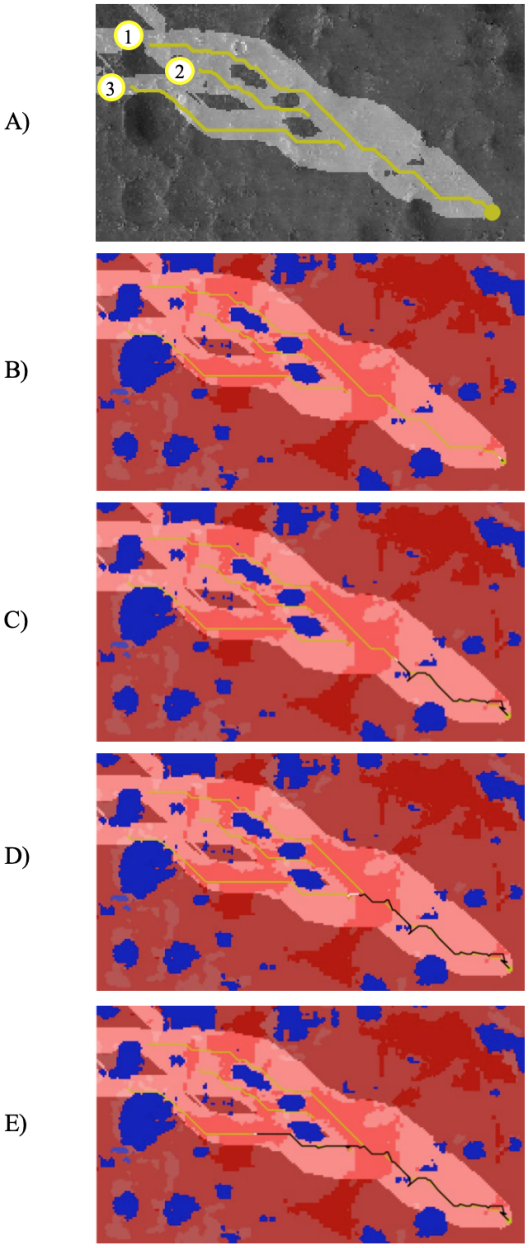


Figure 19: Preliminary results with the RAND algorithm. In A), three candidate trajectories are shown in yellow over the original (uncompressed) strategic search space covering the region in white. The underlying image is a map of Jezero Crater at a resolution of 5 meters per pixel. Subfigures B) to E) show the actual trajectory employed by a simulated rover (in black) over the same terrain represented as a speed map similar to Figure 18. The small white segment represent the intent of the rover based on the trajectories available around it. In Figure D), the rover switched from reference trajectory 1 to 3 as it was (strategically) the most time-optimal decision at that moment.

gauge how the subjects would interpret the depositional and diagenetic history of rocks presented in the same image sets, in addition to a second set of questions asking which test scenario they found to be most scientifically useful and why.

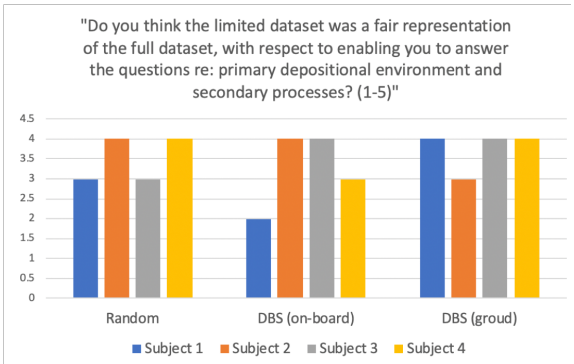
Results— Due to the small sample size and variability in subjects, sol ranges, and image subsets returned by different scenarios, it is difficult to quantify the performance and accuracy of interpretation in each scenario. For example, subjects had different interpretations when viewing the full dataset, compared to the limited dataset, for each scenario, demonstrating differences in their research background. Subjects who were viewing the same sol range of images but testing different scenarios would also interpret the images differently, but this difference is expected because the different scenarios returned a different subset of the images in that sol range. Figure 20 is the summary of semi-quantifiable questions collected in the experiment. Again, no definitive conclusions can be drawn from this experiment due to its small sample size. While the experiment was meaningful in that we could validate the tools and experimental approach, an obvious next step is to scale up the experiment and collect a statistically significant number of samples.

Subjects interpretations of the primary depositional environment were largely unchanged between the limited and full datasets, but in almost half of the tests, additional diagenetic features were recognized in the full dataset compared to the limited dataset. This difference may be due to feature scale and frequency, as sedimentary structures (used to infer primary depositional environment) span entire outcrops and are relatively large in scale, while diagenetic features are smaller features that tend to occur only in portions of outcrops.

Several subjects noted that while the limited datasets did capture the variability in sedimentary and diagenetic features in an image set, the best examples of such features were often available only in the full dataset. Others noted that they didn't notice a significant difference between the two datasets in terms of content, but became more sure of their interpretations when seeing the full dataset. When asked which scenario they preferred, most subjects unsurprisingly chose Scenario 4 (the option to select their own images to downlink).

Future Work—Recommendations for future work are to further refine the automated captions and to separate the use of different automated methods for different science use cases. To the first point, successful improvements to the automated captions would greatly improve trust in automated methods in a realistic operations or research scenario. Several subjects mentioned that they were wary of the captions after noticing inaccuracies between the automated caption and their actual observations and interpretations of that image. If it could be demonstrated that this captioning model was almost always accurate in terms of the geologic features identified, scientists may view the captions as being more reliable in the absence of the higher-resolution images.

Different automation methods may also have different applications in an operations or research scenario. Several subjects mentioned that while they did not want to rely on the automated captions for operations planning, they could imagine automated captions to be very useful in their own research, which can involve viewing and interpreting hundreds to thousands of images. In a research scenario, automated captions could help them quickly locate images that are relevant to their research. Another possibility, which was not



Most preferred method for data triage

Subject 1	Random
Subject 2	DBS (ground)
Subject 3	DBS (on-board)
Subject 4	DBS (ground)

Figure 20: Results summary of the DBS experiment. Note the small sample size.

Figure 21: Athena rover at JPL Mars Yard

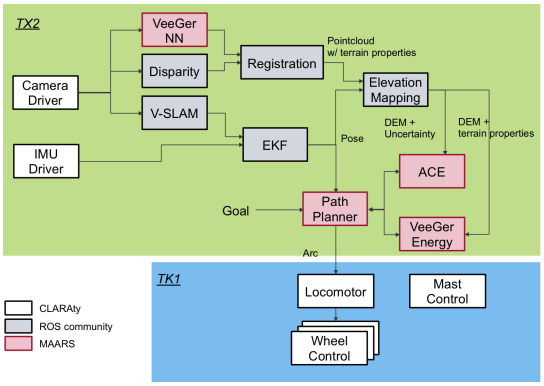


Figure 22: VeeGer Software Architecture on Athena rover

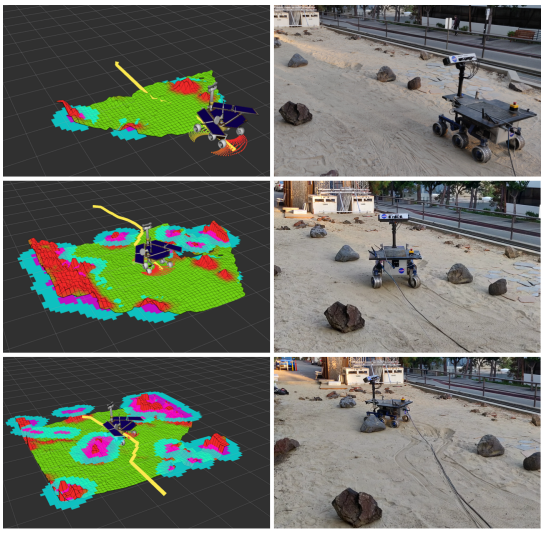


Figure 23: The Athena rover autonomously avoiding geometric obstacles to complete a 7-meters drive.

tested in this test, is to use image similarity searches to aid in an operations scenario. This method, which may be able to identify when a rover is driving across different terrain than was previously traversed, could be useful to identify when the rover has crossed a geologic contact or to track where similar lithologies/rocks are observed.

7. Integration with Athena Rover

Software architecture

All the on-board capabilities described in this paper will be integrated with the Athena Rover and tested in an analogous environment. The integration effort is currently ongoing. This section briefly summarizes the software architecture as well as the current status.

The Athena rover in Fig. 21 was developed at JPL as a testbed for on-board autonomy. Similar to the existing Mars rovers, the navigation system is primarily vision-based using a stereo camera rigidly attached to a movable mast. The Athena rover has a NVIDIA TX2 board as its brain, which has a multi-core ARMv8 CPU and a GPU.

The Athena rover software was written with the ROS [20] and CLARAty [21] frameworks. For portability, most of the MAARS software is written as ROS nodes and uses CLARAty's ROS interface to communicate with the rest of the system. Figure 22 shows a software architecture for VeeGer integration. Each block roughly corresponds to a single ROS node, and these nodes communicate over ROS topics. To focus on the algorithm development, we make use of open-source packages where possible.

Phase 1 results

The first deployment was recently completed and consisted in geometric-only terrain assessment and planning using standard ROS navigation tools and third-party open-source packages. This implementation projects the rovers stereo data onto a 2.5D map, which allows the identification of high steps, steep slopes and dangerously rough terrain. It is a baseline that our algorithms will build upon through incremental deployments and fields tests in the near-future.

8. HPSC/Snapdragon Deployment

We created a toolchain that allows to run a deep learning model on HPSC and Qualcomm's Snapdragon SoCs, as shown in Figure 24. We use TensorFlow to implement deep learning models. The TensorFlow models are converted to

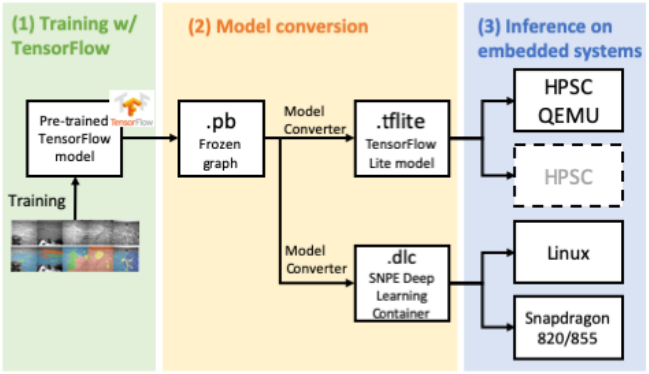


Figure 24: The toolchain for deploying a deep neural network model on HPSC and Snapdragon

TensorFlowLite and DLC formats for HPSC and Snapdragon, respectively.

Since the HPSC chiplet is not available yet, we used a QEMU software emulator running on CentOS. Note QEMU does not represent the performance of actual HPSC. The primary purpose of the QEMU deployment is to make sure the algorithms run on the chip. The secondary purpose is to make relative comparisons of run time with different models (e.g., floating point vs quantized), but again the run time on QEMU is likely not proportional to the actual run time on HPSC. As for Snapdragon, we selected Snapdragon 820 and 855 SoCs, both of which are based on multicore-heterogenious architecture are utilizing several versions of ARM CPUs, GPUs and DSPs. Note that the purpose of the benchmarking effort is *not* the comparison of the two processors as they are distinct from each other, although they share the same instruction set architecture (ARMv8) and have a comparable number of cores (HPSC:8, Snapdragon 820:4, 855:8). The results are unique to each processor.

HPSC platform

HPSC Chiplet integrates three subsystems - High Performance Processing Subsystem (HPPS) - several clusters of Quad Cortex A53 ARM CPUs, Real Time Processing Subsystem (RTPS) - Dual lockstep R52 ARM and Timing, Reset, Health Controller (TRCH) - Triple Modular Redundant low power ARM M4F core.

Snapdragon architecture

Snapdragon 820 is equipped with four ARMv8 cores (2 high efficiency and 2 low power), GPU Adreno 530 and Hexagon 680 DSP. Snapdragon 855 has tri-cluster configuration of 8 ARM cores (1 prime super high efficiency core for single threaded applications and 3 high efficiency Cortex-A76 cores, and 4 low power ARM Cortex-A55 cores), GPU Adreno 640, and Hexagon 690 DSP (with wide vector extensions (HVX), dynamic multi-threading, VLIW and SIMD instruction sets and AI Processor AIP)

Benchmarking

Qualcomm's Snapdragon Neural Processing Engine (SNPE) SDK provides tools to convert trained deep learning model from frameworks like Caffe/Caffe2, TensorFlow, Py-Torch to Qualcomm format DLC file which then is used to perform forward inference passes using one of the Snapdragon accelerated computing cores. To collect performance metrics we ran inference of SPOC lite model converted to DLC format on Snapdragon 820 and 855 SoCs and benchmark scripts provided with SNPE. Tables 1 and 2 shows our prelimiary benchmark results. More benchmarking will be performed in the third year of the project. In general, 8-bit quantization results in a non-trivial speed up on both platforms without significant changes in the classification accuracy. The computation time on HPSC QEMU is not reflective of the computation time of the real hardware, which would be significantly faster. On Snapdragon, GPU acceleration was not very effective due to the most computationally demanding operation (depth-wise convolution) was not supported by SNPE and depended on CPU fallback.

9. Conclusions

This paper describes the snapshot of all activities of the MAARS project at the end of the second year. The goals of the third and the last year of the project are to i) integrate all the algorithmic capabilities, ii) deploy the integrated capabilities on Athena rover and test them in analogous environments such as JPL's Mars Yard, and iii) further benchmarking of algorithms on HPSC QEMU and Snapdragon SOCs. At the end of the project, the technologies are expected to be matured to TRL 4-5.

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. Copyright 2019 California Institute of Technology. U.S. Government sponsorship acknowledged.

References

[1] R. Doyle, R. Some, W. Powell, G. Mounce, M. Goforth, S. Horan, and M. Lowry, "High performance spaceflight computing (hpsc) next-generation space processor (ngsp): a joint investment of nasa and afrl," in *Proceedings of the Workshop on Spacecraft Flight Software*, 2013.

[2] B. Balaram, T. Canham, C. Duncan, H. F. Grip, W. Johnson, J. Maki, A. Quon, R. Stern, and D. Zhu, *Mars Helicopter Technology Demonstrator*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2018-0023

[3] M. Ono, B. Rothrock, C. Mattmann, T. Islam, A. Didier, V. Z. Sun, D. Qiu, P. Ramirez, K. Grimes1, and G. Hedrick, "Make planetary images searchable: Content-based search for pds and on-board datasets," in *50th Lunar and Planetary Science Conference 2019*, March 2019.

[4] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *32nd International Conference on Machine Learning (ICML)*, 2015.

[5] L. Knaany, "Fast elasticsearch vector scoring," https://github.com/lior-k/fast-elasticsearch-vector-scoring, 2017.

[6] B. Rothrock, R. Kennedy, C. Cunningham, J. Papon, M. Heverly, and M. Ono, "Spoc: Deep learning-based terrain classification for mars rover missions," in *AIAA*

Table 1: Preliminary benchmark results on the HPSC QEMU software emulator. Note: [1]Computation time on QEMU is significantly slower than and not representative of the computation time on the real hardware.

Processor	Model	Format	Average computation time[1] [sec]
HPSC QEMU 2.0	SPOC (DeepLabv3)	TFLite floating point	82
		TFLite quantized (8 bit)	32
	SCOTI	TFLite floating point	27
		TFLite quantized (8 bit)	15

Table 2: Preliminary benchmark results on Snapdragon 820/855. Note: [2]Computation time includes CPU fallback, which accounts for most ($\geq$ 95%) computation time because the particular convolution operation used by DeepLabV3 was not supported by SNPE at the time of benchmarking.

Processor	Model	Format	Acceleration	Average computation time [sec]
Snapdragon 820	SPOC(DeepLabv3)	DLC floating point	CPU only	36.8
			GPU	9.6[2]
		DLC quantized (8 bit)	DSP	0.62
Snapdragon 855	SPOC (DeepLabv3)	DLC floating point	CPU only	8.2
			GPU	5.6[2]

SPACE 2016, September 2016, pp. 1–12. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2016-5539

[7] K. Otsu, G. Matheron, S. Ghosh, O. Toupet, and M. Ono, "Fast approximate clearance evaluation for rovers with articulated suspension systems," *Journal of Field Robotics*, vol. 0, no. 0. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21892

[8] S. Ghosh, K. Otsu, and M. Ono, "Probabilistic kinematic state estimation for motion planning of planetary rovers," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5148–5154.

[9] S. Higa, Y. Iwashita, K. Otsu, M. Ono, O. Lamarre, A. Didier, and M. Hoffmann, "Vision-based estimation of driving energy for planetary rovers using deep learning and terramechanics," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3876–3883, Oct 2019.

[10] K. Otsu, G. Matheron, S. Ghosh, O. Toupet, and M. Ono, "Fast approximate clearance evaluation for rovers with articulated suspension systems," *Journal of Field Robotics*, 2019.

[11] A. Jain, J. Guineau, C. Lim, W. Lincoln, M. Pomerantz, G. Sohl, and R. Steele, "ROAMS: Planetary surface rover simulation environment," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2003.

[12] A. Jain, J. Balaram, J. Cameron, J. Guineau, C. Lim, M. Pomerantz, and G. Sohl, "Recent developments in the ROAMS planetary rover simulation environment," in *IEEE Aerospace Conference*, vol. 2, 2004, pp. 861–876.

[13] S. B. Goldberg, M. W. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Proceedings, IEEE aerospace conference*, vol. 5. IEEE, 2002, pp. 5–5.

[14] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[15] M. Ono, M. Heverly, B. Rothrock, T. Ishimatsu, E. Almeida, F. Calef, T. Soliman, N. Williams, H. Gengl, A. Nicholas, E. K. Stilley, K. Otsu, M. Trautman, R. Lange, and S. Milkovich, "Mars 2020 surface mission performance analysis: Part 2. surface traversability," in *AIAA SPACE Symposium 2018*, 2018.

[16] R. L. Fergason, P. R. Christensen, and H. H. Kieffer, "High-resolution thermal inertia derived from the thermal emission imaging system (themis): Thermal model and applications," *Journal of Geophysical Research: Planets*, vol. 111, no. E12, 2006.

[17] N. E. Putzig, M. T. Mellon, K. A. Kretke, and R. E. Arvidson, "Global thermal inertia and surface properties of mars from the mgs mapping mission," *Icarus*, vol. 173, no. 2, pp. 325–341, 2005.

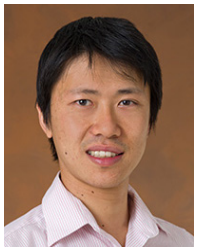
[18] M. Ono, B. Rothrock, E. Almeida, A. Ansar, R. Otero, A. Huertas, and M. Heverly, "Data-driven surface traversability analysis for mars 2020 landing site selection," in *2016 IEEE Aerospace Conference*, March 2016, pp. 1–12.

[19] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," in *Robotics: Science and Systems VI*, 2006.

[20] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: An Open-source Robot Operating System," in *IEEE International Conference on Robotics and Automation, Workshop on Open Source Software*, 2009.

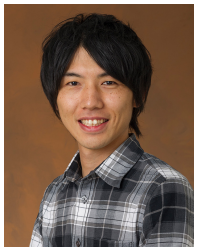
[21] I. A. Nesnas, R. Simmons, D. Gaines, C. Kunz, A. Diaz-Calderon, T. Estlin, R. Madison, J. Guineau, M. McHenry, I.-H. Shu, and D. Apfelbaum, "CLARAty: Challenges and steps toward reusable robotic software," *International Journal of Advanced Robotic Systems*, vol. 3, no. 1, pp. 23–30, 2006.

BIOGRAPHY



Masahiro (Hiro) Ono *is a Research Technologist at NASA Jet Propulsion Laboratory, California Institute of Technology. His broad interest is centered around the application of robotic autonomy to space exploration, with an emphasis on machine learning applications to perception, data interpretation, and decision making. Before joining JPL in 2013, he was an assistant professor at Keio University in Japan. He graduated from MIT with PhD in Aeronautics and Astronautics in 2012.*

Brandon Rothrock *is a former member of the computer vision group at JPL. His research is focused on models for semantic image understanding, visual perception and obstacle avoidance for micro UAVs, and object detection. In September 2019, he joined Paige, where he is applying machine learning methods to medical image analyses.*



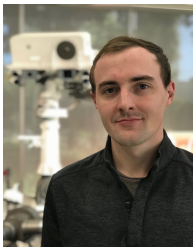
Kyohei Otsu *is a Robotics Technologist at NASA Jet Propulsion Laboratory, California Institute of Technology. He received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from the University of Tokyo. He is interested in autonomous navigation technologies and robotic system architecture.*

Shoya Higa *is a postdoctoral researcher at the NASA Jet Propulsion Laboratory, California Institute of Technology. His research interests focus on mobility performance analysis of lunar/planetary rovers, particularly wheelsoil interaction (terramechanics), sensing and estimation of rover's wheel performance, robot autonomy, in-situ terrain assessment, and ice climbing robots. He received his master's degree and Ph.D. (Engineering) from Tohoku University in 2014 and 2017, respectively.*

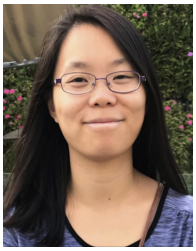
Yumi Iwashita *is a research technologist at the NASA Jet Propulsion Laboratory, California Institute of Technology (Caltech), Pasadena, California and an adjunct associate professor at Kyushu University, Japan, from 2016. She received her M.S. degree and her Ph.D. from the Graduate School of Information Science and Electrical Engineering, Kyushu University.*

Annie Didier *is a data scientist at NASA Jet Propulsion Laboratory, California Institue of Technology. She received her M.S degree in Analytics from Northwestern University in 2017, and her M.A. in Applied Mathematics (computation neuroscience) from Arizona State University in 2015.*

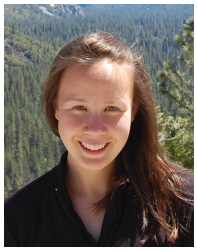
Tanvir Islam *is a Research Technologist at NASA Jet Propulsion Laboratory, California Institute of Technology. He received the Ph.D. degree in Engineering (remote sensing and machine learning) from the University of Bristol, Bristol, UK, in 2012. He has published four books and more than 70 peer-reviewed papers in leading international journals. His research interests include deep learning and machine learning applications to computer vision, data science, and geosciences problems.*



Christopher Laporte *is a data scientist in the Chief Technology and Innovation Office at the NASA Jet Propulsion Laboratory, California Institute of Technology. He received his B.Sc. in Computer Science at the University of British Columbia. His interests span the domains of data visualization, machine learning, robotics, and software engineering. Most recently his work has been focused on visualizing intelligent systems.*



Vivian Sun *is a planetary scientist and systems engineer at NASA Jet Propulsion Laboratory, California Institute of Technology. She received her Ph.D. in Planetary Science from Brown University in 2017 and her B.S. in Planetary Science from the California Institute of Technology in 2012. She is currently a Science Operations Systems Engineer for the Mars 2020 mission, and her science research focuses on understanding Mars surface geology and mineralogy using orbital datasets, reflectance spectroscopy, and rover datasets.*



Kathryn Stack *is a planetary geologist at NASA Jet Propulsion Laboratory, California Institute of Technology. She received her Ph.D. in Geology from Caltech in 2015 and her B.A. in Geosciences and Astronomy from Willams College in 2008. She is a participating scientist on the Mars Science Laboratory Curiosity rover and the Deputy Project Scientist for the Mars 2020 mission. Her science research focuses on studying the ancient sedimentary rock record of Mars using rover and orbiter images to understand past surface processes and habitability.*

Jacek Sawoniewicz *is a Robotics Systems Engineer in the Robotic Mobility Group at the NASA Jet Propulsion Laboratory, California Institute of Technology. He received his M.S. degree in Electronics Engineering and Robotics from Warsaw University of Technology, Faculty of Electronics and Information Technology in 1999. His main interests are hard real-time software and space robotics.*

Shreyansh Daftry *is a Robotics Technologist at NASA Jet Propulsion Laboratory, California Institute of Technology. He received his M.S. degree in Robotics from Carnegie Mellon University , and his B.S. degree in Electronics and Communication Engineering in 2013. His research interests lies at the intersection of computer vision, machine learning and autonomous robotics.*

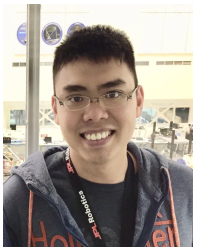
Virisha Timmaraju *is a data scientist at the NASA Jet Propulsion Laboratory, California Institute of Technology. She received her M.S (2018) in Electrical Engineering from Texas A & M University, and her B.Tech (2016) in Electronics and Communications Engineering from GITAM University. Her research interests are in applying information theory and machine learning to astrophysics and planetary sciences.*

Sami Sahnoune *is a data scientist in the Chief Technology and Innovation Office at the NASA Jet Propulsion Laboratory, California Institute of Technology. He received his B.Sc. in Computer Science and completed a minor in Physics at McGill University in 2019. His work is focused on data analytics and visualization.*

Chris A. Mattmann *is the Deputy Chief Technology and Innovation Officer in the Chief Technology and Innovation Office at the NASA Jet Propulsion Laboratory, California Institute of Technology. He received his B.S. (2001), M.S. (2003) and Ph.D. (2007). in Computer Science from the University of Southern California and has been at JPL for nearly 20 years. His research interests are in data science, machine learning, open source and information retrieval.*

Olivier Lamarre *is currently an intern at the NASA Jet Propulsion Laboratory, California Institute of Technology and a Ph.D. student in Aerospace Engineering and Robotics at the University of Toronto. He received his B.Eng. in Mechanical Engineering and completed a minor in Geology at McGill University in 2017. His research focuses on enabling long-term and resource-aware autonomy onboard planetary rovers.*

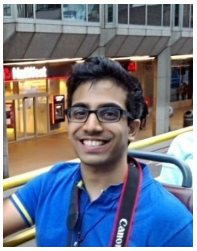
Sourish Ghosh *is a former intern at NASA Jet Propulsion Laboratory. He is currently a Ph.D. candidate in Robotics at Carnegie Mellon University. He received his B.Sc. and M.Sc. degrees in Mathematics and Computing from Indian Institute of Technology, Kharagpur in 2019. His interests lie in developing general long-term autonomy for agile aerial vehicles.*



Dicong Qiu *is a former intern at NASA Jet Propulsion Laboratory. He is currently a research engineer in planning and decision making for self-driving at ISEE AI. He received his M.S. degree in Robotics from Carnegie Mellon University in 2018 and B.Eng. degree from Sun Yat-sen University in 2015. His interests lie in robotics, machine learning for concept cognition, planning and decision making under uncertainty.*

Shunichiro Nomura *is a former intern at the NASA Jet Propulsion Laboratory, California Institute of Technology. He is currently a Ph.D. candidate in Aeronautics and Astronautics at the University of Tokyo studying robust space infrastructure deployment in an uncertain environment. He is also a lead ADCS engineer in a deep-space CubeSat project EQUULEUS.*

Hiya Roy *is currently an intern at the NASA Jet Propulsion Laboratory, California Institute of Technology. She is a Ph.D. candidate in the Electrical Engineering and Information Systems department at the University of Tokyo. Her research interests lies at the intersection of deep learning and planetary sciences.*



Hemanth Sarabu *is a research affiliate and former intern at the NASA Jet Propulsion Laboratory, California Institute of Technology. He is currently a Master student pursuing Computational Science and Engineering at Georgia Tech conducting machine learning and robotics research at the Georgia Tech Research Institute and the Georgia Tech Robot Learning Lab. His research pursuits include safe reinforcement learning, neural network robustification, and planning for unstructured environments.*

Gabrielle Hedrick *is a Ph.D. candidate in Aerospace Engineering at West Virginia University, working on terramechanics and path planning for Mars rovers. Before transferring to WVU, she was a Ph.D. candidate at Washington University in St. Louis, where she earned an M.A. in Earth and Planetary Sciences in 2013 and was a member of the Mars Exploration Rover Science team. Prior to this, she graduated with a master in Mining Engineering from the Ecole Nationale de Geologie in France.*

Larkin Folsom *is a Ph.D. candidate in Computational Science and Engineering at North Carolina A&T State University whose research is focused on path planning in stochastic and dynamic environments. He received his M.S. in Computational Science and Engineering from North Carolina A&T State University.*

Sean Suehr *is a former intern at the NASA JetPropulsion Laboratory, California Institute of Technology. He is currently an operations research scientist at Naval Surface Warfare Center, Crane Division and working towards implementing combinatorial optimization methods for system reliability, availability, and maintainability. He recently received his Master's degree in Industrial and Systems Engineering from North Carolina AT State University with research in clique relaxations applied to criminal networks, and previously earned a Bachelor's of Science from Lake Superior State University in mathematics. His research interests include combinatorial and network optimization for Department of Defense applications.*

Hyoshin Park *is an Assistant Professor of the department of Computational Science and Engineering at the North Carolina A&T State University. He received his B.S. (2005) in Physics and B.S. (2005), M.S. (2007) in Urban Planning & Engineering (2005) from the Yonsei University, Ph.D. (2016). in Civil Engineering from the University of Maryland College Park. His research interests are in information-theoretic adaptive navigation and knowledge discovery.*