DOI: 10.13009/EUCASS2019-813

8TH EUROPEAN CONFERENCE FOR AERONAUTICS AND AEROSPACE SCIENCES (EUCASS)

DOI: ADD DOINUMBER HERE

# Open-source AI Assistant for Cooperative Multi-agent Systems for Lunar Prospecting Missions

*Z. M. Kakish*[1†], *F. RodrÃguez-Lera*[2], *D. Bischel*[3], *A. Mosquera*[4], *R. Boumghar*[5],
*S. Kaczmarek*[6], *T. Seabrook*[7], *P. Metzger*[8], *J.L. Galanche*[4]
[1]*Arizona State University,* [2]*University of León,* [3]*University of California Santa Cruz,* [4]*Aten Engineering,*
[5]*Librespace Foundation,* [6]*Imperial College London,* [7]*Oxford University,* [8]*University of Central Florida*
zahi.kakish@asu.edu · fjrodl@unileon.es · dbischel@ucsc.edu · ana@atenengineering.com
redouane.boumghar@free.fr · s.kaczmarek17@imperial.ac.uk · timothy.seabrook@cs.ox.ac.uk
Philip.Metzger@ucf.edu · jl@atenengineering.com
†Corresponding author

## Abstract

Mission planning for space prospecting missions is a complex undertaking that involves several phases, variables, and systems. MARMOT, Multi-Agent Resource Mission Operations Tools, is an extensible, open-source tool that allows operators to expand and improve current functionalities and automation capabilities of mission planning systems for extra-terrestrial operation by aiming to reduce the exhaustive mental load accumulated by mission operators. This paper presents the first iteration of the tool that provides homogeneous and heterogeneous multi-agent global trajectory planning and optimization suggestions for different tasks and goals in a Lunar environment. The resulting tool leverages collaborative action plans and optimization strategies for enhancing in-situ resource exploitation and discovery on the Moon.

## 1. Introduction

In recent years, we have seen a major turning point in the development of Space exploration capabilities. Government Space agencies are achieving great milestones in this *NewSpace* race by mixing and assisting commercial ventures. All these advances, and humanity's inherent hunger for adventure and exploration, are pushing forward the development of the space sector in an exponential way, helping envision a near future with settlements on the Moon and Mars. For this aspiration to come true, and establishing permanent presence in space, in-situ exploration, identification, and prospecting of space resources is paramount to sustain human activities. Mining water, for example, will be essential for life-support systems, and can, additionally, be split and used as rocket propellant. If carbon is found in water-ice, as it's believed to exist, it can be transformed into plastics that can be used to protect against radiation, and other minerals can be targeted with the goal of construction.

Five years ago, NASA started the Resource Prospector (RP) mission[1]. This mission aims to prospect for volatiles such as water-ice in polar regions of the Moon in order to demonstrate an In-Situ Resource Utilization (ISRU) capability. These resources must lay the foundations for space exploration. Nowadays, the NASA Frontier Development Lab [13], is offering alternatives and solutions to many of the problems associated to the RP mission. NASA Frontier Development Lab is an AI accelerator geared towards space applications. This program joins multidisciplinary researchers working together to solve some of the most interesting problems in current space missions.

The authors, as a team, have put forward a stepping stone towards that goal in the form of MARMOT, or Multi-Agent Resource Mission Operations Tools. *MARMOT* is an extensible, open-source tool that allows Earth operator to expand the current functionality of mission planning systems for extra-terrestrial operation. The first iteration of the tool provides heterogeneous multi-agent global planning optimization of environments for different tasks and goals. Future iterations of the work will allow extensibility into other areas of machine intelligence and robotics, and further enhance the goals of both public and private space operations.

DOI: 10.13009/EUCASS2019-813



## 2. Prospecting Mission Overview

As explained before, the Resource Prospector (RP) mission aims to perform "In-Situ resource utilization" using a rover on the Lunar surface. However, the inclusion of ISRU instruments on the rover increases the complexity of classical navigation and exploration tasks. Thus, the process for controlling a rover during a prospecting mission is composed of two Mission Control systems: the Mission Control located on Earth surface (MCoE) and the Mission Control located on the rover vehicle (MCoR). The former, is the one managed by a mission team (human operators) facing a front-end interface and selecting the best driving distances and areas using available information from rover sensors. As a result of this process, they will command the rover to visit a new position or to perform ISRU. The latter is the one deployed on the robot, and allows manual control, teleoperated by a human operator, and semi-autonomous control, where the robot is running with partial supervision. The MCoR would be seen as the control unit for the localization, navigation, and prospecting systems. The robot generates actions and gathers information from its sensors for relaying to the MCoE. Nevertheless, current missions have associated with them several issues that may be abstracted into three categories: 1) Moon to Earth communications and interactions with humans operators in the loop, 2) harshness and unpredictability of the environment, and 3) single robot mission.

Given the Earth-Moon geometry, communication between a rover on the Moon and mission control on Earth is not a straightforward process. The round trip communication latency for a mission on the Moon is significantly shorter (varying between six seconds to over 25 seconds for RP) than Mars, which could expect a response time of 28 minutes for a software ping or to perform a rover operation citeGordon:2012. This scenario discourages real-time command and control. Thus, any action undergone by the rover is prepared and meditated by a team that decomposes each task to fine-grain steps. The idea is to define decision cycles for providing near real-time command and control by means of driving the rover using a waypoint approach [1].

The Moon's environment [16] is composed of numerous impediments to simple operations which operators are accustomed to on Earth. It has a solid-surface body made up of a rocky surface covered with regolith. Consequently, the regolith material and spectral properties make traversal difficult using conventional platforms. Regolith covering the surface varies between 3 meters and 20 meters. The Moon presents[4] a crust size of 60 to 100 kilometers thick. In addition, the Moon has other properties[3] such as dust constrains, thermal properties, and vacuum conditions which make operation difficult.

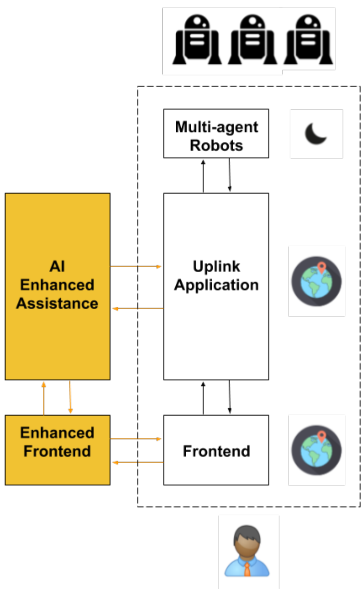


Figure 1: Overview of a mission control. Expected goals (supported on AI) vs current scenarios (supported on Earth surface mission control).

In the RP mission, the single rover lacks autonomy, traversal versatility, communication capabilities and characteristics for reaching the goal in an expected amount of time. These challenges can be significantly reduced by multi-robot operations, which can improve mission efficacy and lower risks by autonomously coordinating the agents. Despite

DOI: 10.13009/EUCASS2019-813



continual advances in the field, there are still numerous challenges to reduce the complexities associated with the coordination of multi-robot systems and mission planning strategies, which motivates further research.

This project proposes a set of tools capable of enhancing the planning capabilities of the Mission Control Team on Earth during the decision making process of prospecting and exploration missions. In addition to providing operators important information, the system assists controlling a Multi-agent configuration of rovers. The first iteration of development presents a mission generator that provides alternatives to a prospecting mission on the Moon. Figure 1 presents the overview of our project, where this project is contributing and current status of a mission definition.

## 3. Mission Generator: AI Assistance

Before explaining the MARMOT solution, it is necessary to identify the main features associated to MCoE and MCoR. The Moon context clearly bounds the possibilities available for offering a guidance system for Mission Control on Earth. In addition, current information from rover sensors and satellites limits the information available to make the best decision. Thus, this project initially selects three main elements: 1) Cost Map generation, 2) Waypoint Generation and Guidance, and 3) Multi-agent strategy.

### 3.1 Cost Map Generation

To produce a usable cost map, a blank grid-graph containing vertices and edges the size of the operational area is generated. Next, each vertex is associated with it information about the environment. The information is used to generate a cost associated with that vertex for eventual use in path planning strategies. For example, the distance of a vertex to an obstacle is assigned a low value if the vertex is far from obstacle, but a high value is assigned if it is closer. The values given to vertices on the grid-graph are taken from fusing multiple maps in layers of the same area containing different information [18][7]. In the case of a Lunar environment, the fusion entailed layering terrain, slope, shadows, and communication maps.

#### 3.1.1 Lunar Terrain and Slope Maps

Currently, there are no publicly available high-resolution, finely-detailed maps of the Lunar surface. A terrain map must thus be made utilizing information that *is* publicly available. *Moon Trek* [17], a website hosting publicly available Moon data organized by NASA JPL, provides easy and free access to Lunar surface data. We focused our efforts on the crater *Hermite A* located on the Moon's North Pole since this was the tentative operational area of the Resource Prospector [19] mission. Inspired by images from the Lunar **O**rbiter **L**aser **A**ltimeter on board the *Lunar Reconnaissance Orbiter* (LRO) of the area, an artificial Lunar surface was created with the help of a 3D open source creation suite, *Blender* [12].

From the LRO mission, Digital Elevation Models (DEM) and other surface information provided approximate values of the Moon's topological surface, which was used for creating mock Lunar area called the *terrain-slope* map. Figure 2 is one such example of real data topological surface where each pixel represents terrain inclination. In the image, black pixels correspond to flat surfaces while white pixels are steep slopes. The grayscale variations in-between black and white scale from shallow (dark grey pixels) to steep (light grey pixels) surfaces. Figure 3 demonstrates the original image DEM converted and modified to a similar artificial surface using Blender.

#### 3.1.2 Shadow Map

The 1.5° tilted position of the Moon causes different shadow periods on surface. Near the poles, some craters have local depressions in continual permanent shadow while, on the other hand, we can find craters with the outer edge remaining sunlit throughout the year. There are identified four different areas on Lunar surface: Sunlit, Short Duration Sunlit, Shadowed Near Sunlit, and Permanently Shadowed Regions (PSR). It is important to highlight that data obtained by Lunar Reconnaissance Orbiter suggest it is possible to find thick ice and other volatiles on the top 1-2 meters of regolith on shadowed regions at the lunar poles. Those are thus regions of interest for the prospecting missions, and, for this reason, current missions to the Moon require solar power for energy. A rover caught in a shadowed region can have a detrimental effect on a mission, which include the inability to recharge and having to endure large deviations in

DOI: 10.13009/EUCASS2019-813



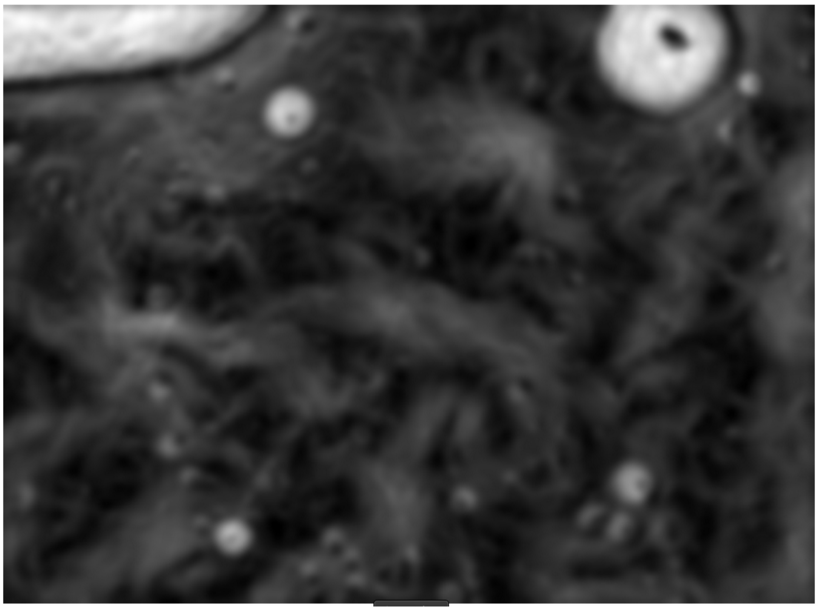

Figure 2: Terrain slope representing terrain inclination at each pixel.

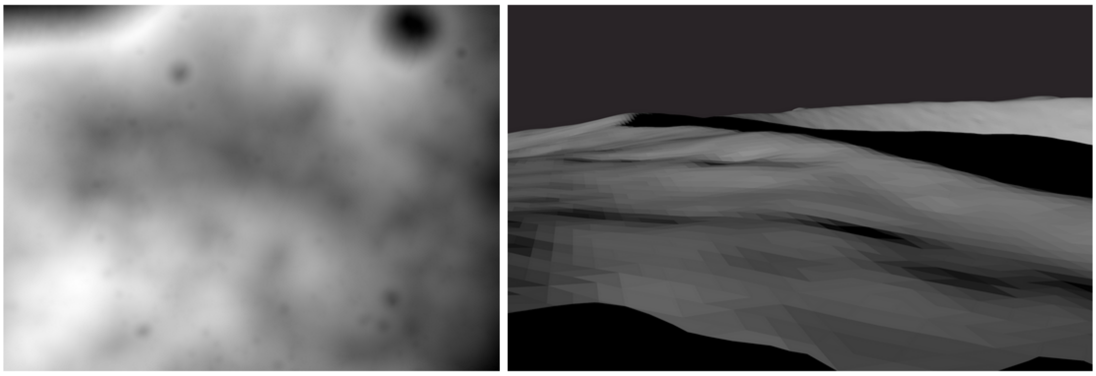

Figure 3: Moon surface digital elevation model DEM (left) and a generated view in Blender from this DEM (right).

temperatures. Data has shown that temperatures in shadowed regions are as low as $-250°C$. Under these circumstances, the *shadow* map layer consists of marking areas that are consistently covered in shadow. However, as explained earlier these shadows change throughout time. A path obtained taking into account the lunar shadow layout at a certain period of time could be dangerous several hours later.

Using Blender, a simulated Moon cycle can provide the needed shadow information for a potential site as demonstrated and expanded upon in Figure 4. The figure provides a great visualization of the variability in surface illumination over time. The map is then used as a basis for generating a cost attributed to shadow coverage. Areas that are more frequently covered in shadow are given a higher costs, while areas with little shadow coverage are given lower costs. This translates to a shadow map that can assist a rover in navigating away from areas that are deemed high-risk due to the prevalence of shadows.

#### 3.1.3 Communications Map

During a Lunar mission, direct communication with Earth is required to oversee and control rover operations. However, given certain rover hardware and autonomy limitations, including issues with geographical location, maintaining direct communication is not consistently achievable. Improving the communication hardware available to a rover is a simple solution, yet this is economically infeasible for many small space companies or research initiatives. Therefore, we explore indirect communication methods using multiple agents as a possible solution. For example, a stationary lander

DOI: 10.13009/EUCASS2019-813



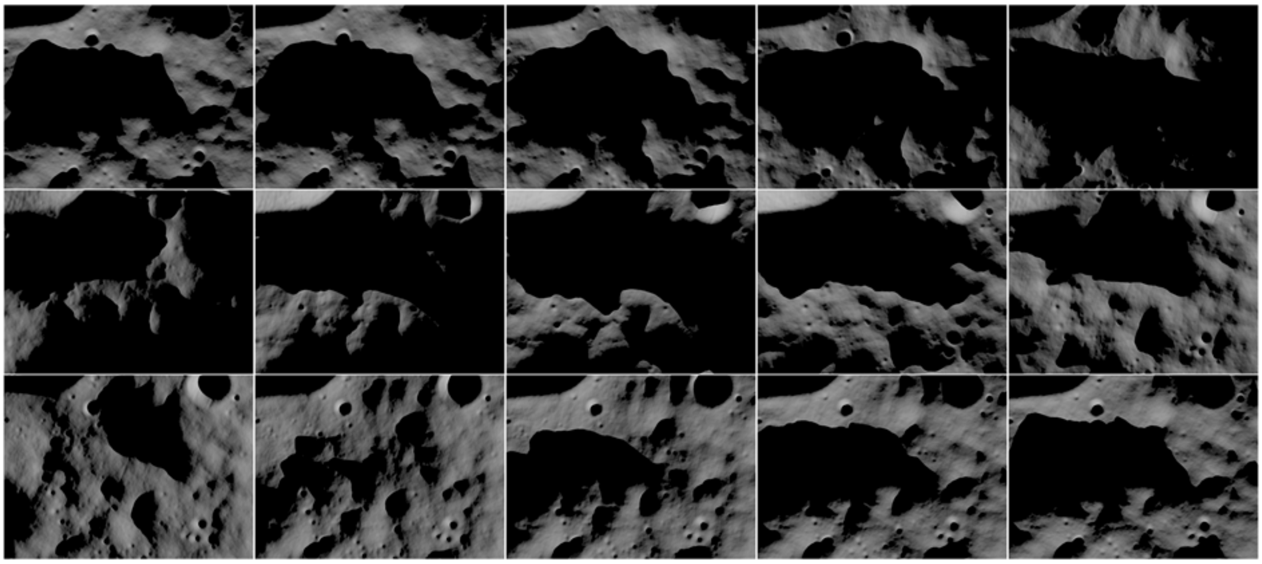

Figure 4: Shadow shapes on Moon surface greatly change with respect to moon-sun angle along the moon cycle (28 days). The images show the evolution every 44 hours. Brighter areas are those with exposure to the sun, while the darker areas are occluded by terrain into darkness.

acting as an intermediary between a rover and Earth. Pal *et al.*[20] analyzed this type of multi-agent scenario with assumed perfect communication between the lander and rover. This, however, does not accurately represent the communication connection strength between a lander and rover on a real Lunar mission. Multiple factors such as the Moon surface topology, distance-dependent constraints, and antenna intrinsic features [2] significantly complicate communication in this configuration. Additionally, certain operations require the rover to explore areas where communication is completely hindered (such as a crater), and necessitates the assistance of a relay rover to maintain connection.

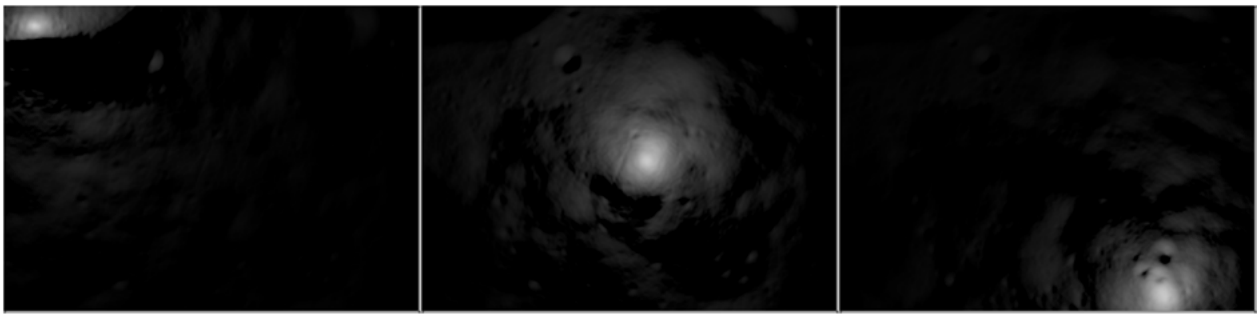

Figure 5: Three communications models generated in Blender (ray tracing) showing scientific rover (left), relay rover (middle), and lander (right) communication regions. The scale of worst to best communication coverage and strength.

A *communication* map is thus derived from modeling these constraints and simulating them in a Blender environment. Figure 5 shows a Blender generated communication map. The figure shows three images with a representation of the strength and coverage of rover transmission based on its location. The values scale from black (worst) to white (best) for each pixel in the image. The fusion of thse single communication models produces the complete communication region. Communication cost maps generated from this image contain values analogous to the image's grayscale values. Figure 6 is obtained as a result of this process.

### 3.2 Waypoint Generation and Guidance

It is necessary to locate and select those scientifically interesting targets which could be great sources of information to the missions. Traversal between these targets –points-of-interest– is performed by a navigation system. This navigation system provides the mechanisms to move the rover on the Moon surface between two or more points. Current process

DOI: 10.13009/EUCASS2019-813



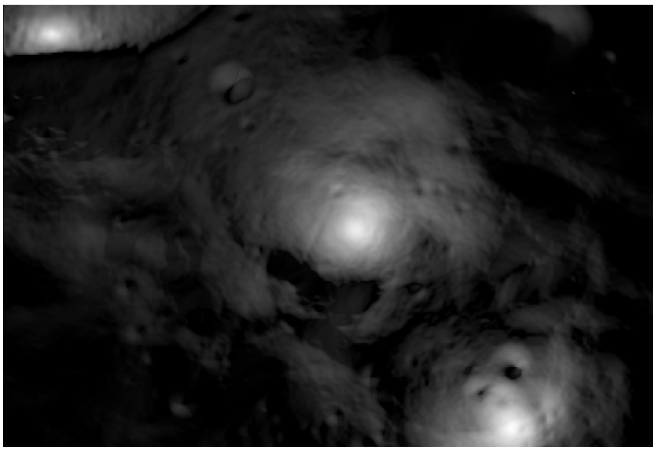

Figure 6: Total communication coverage.The grayscale of worst to best communication coverage and strength.

includes a team on Earth selecting the region of interest that the rover should visit. This process is supported on several sources of information such as local images, information from Lunar Reconnaissance Orbiter, and rover sensors.

#### 3.2.1 Points-of-Interest Selection

Since our work was modeled on prospection tasks, and more specifically the Resource Prospector mission, points-of-interest (POI) for a Lunar operation consisted of locating the presence of water and predicting the thermal stability of ice with respect to the depth of the Lunar surface. According to Colaprete[10], these *Resource Target Regions* (RTRs) are defined as four regions. *Dry regions* where temperatures are too warm for ice to be stable. *Deep regions* where a stable layer of ice is expected between 50-100 *cm* from the surface. *Shallow regions* where a stable layer of ice is expected 50 *cm* from the surface. *Surface regions* where surface ice is expected, which is typically in a PSR location.

#### 3.2.2 Opportunistic Path Planning

Current approaches for solving these challenges are based on two main methods [9]: *blind drive*, where Earth operator defines the robot path and the rover follows that route with no identification of hazardous conditions or new points of interest; *autonomous navigation*, where the rover identifies local threats such as non-traversable rocks or shadowed regions surrounding them on its path to the target location. In the case of Lunar prospecting, opportunistic path planning for multiple agents consists of selecting points-of-interest that will yield the greatest amount of scientific return while maintaining constant communication between the agents. Attempting to solve this requires an extensive state-action space due to discretizing states into number of agents, number of points-of-interest, and path proposals.

One can reduce this space by splitting the problem into two parts. The first selects and manages the relation of the multiple agents and the points-of-interest by using a brute-force traveling salesman solver as specified by Gutin and Punnen [14]. An $A^*$ algorithm is then used to establish an approximated best traversable route between 2 or $n$ points-of-interest. The second part takes the approximated paths for each agent and uses a version of the Distributed Path Consensus (DPC) algorithm [5] to augment the traversal paths for maintained communication between the agents.

### 3.3 Multi-Agent Strategy

As briefly mentioned in the Communication Map section, a multi-agent system can significantly expand the capabilities of a Lunar prospecting and exploration mission, yet this paradigm comes with challenges [11]. Agents in this setting lack full-autonomy and are remotely controlled by human operators on Earth. Increasing the number of agents without any additional autonomy concurrently increases human operator burden. The work presented in this paper provides a new multi-agent automation strategy to make multi-agent missions more viable for future missions. Table 1 presents a general overview of the pros and cons of the mission using a single robot or deploying a multi-agent cooperative configuration.

DOI: 10.13009/EUCASS2019-813



For a prospecting mission, we examined cooperative strategies that provided extended, reliable communication on the Lunar surface because agents may enter RTRs that limit communication. The two possible agent configurations are expanded upon in following subsections. The configurations proposed in this research are based on three different agents with different physical characteristics and roles:

- **Lander spacecraft**: This agent serves multiple roles during a Lunar mission. At the beginning of the mission, the Lander acts as the payload module containing the rovers for arrival on the Lunar surface. The lander is strategically placed on higher elevation to allow consistent, unfettered communication with Earth as its second role.
- **Scientific rover**: A robotic rover capable of every possible mission defined scientific experiment. It communicates directly with Earth, or by intermediaries such as a Relay rover or a Lander spacecraft.
- **Relay rover**: Unlike the Scientific rover, this rover is primarily tasked with assisting other rovers maintain communication with Earth. These rovers are less capable of doing thorough scientific experiments, but may be equipped with some basic experiment support. Some descriptions have the relay rover solely act as a communication relay and have no scientific capabilities.

#### 3.3.1 Single-class configuration

A lander spacecraft on the Lunar surface deploys two Scientific rovers of the same type. Both rovers have the same capabilities, but are more conservative in their task allocation so as to maintain communication to the lander. If a rover must enter an area with little possibility of communication with the lander (such as a crater or a low elevation PSR), the other scientific rover may act as a relay to maintain communication.

#### 3.3.2 Multi-class configuration

A lander spacecraft on the Lunar surface deploys two rovers: a scientific and a relay rover. The lander acts as a relay to communication back to Earth. The scientific rover undergoes the necessary scientific tasks for the mission while the relay rover maintains a communication connection between the scientific rover and the lander. Additionally, the relay rover may provide limited secondary scientific task capabilities.

## 4. MARMOT

Multi-Agent Resource Mission Operations Tool (MARMOT) was developed to help create better automation solutions to multi-agent problems. Specifically, MARMOT gives users the ability to define an optimized set of trajectories to enhance mission performance for exploration and/or prospecting tasks.

### 4.1 Pipeline

From a macro-abstract view, MARMOT acts as a simple tool capable of easy integration in existing backend systems. Building a mission pipeline utilizing MARMOT begins with by providing databases (maps and points-of-interests related to the mission), mission constraints, and operator preferences as detailed in Figure 7. The latter two are user-defined and contain required mission information to further tune and modify the generated cost map. For example, mission constraints such as the mission-operation time window (date and length), which will be used to generate the relevant time-dependent version of the shadow maps.

Once the data of each map is layered and fused to create the overall terrain cost map, MARMOT selects POIs to integrate into said map. Finally, MARMOT uses the master cost map to generate the multi-agent trajectories by deploying built-in path planning algorithms.

DOI: 10.13009/EUCASS2019-813



Table 1: Prospecting mission tasks using Single-robot (SR) configuration vs Multi-robot (MR) configuration (SC:Single-class; MC:Multi-class)

| | **Strategy** | **Pros** | **Cons** |
|---|---|---|---|
| General Overview | SR | Cost, control | No backup, communications range |
| | MR-SC | Specialized tasks, Backup rover | Cost, Coordination, Payload |
| | MR-MC | Specialized tasks, Reduced costs (compared to SC), Payload | Coordination, No backup rover |
| **Resource Prospecting Task** | **Recommended** | **Comments** | |
| Visit PSR | MR-SC | It is necessary to maintain communication flow using two scientific robots. Possibility of continuing mission in case of lost robot in PSR. It increases the cost. Similar robots contain the same control strategy. | |
| Explore regions | MR-MC | Relay rover could extend the range of the explorable region. | |
| Maximize the scientific results | MR-SC | Two scientific rovers could perform tasks in parallel while simultaneously offering communication extension. | |
| Deploy infrastructures | MR-SC | Coordinated tasks are simplified when the same robot is being used. When the same robot is a scientific it increases the cost and dangers over the platform. | |
| Rescue robots | MR-MC | This would require the addition of necessary tools for rescue tasks. | |
| Extend communications range | MR-MC | A specialized rover for communication will be a right option. Minimal scientific rover with extended comms will decrease costs. | |

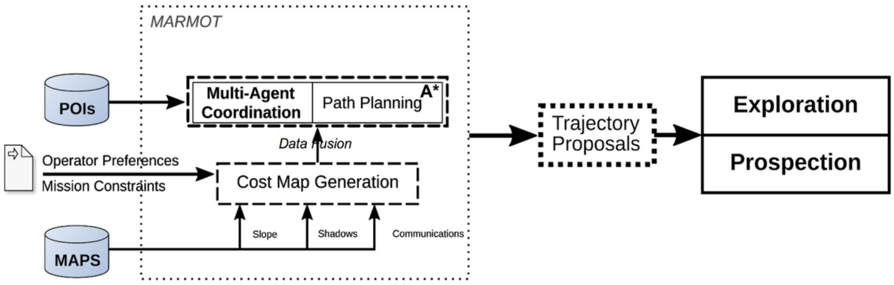


Figure 7: Layout of data pipeline and algorithm structure.

### 4.2 Software Package Description

MARMOT is a software deployment that aims to help human operators. It consists of three stand-alone support packages, Graphery, Mapstery, and Utils. The first two packages were built separately from MARMOT to act as dependencies and for utilization in other projects. Graphery performs general graph-theoretic network graphing, while Mapstery generates cost maps. The Utils component of MARMOT is not a separate package, but it comprises a set of generic tools for manipulating general data that companies and researchers may find useful.

Our aim in creating these packages tools was broad generalizability and extensibility with other open-source projects, such as QGIS [21], who would like to employ similar approaches for multi-agent missions. Additionally, MARMOT is built from many open source libraries such as *gdal* from OSGeo [24], *NetworkX* [15], *OpenCV* [8], and *NumPy* [23]. Figure 8 provides a basic overview of MARMOT's software structure.

DOI: 10.13009/EUCASS2019-813



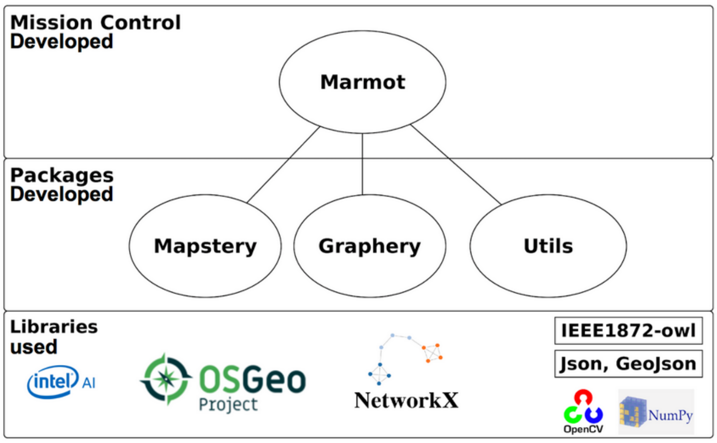


Figure 8: MARMOT presents three main components Mapstery: Map fusion and cost map generation, Graphery: Graph generation and path planning, Utils: general purpose tools for data formalization and manipulation

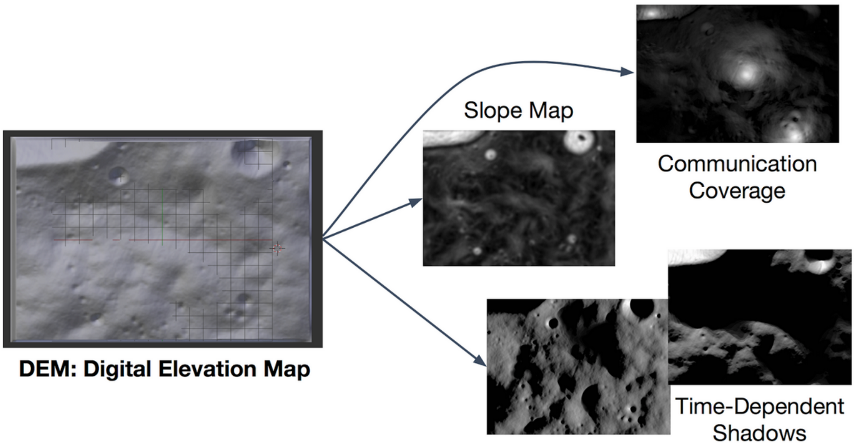


Figure 9: Using a Digital Elevation Model (DEM), viewed in Blender on the left, multiple cost maps can be generated.

### 4.3 Mapstery

Figure 9 presents an illustration of how a DEM is used to make the slope, shadow, and communication maps. Before the layered cost map is generated, mission objectives or preferences may require certain maps to take preference over other maps. The user weighing the information provided by each map can result in different path planning strategies. For instance, if shadowed regions are to be avoided at all cost, a higher weight is associated with the shadow map relative to the communication and slope maps. The layered and modified cost map is thus visualized as a linear combination of individual maps,

$$CostMap = \alpha \cdot M_{shadows} + \beta \cdot M_{slope} + \gamma \cdot M_{comm} \tag{1}$$

where $\alpha$, $\beta$, and $\gamma$ are the user-defined weights, and $M_{shadows}$, $M_{slope}$, and $M_{comm}$ are the respective map associated to each weight.

### 4.4 Graphery

Creation of grid-graphs are done primarily through the Graphery Python package. Given a requisite map size, Graphery will generate the appropriate size blank grid-graph for use with the layered cost map provided by Mapstery. The graph allows users to modify or create certain attributes (or costs) on-the-fly.

DOI: 10.13009/EUCASS2019-813



### 4.5 Utils

Additionally, utility functions were created to ease MARMOT's use in existing codebases. Waypoints, tasks, and other mission primitives are loaded using JSON format files. Visualization tools were developed for users who wish to visualize the changes in non-web frontend formats using matplotlib[22].

### 4.6 Coordination and Trajectory Proposals

#### 4.6.1 Point-of-Interest Selection

For a given Lunar mission, numerous points-of-interest exist for exploration; however, very limited operation time by rovers make assessing each of these points intractable. With MARMOT, we develop a means of generating a visit order of waypoints for a subsets of POIs that are explorable given mission constraints. The geographical coordinates of a POIs are input into a list of coordinate-lists, where POIs of equal visit importance reside together. Each coordinate-list constitutes a strata of importance for the mission. The algorithm then selects one coordinate from each coordinate-list to generate a rover's path, up to $N$ user-defined points.

Additionally, the algorithm optimizes the selection based on a distance function. The more POIs the rover can assess in a shorter distance, the better the path is considered. Numerous permutations of the paths are provided to enable the user to make the final selection. Additionally, variations of this algorithm were provided to allow start or end points, or use of different distance functions i.e. Euclidean distance or $A^*$ search using the fused cost-maps mentioned earlier.

#### 4.6.2 Multi-agent Optimal Path Planning and Synchronization

MARMOT utilizes staple graph search algorithms such as $A^*$ search to effectively generate path trajectories over the terrain for each agent. However, the system's capabilities were extended to use more modern search algorithms to effectively produce optimal trajectories while still being respectful of mission constraints. Distributed Path Consensus (DPC) [5] and its supplementary work, Distributed Path Consensus with Tasks (DPCT) [6], are a modified form of $A^*$ search that provides the ability to optimize each agent's search path by iteratively applying soft constraints on their individual paths while satisfying a user-defined mutli-agent constraint.

In particular, this user-defined constraint is a cost function added to the $A^*$ algorithm's heuristic function. The cost function produces a corrective penalty or reward to each iteration of this modified $A^*$ search. Moreover, an incremental weight is applied to the cost function to improve its efficacy after each subsequent path search. In the Lunar prospection scenario, maintaining communication between rovers is crucial to a successful mission. One can define an acceptable distance rovers must maintain during traversal, and each iteration of the algorithm will result in a path which maintains the distance constraint. Figure 10 show the first and last iteration for two agents using the DPC algorithm.

## 5. Experimental Study

To illustrate the capabilities of MARMOT, an initial mission scenario was formulated using a three agent configuration consisting of a lander, a relay rover, and a scientific rover. Operators at mission control direct the science rover to initiate a prospection task within a PSR. Moreover, this region happens to be a within crater making communication to the lander impossible. The relay rover will thus act as the intermediary to establish proper communication link between the lander and the scientific rover.

Figure 11 presents the above example with trajectories generated by MARMOT. In this case, the overall terrain cost map favored avoiding the shadow regions by placing a higher weight on the individual shadow map before MARMOT fused them with the other maps. The path planning generated by MARMOT is an $A^*$ heuristic search algorithm that avoids shadows while maintaining communication.

Figure 12 presents a slightly modified version of the mission scenario, but less weight is applied to the shadow map. Additionally, the path planning generated by MARMOT is an opportunistic solution. This algorithm adds intermediary points-of-interest for a relay and scientific rover to target as they progress towards their end goal. Once those points-of-interests are chosen, the $A^*$ search algorithm is used to calculate the best path between the waypoints.

DOI: 10.13009/EUCASS2019-813



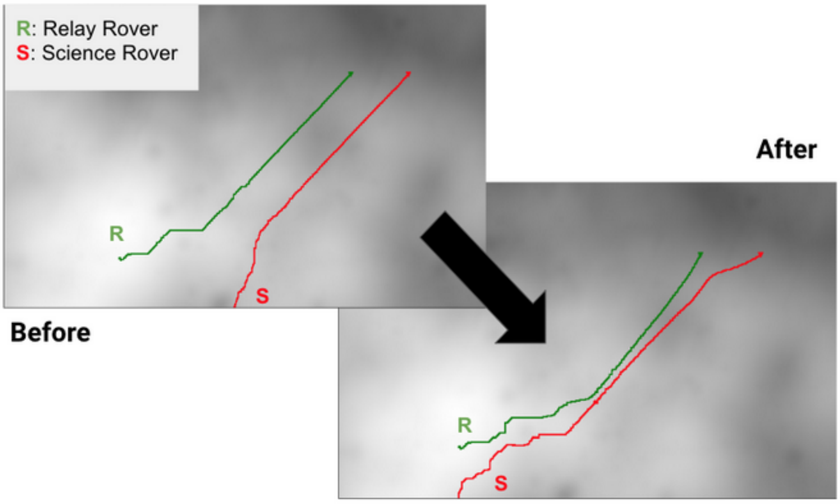


Figure 10: DPCT Applied to Static and Dynamic Agents. Applying soft penalty constraints allow path generation/modification that maintain constant communication between a science and relay rover as they proceed to different tasks.

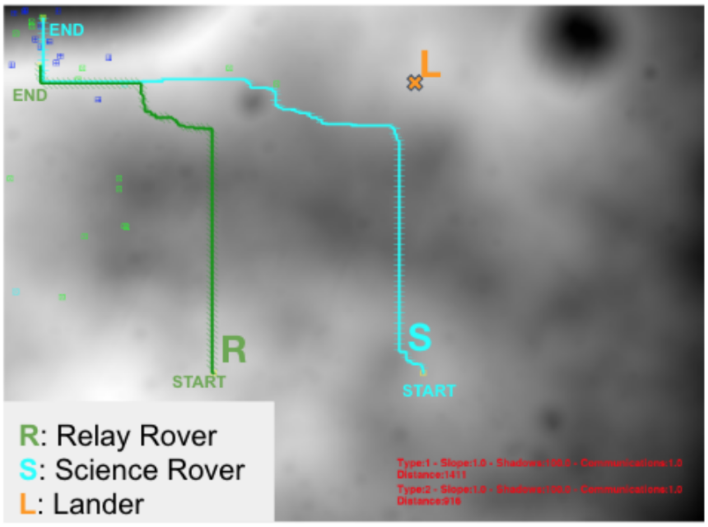


Figure 11: Case 1: Relay rover extends lander communication along the crater border while science rover explores a point-of-interest in a crater without communications to the lander. A fused cost map is used for trajectory calculation.

## 6. Conclusion and Future work

MARMOT's purpose is to build a set of open source tools for small space companies and researchers interested in developing space exploration tools for multi-agent mission planning operations. MARMOT provides a framework for users to increase the automation capabilities of multi-agent operations in space exploration and prospecting missions. Users can adjust to various operation and mission requirements via custom map and waypoint input. By combining cost map fusion, waypoint prioritization, and multi-agent trajectory planning, the current iteration of the work enables global planning for multi-agent systems. Future work seeks to give MARMOT local planning capabilities where sensory input and environmental feedback allows agents to retain more complex information during exploration. Furthermore, a comprehensive local or global simulation environment would lay groundwork for development of algorithms that utilize modern machine learning paradigms such as reinforcement learning and object identification. MARMOT's subsequent development would thus enable a higher degree of automation from the global to local scale.

DOI: 10.13009/EUCASS2019-813



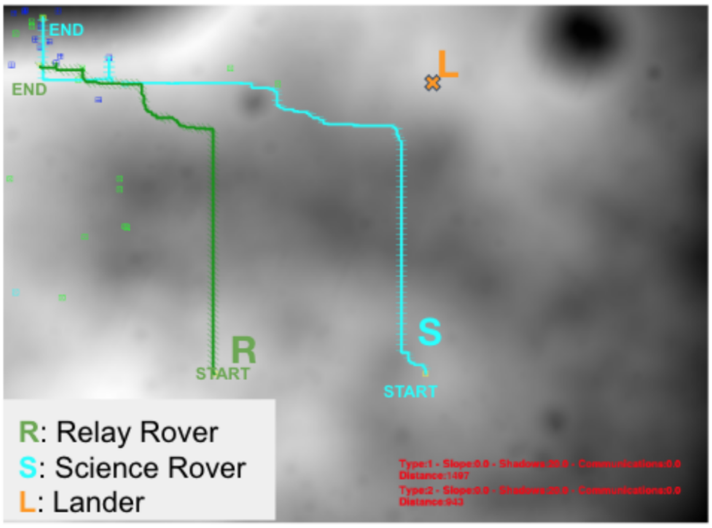


Figure 12: Case 2: The rovers goals are the same as in Case 1. A set of traversable points-of-interest are suggested for extending the prospecting mission.

## 7. Acknowledgments

The authors would like to acknowledge the contributions and support of the NASA Frontier Development Lab, Intel, X Prize, and SpaceResource.LU.

## References

[1] Daniel R Andrews. Resource prospector (rp)-early prototyping and development. In *AIAA SPACE 2015 Conference and Exposition*, page 4460, 2015.

[2] D. Bapna, M. Martin, and W. Whittaker. Earth-moon communication from a moving lunar rover. *Proceedings of the 42nd International Instrumentation Symposium*, pages 5–9, May 1996.

[3] Peter Berkelman, Mei Chen, Jesse Easudes, John Hancock, Martin C. Martin, Andrew Mor, Eric Rollins, Alex Sharf, Jack Silberman, Tom Warren, and Deepak Bapna. Design of a day / night lunar rover. Technical Report CMU-RI-TR-95-24, Carnegie Mellon University, Pittsburgh, PA, June 1995.

[4] Robert Besancon. *The encyclopedia of physics*. Springer Science & Business Media, 2013.

[5] S. Bhattacharya, M. Likhachev, and V. Kumar. Distributed path consensus algorithm. Technical Report MS-CIS-10-07, University of Pennsylvania, July 2010.

[6] S. Bhattacharya, M. Likhachev, and V. Kumar. Multi-agent path planning with multiple tasks and distance constraints. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 953–959, 2010.

[7] R. Boumghar, S. Lacroix, and O. Lefebvre. An information-driven navigation strategy for autonomous navigation in unknown environments. *Safety Security and Rescue Robotics*, 2011.

[8] G. Bradski and A. Kaehler. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000.

[9] Joseph Carsten, Arturo Rankin, Dave Ferguson, and Anthony Stentz. Global path planning on board the mars exploration rovers. In *2007 IEEE Aerospace Conference*, pages 1–11. IEEE, 2007.

[10] A. Colaprete. Resource prospector mission goals and landing site constraints. *SSERVI: Lunar Science for Landed Missions Workshop*, 2018.

[11] M. Colby, L. Yliniemi, and K. Tumer. Autonomous multi agent space exploration with high-level human feedback. *Journal of Aerospace Information Systems*, pages 301–315, Apr 2016.

DOI: 10.13009/EUCASS2019-813




[12] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2019.

[13] NASA FDL. Nasa frontier development lab. `https://frontierdevelopmentlab.org`. Accessed: 2019.

[14] G. Gutin and A.P. Punnen. The traveling salesman problem and its variations. *Springer Science and Business Media*, 12, 2006.

[15] A. Hagberg, D. Schult, and P. Swart. Exploring network structure, dynamics, and function.

[16] Grant H Heiken, David T Vaniman, and Bevan M French. Lunar sourcebook-a user's guide to the moon. *Research supported by NASA,. Cambridge, England, Cambridge University Press, 1991, 753 p. No individual items are abstracted in this volume.*, 1991.

[17] NASA JPL. Moon trek. `https://trek.nasa.gov/moon/index.html`. Accessed: 2018-7-30.

[18] D.V. Lu, D. Hershberger, and W.D. Smart. Layered costmaps for context-sensitive navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 709–715, Sep 2014.

[19] E. Mahoney. Resource prospector. `https://www.nasa.gov/resource-prospector`. Accessed: 2018.

[20] A. Pal and R. Tiwari. Communication constraints multi-agent territory exploration task. *Applied Intelligence*, 3(38):357–383, April 2013.

[21] QGIS Development Team. Qgis. `https://www.qgis.org`. Accessed: 2018.

[22] The Matplotlib Development Team. matplotlib. `https://matplotlib.org`. Accessed: 2019.

[23] S.V. Walt, S.C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, 13(2):22–30, March 2011.

[24] F. Warmerdam. The geospatial data abstraction library. *Open Source Approaches in Spatial Data Handling*, pages 87–104, 2008.