2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Exploration of Large Outdoor Environments Using Multi-Criteria Decision Making

Hannah Lehner[1] Martin J. Schuster[1] Tim Bodenmüller[1] Rudolph Triebel[1]

Abstract— **We present a Multi-Criteria Decision Making (MCDM) framework specifically designed for planetary exploration. Our work is based on PROMETHEE II, which allows operators to add task-specific criteria and conditions. We extended this algorithm to improve its resource usage by reducing the number of candidate exploration goals that have to be evaluated and compared. This is crucial when given a large number of goals, as is typical for outdoor environments. In addition, we identified five different criteria for planetary exploration, including a novel criterion that we call Direction of Interest (DOI), and use a categorization of these for further resource optimizations. We thereby ensure that the CPU usage of our decision making method can meet the limited budget of a space rover. We present simulated and real-world experiments with the Lightweight Rover Unit (LRU) and show a reduction of the processing time for decision making of approx. 70%.**

I. Introduction

An autonomous agent exploring expansive and hazardous environments on a foreign planet, e.g., as shown in Fig. 1, has to decide on a crucial question: *"where to go next?"* The challenge is to map as much scientific interesting terrain as possible, while keeping the localization uncertainty low and the map quality high – and all of that with limited resources. This demands careful decision making, considering several criteria and mission constraints – to prevent mission failures and to guarantee success. We present, to the authors' best knowledge, the first comprehensive autonomous exploration approach dealing with the special requirements of a planetary exploration mission. Our work is based on the Preference Ranking Organization METHod for Enrichment of Evaluations (PROMETHEE II) [1]. The well-known Multi-Criteria Decision Making (MCDM) approach evaluates multiple criteria and ranks exploration goals accordingly. We identify five exploration criteria especially designed to meet the challenges of large, unstructured, and hazardous environments. Our approach accounts for the limited energy and time budget of the robot by prioritizing exploration directions. By actively triggering loop closures, the algorithm keeps the localization uncertainty within limited bounds. The exploration performance is maximized, by considering the costs to travel to a goal as well as the information to be gained at the goal.

Unlike in indoor scenarios, in large outdoor environments, the number of exploration goals is growing fast with each exploration step (see Fig. 2). To reach a decision on the next goal, the rover needs to consider hundreds of exploration goals, which is a CPU-intensive operation. As the CPU resources of a real space rover are very limited, one essential task is to prune the number of exploration goals considered in the comparison. We show how to extend PROMETHEE II in order to extract a subset of candidate exploration goals. Our algorithms calculates a threshold based on a single criterion. All exploration goals below this threshold can be neglected during the further decision process. In addition, to further speed up decision making, we introduce three classes of criteria: robot-dependent, map-dependent and environment-dependent criteria. This classification saves processing time, since our algorithm only needs to re-evaluate robot-dependent criteria for all exploration goals at each exploration step. For most goals, map- and environment-dependent criteria remain the same at each exploration step and thus do not need to be re-evaluated each time.

[1]German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Department of Perception and Cognition, Münchner Str. 20, 82234 Wessling, Germany {firstname.lastname} at dlr.de

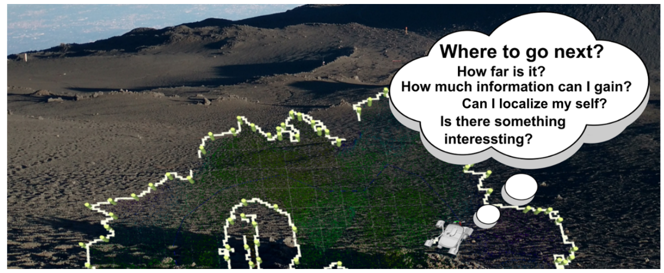


Fig. 1: Challenging analogue planetary exploration mission in the volcanic environment of Mt. Etna, Sicily, Italy. During exploration, the robot itself has to answer several questions to decide on *"where to go next?"*

Summarizing, our work has the following contributions:

- Extension of PROMETHEE II to reduce resource usage by classification of criteria in three categories and extraction of a subset of exploration goals
- Identification of five exploration criteria for large-scale outdoor environments
- Introduction of a new Direction of Interest (DOI) criterion that enables drive-by science
- Experimental evaluation on a space-rover prototype

II. Related Work

The central question during autonomous exploration is: *"where to go next?"* The answer depends on the task and the criteria used to evaluate the exploration goals. The task of classical robotic exploration is to map an unknown environment as fast as possible [7]–[9]. Most authors consider two criteria: cost to go to an exploration goal and expected information gained when reaching an exploration goal. Integrated exploration strategies try to solve the problem of mapping, localization, and exploration at the same time. Apart from the cost and the expected information gain

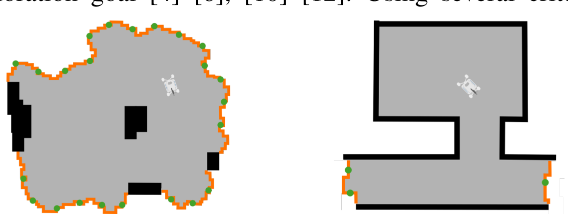
978-1-7281-9077-8/21/$31.00 ©2021 IEEE

		[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	**ours**
Scenario	Large-scale space	✓	✓	✓	–	–	✓	✓	–	✓	✓	✓	–	✓
	Outdoor environment	–	–	–	–	–	✓	–	✓	–	–	–	–	✓
	Realistic environment	–	–	✓	✓	–	–	✓	✓	–	–	–	✓	✓
	No a-priori knowledge	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓
Exploration	Exploration efficiency	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Re-localization perf.	–	–	✓	✓	✓	–	–	–	✓	✓	–	–	✓
	Revisiting behaviour	–	–	✓	–	–	–	–	–	✓	–	–	–	✓
	Prio. areas/directions	✓	✓	–	–	–	–	–	–	–	–	✓	✓	✓
Dec. Making	MCDM	✓	✓	–	–	–	–	–	–	–	–	–	–	✓
	Adapt criteria/weights	✓	✓	–	–	–	–	–	–	–	–	–	–	✓
	Variable # of criteria	✓	✓	–	–	–	–	–	–	–	–	–	–	✓
	Criteria classification	–	–	–	–	–	–	–	–	–	–	–	–	✓
	Subset of goals	–	–	–	–	–	–	–	–	–	–	–	–	✓

TABLE I: Table comparing autonomous exploration approaches.

criteria, the approaches of [4]–[6], [10], [11] consider the re-localization performance when deciding for the next best exploration goal. To re-localize itself, the robot visit known locations, therefore, we include re-visiting goals similar to the approaches of Sim and Roy [10] and Stachniss et al. [4].

Directed exploration strategies [2], [3], [12], [13] consider that the environment is not of uniform interest. Both Basilico and Amigoni [3] and Taillandier and Stinckwich [2] prioritize areas where a communication to the operator is guaranteed and Li et al. [12] and Dang et al. [13] apply visual saliency to prioritize areas in a search and rescue scenario. We prioritize a direction pointing to a point of interest (POI) instead of a region of interest (ROI). This direction can be computed by only considering 2D Information, e.g., of a camera image. Prioritizing a ROI, requires prior knowledge [12] or the 3D structure of the environment [2], [3]. Exploring large outdoor environments differs from exploring small structured indoor environments. In outdoor environments the robot is confronted with several exploration goals all leading into different directions, whereas in indoor environments the number of exploration goals and possible directions is limited. As stated in Tab. I, most exploration approaches [2]–[6], [8], [10]–[13] are not developed for outdoor environments and only tested in bounded indoor environments. In contrast to the related work, our approach is a combination of classical exploration, integrated exploration, and directed exploration. We consider the exploration efficiency (cost, information gain), the re-localization performance (loop closure likelihood, loop closure impact) and prioritize a direction leading to POIs (direction of interest). To handle several criteria, we believe a multi-criteria decision making approach should be applied to decide on *"where to go next?"* Most exploration strategies found in literature apply fixed utility functions to combine the exploration criteria evaluating an exploration goal [4]–[6], [10]–[12]. Using several criteria,



(a) Outdoor exploration: large frontiers with many candidate goals (b) Indoor exploration: small frontiers with few candidate goals

Fig. 2: Comparison of autonomous exploration in typical large unstructured outdoor environments and small structured indoor environments.

a utility function gets complex, even for a human expert. Only few authors [2], [3] address the use of general MCDM approaches to decide on *"where to go next?"* Basilico and Amigoni [3] first introduced MCDM for exploration. They exploit the Choquet fuzzy integral to compute global utilities. This approach allows to integrate redundant criteria. Similar to our work, Taillandier and Stinckwich [2] utilize the PROMETHEE II method. The advantage of our approach is that the difference of the criterion values is used to estimate the global utility, instead of the criterion values themselves. This allows to model criteria with non-linear importance, e.g., our novel criterion, the direction of interest. Comparing hundreds of goals as found in large outdoor environments for several criteria is a CPU-intensive operation. Both Basilico and Amigoni [3] and Taillandier and Stinckwich [2] do not treat the processing time of a MCDM. We present an extension to PROMETHEE II to prune the number of goals to be compared and categorize our criteria into three classes to reduce the number of criteria to be computed at each exploration step. We conduct experiments in a large outdoor environment with a real space-rover prototype (LRU [14]). In contrast, Basilico and Amigoni [3] and Taillandier and Stinckwich [2] test their approach in an indoor environment and in simulation only.

III. Multi-Criteria Decision Making

To decide on *"where to go next?"*, we apply the Preference Ranking Organization METHod for Enrichment of Evaluations, better known as PROMETHEE II [1], a well-studied Multi-Criteria Decision Making framework, which outputs a complete ranking of all alternatives. In this section, we briefly introduce the basic PROMETHEE II algorithm and then point out our own extensions and contributions to it.

A. Brief Introduction to PROMETHEE II

Let A be a set of alternative exploration goals, i.e., $A = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\}$, and $C = \{c_1, c_2, ..., c_m\}$ be a set of criteria to rate an exploration goal. We can think of the exploration goals as vectors $\mathbf{a} \in \mathbf{SE}(3)$ of potential robot poses and the criteria as real-valued functions $c : \mathbf{SE}(3) \to \mathbb{R}$, where higher values denote a higher preference for the particular goal regarding the respective criterion. Hence, let

$$d_k(\mathbf{a}_i, \mathbf{a}_j) = c_k(\mathbf{a}_i) - c_k(\mathbf{a}_j) \quad \forall k = 1, \ldots, m \tag{1}$$

denote the difference between two goals $\mathbf{a}_i$ and $\mathbf{a}_j$ with respect to criterion c_k. Further, let $P_k : \mathbb{R} \to [0, 1]$ be a function that maps d_k to the unicriterion preference degree

$$\pi_k(\mathbf{a}_i, \mathbf{a}_j) = \begin{cases} 0 & \text{if} \quad d_k(a_i, a_j) < 0 \\ P_k(d_k(\mathbf{a}_i, \mathbf{a}_j)) & \text{else} \end{cases} \tag{2}$$

Examples for preference functions are piecewise linear functions or Gaussian functions (see Tab. II). The multi-criteria preference degree π is defined by the sum of all weighted unicriterion preference degrees π_k:

$$\pi(\mathbf{a}_i, \mathbf{a}_j) = \sum_{k=1}^{m} \pi_k(\mathbf{a}_i, \mathbf{a}_j) \cdot w_k \tag{3}$$

with the weights $w_k \in [0,1]$ and $\sum_{k=1}^{m} w_k = 1$.

We can now define the positive and negative *multi-criteria flow* ϕ^+ and ϕ^- for each exploration goal $\mathbf{a}_i$ as

$$\phi^+(\mathbf{a}_i) = \frac{1}{n-1} \sum_{\mathbf{x} \in A \setminus \mathbf{a}_i} \pi(\mathbf{a}_i, \mathbf{x}) = \sum_{k=1}^{m} \phi_k^+(\mathbf{a}_i) w_k \tag{4}$$

$$\phi^-(\mathbf{a}_i) = \frac{1}{n-1} \sum_{\mathbf{x} \in A \setminus \mathbf{a}_i} \pi(\mathbf{x}, \mathbf{a}_i) = \sum_{k=1}^{m} \phi_k^-(\mathbf{a}_i) w_k \tag{5}$$

with the positive *unicriterion net flow* as

$$\phi_k^+(\mathbf{a}_i) = \frac{1}{n-1} \sum_{\mathbf{x} \in A \setminus \mathbf{a}_i} \pi_k(\mathbf{a}_i, \mathbf{x}). \tag{6}$$

and the negative *unicriterion net flow* $\phi_k^-(\mathbf{a}_i)$ analogous. Finally, the *multi-criteria net flow* $\phi(\mathbf{a})$ is given by

$$\phi(\mathbf{a}_i) = \phi^+(\mathbf{a}_i) - \phi^-(\mathbf{a}_i) = \sum_{k=1}^{m} \phi_k(\mathbf{a}_i) w_k \tag{7}$$

with the *unicriterion net flow*

$$\phi_k(\mathbf{a}_i) = \phi_k^+(\mathbf{a}_i) - \phi_k^-(\mathbf{a}_i) \quad \text{with} \quad \phi_k(\mathbf{a}_i) \in [-1, 1]. \tag{8}$$

$\phi_k(\mathbf{a})$ observes how one goal is outranking all other goals for a single criterion and gives the operator relevant information to understand decisions. It thus is useful information for operators who want to adapt the criteria and their weights. The goal with the highest rank $\mathbf{a}^\star$ is given by

$$\mathbf{a}^\star = \underset{\mathbf{a} \in A}{\operatorname{argmax}}\, \phi(\mathbf{a}). \tag{9}$$

B. Calculation of a Threshold of Relevance

To reduce processing time, our approach prunes the number of exploration goals for the most time-consuming step of unicriterion net flows calculation for each goal. We select a subset of exploration goals based on a threshold in the unicriterion net flow $\phi_k(\cdot)$ of one criterion. All goals above the threshold are set to be in the candidate subset, whereas all goals below the threshold can be neglected during the further decision process. All other criteria are only computed and compared for the goals in the candidate subset.

Let us assume that the exploration goal $\mathbf{a}_i$ has the largest unicriterion net flow for criterion c_m, i.e.,

$$\phi_m(\mathbf{a}_i) = \phi_m^{max} \; > \; \phi_k(\mathbf{a}_j) \quad \forall \mathbf{a}_j \in A \setminus \mathbf{a}_i\,, \tag{10}$$

and the smallest possible value $\phi_k(\mathbf{a}_i) = -1$ for all other criteria. Further, let $\mathbf{a}_j$ be another goal that has the highest possible value in all other criteria, i.e., $\phi_k(\mathbf{a}_j) = 1$ for $k \neq m$. Hence, $\mathbf{a}_j$ can only be higher ranked than $\mathbf{a}_i$ if $\phi(\mathbf{a}_j) > \phi(\mathbf{a}_i)$. Substitution of Eq. 7, splitting the values for c_m, as well as substitution of the known net flow values results in

$$\sum_{k=1}^{m-1} 1 \cdot w_k + w_m \phi_m(\mathbf{a}_j) > \sum_{k=1}^{m-1} -1 \cdot w_k + w_m \phi_m^{max}\,. \tag{11}$$

Solving the relation for the minimum $\phi_m(\mathbf{a}_j)$ results in the threshold t_m for unicriterion net flow c_m:

$$t_m = \frac{1}{w_m} \left(\sum_{k=1}^{m-1} (-w_k) + w_m \cdot \phi_m^{max} - \sum_{k=1}^{m-1} w_k \right) \tag{12}$$

All exploration goals with $\phi_m(\cdot) < t_m$ cannot have the highest multi-criteria net flow value and hence are omitted. The threshold t_m depends on the criteria weights w_m and the unicriterion net flow ϕ_m^{max} of the criterion c_m chosen to calculate the threshold. To prune many goals, the weight of the threshold criterion c_m should be quite large and there should exist a dominant goal for criterion c_m, i.e., $\phi_m(\mathbf{a}_i) \approxeq 1$. In case there exists no criterion with a strong weight and dominant goal, we suggest to combine several criteria for the threshold calculation despite the increase of the computational burden.

Note that although our method guarantees that the best goal is within the subset, excluding goals from the estimation of the multi-criteria preference degree can theoretically change the final ranking. However, this is unlikely and only possible when the difference in the multi-criteria preference degree of two goals is very small or too many goals were excluded from the estimation. In all of our experiments, approx. 70% of all goals are in the subset and we never observed a change in the ranking using our algorithm.

C. Criteria Classification

Computing the criteria for all goals takes most processing time. Especially criteria based on the 3D structure of the environment often require complex and time-consuming calculations. To further speed up the computation, we classify them into three classes. These describe how often a criterion has to be recomputed for an existing exploration goal.

Robot-dependent criteria: Criteria values change when the robot moves and thus have to be recomputed at each exploration step. Examples for robot-dependent criteria are the distance or the required energy to go to a goal.

Map-dependent criteria: Criteria values change when the robot map changes and thus have to be recomputed when the robot travels close to the location of the exploration goal or when the global map is updated, e.g., by loop closure optimization. An example for a map-dependent criterion is the information gained when reaching a goal.

Environment-dependent criteria: Criteria values never change as they only depend on the environment itself, which we assume to be static in space exploration. Thus, these criteria have to be evaluated only once, when a new goal is generated. To relax the assumption that the environment is static and the criteria never change, we provide a manual trigger to update the criteria values. An example for an environment-dependent criterion is the scientific interest of the environment at the goal location.

IV. Criteria for Large Outdoor Environments

Exploring large unstructured and unbounded environments requires the robot to carefully decide on *"where to go next?"* We propose five criteria to rank exploration goals, especially designed to meet the challenges of space exploration. In the following, we describe them in general, and in Sec. V-B, we state their implementation as used in our experiments.

A. Direction of Interest Criteria

Our novel criterion, the Direction of Interest (DOI) c_{doi}, can be used to indirectly prioritize areas. E.g., if the robot or an operator detects something of interest, the DOI guides the robot to this spot. This also allows to set up a drive-by science missions. For this, an operator defines several locations of interest that the robot should visit. At the beginning, the DOI points to the direction of the first location and the robot starts exploring until the location is reached. Next, the DOI is switched to the next location. While driving from one location to another, the robot explores and decides autonomously for the next best local exploration goal.

The DOI criterion $c_{d_{oi}}(\mathbf{a}_i)$ is defined as the angle between the directional vector $\mathbf{d}_{oi}$ pointing into the current DOI and the directional vector $\mathbf{g}_i$ pointing from the current robot position to the goal $\mathbf{a}_i$.

$$c_{d_{oi}}(\mathbf{a}_i) = \cos^{-1}\left(\frac{\mathbf{d}_{oi} \circ \mathbf{g}_i}{|\mathbf{d}_{oi}| \cdot |\mathbf{g}_i|}\right) \tag{13}$$

The direction of interest $\mathbf{d}_{oi}$ can either be set by a human operator or by the robot itself.

B. Cost Criteria

The cost criterion $c_{cost}(\mathbf{a}_i)$ describes the costs to pay to reach an exploration goal from the current robot position. We define the costs as the length len of the shortest traversable path W from the current robot position $\mathbf{p}_r$ to the goal $\mathbf{a}_i$.

$$c_{cost}(a_i) = len(W(\mathbf{a}_i, \mathbf{p}_r)) \tag{14}$$

C. Information Gain Criteria

The information gain criterion $c_{IG}(\mathbf{a}_i)$ is a measure of how much new information the robot will gain when arriving at the goal. The IG is the sum of the differences between the current entropy $H(x)$ and the expected entropy $E[H'(x)]$ at each voxel $x \in \mathcal{X}(\mathbf{a}_i)$, where $\mathcal{X}$ is the voxel space in the neighborhood of goal $\mathbf{a}_i$. The current entropy $H(x)$ and the expected entropy $E[H'(x)]$ are calculated according to [15].

$$c_{IG}(\mathbf{a}_i) = IG = \sum_{x \in \mathcal{X}(\mathbf{a}_i)} H(x) - Q_v(x) \cdot E[H'(x)] \tag{15}$$

Optionally, one can take into account the probability of observability $Q_v(x)$, which decreases with an increasing distance from the 3D sensor [16].

D. Active Loop Closure Criteria

While exploring the environment, the localization errors of the robot accumulate over time, which leads to an inaccurate map. By forcing a loop closure, the robot can re-localize and optimize the map. To evaluate the re-localization performance of a goal, we suggest to use the two criteria: loop closure likelihood $c_{ll}(\mathbf{a}_i)$ and loop closure impact $c_{li}(\mathbf{a}_i)$. The loop closure likelihood $c_{ll}(\mathbf{a}_i)$ is the probability that a loop closure l between the new goal $\mathbf{a}_i$ and a goal visited in the past b_z generated at exploration step z can be calculated.

$$c_{ll}(\mathbf{a}_i) = p(l(\mathbf{a}_i, b_z)) \tag{16}$$

The loop closure impact $c_{li}(\mathbf{a}_i)$ is a measure for the expected reward of the loop closure for the localization accuracy, i.e., to which extend the loop closure can reduce the current uncertainty s_g of the global map and global robot pose. The loop closure impact is then the difference between the current uncertainty and the expected uncertainty $E(s_g)$ after optimization when closing the loop.

$$c_{li}(\mathbf{a}_i) = s_g - E(s_g(l)) \tag{17}$$

V. EXPERIMENTS

We evaluated the proposed approach, both in simulation and real-world on our Lightweight Rover Unit (LRU) [14]. In the following, we describe the experiment setup in detail.

A. System Architechture

The LRU is equipped with an *Inertial Measurement Unit (IMU)* and a stereo camera system, which is used for dense depth image calculation and visual odometry. For local pose estimation, a local reference filter fuses visual odometry measurements with the IMU measurements. To generate a global consistent 3D map, we apply a 6D global localization and mapping system based on submaps [17]–[19]. The SLAM graph is optimized by matching submaps and estimating the relative transformation between pairs of submaps.

Our work is based on our integrated exploration strategy [16] with active loop closing. During each exploration step, the algorithm generates exploration goals and revisiting goals. At so-called frontiers to unknown space, we sample several new exploration goals. The revisiting goals are located at existing submaps, more precisely at the middle of the bounding box of the obstacle point cloud of the submap. To answer the question *"where to go next?"*, we apply in all experiments the approach presented in this paper. All computations are performed on-board with an Intel Core i7-3740QM CPU (2.70 GHz).

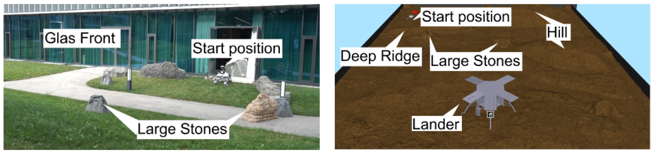
B. Criteria

We apply all criteria described in Sec. IV. Tab. III shows the weights used during the experiments.

Cost c_{cost}: Shortest traversable path from the robot to the goal. To calculate this path, we apply a simple wavefront planner. We apply a linear preference function (Tab. II) with $q = 5$, $r = 30$.

Information Gain c_{IG}: To estimate the IG, we simulate a horizontal 360 degree sensor swipe with a fixed angular resolution and a 40 degree vertical FoV [16]. We calculate the IG applying Eq. 15. All measurements in a range of 4m are included and the map resolution is 10cm. We apply a linear preference function (Tab. II) with $q = 0.1$, $r = 0.8$.

Loop Closure Likelihood c_{ll}: Likelihood that the intended submap match to generate a loop closure will succeed. We use a heuristic measure to describes the matching quality



(a) Real outdoor environment (b) Simulation environment

Fig. 3: Experimental setup for our real outdoor experiments (a) and our experiments conducted in simulation (b)

of the submap at goal $\mathbf{a}_i$. It accounts for the distribution of feature points d_o, computed by a nearest neighbor analysis, and the ratio between the number of feature points n_o in a submap and the total number of points n in a submap [16].

$$c_{ll}(\mathbf{a}_i) = \frac{n_o}{n} + d_o \tag{18}$$

A uniform distribution of feature points and a large number of feature points compared to all points increase the loop closure likelihood. We apply a linear preference function (Tab. II) with $q = 0.2$, $r = 0.8$.

Loop Closure Impact c_{li}: Expected reward of the submap match for localization accuracy and map quality. We use the marginalized uncertainty between the submap at the current robot position $s^2(\mathbf{p}_r)$ and the submap at the goal $s^2(\mathbf{a}_i)$ as measure for the loop closure impact.

$$c_{li}(a_i) = s^2(\mathbf{a}_i) - s^2(\mathbf{p}_r) \tag{19}$$

We apply a linear preference function with $q = 0.2$, $r = 0.8$.

Direction of interest $c_{d_{oi}}$: Is set by the operator before the exploration starts. We set the $\mathbf{d_{oi}} = (1, 0, 0)$ for all our experiments and calculate $c_{d_{oi}}$ via Eq. 13. As $c_{d_{oi}}$ depends on an angle, which effect increases with larger distance values to goals, we apply a Gaussian Preference function (Tab. II) with $\sigma = 0.6$ to model the criteria.

C. Experiment Environment

Simulation: We use the RoverSimulationToolkit [20] for simulating the LRU. The simulation environment, shown in Fig. 3, measures approximately 40 m $\times$ 40 m and contains a steep ramp, a deep ridge, and several rocks.

Real-World: We conducted real-world experiments with our LRU. In Fig. 3, we give an impression of our setup. The environment to be explored is about 60 m $\times$ 100 m.

D. Experiment Task

For all experiments, the task for the rover is to explore an unbounded large area within 10 min. The parameters used are stated in Tab. III. Experiment 1 evaluates the extensions of PROMETHEE II to reduce computation time. We compare the performance of the basic PROMETHEE II algorithm [2] and our extended decision making process. The subset is extracted based on the cost criterion c_{cost}. The Experiments 2 examine the effect of our criteria, particular our new criterion, the direction of interest. For this, we run the exploration mission with different sets of parameters stated in table Tab. III. When applying the active loop closure criteria, we observe the positional covariance of the robot during the exploration. If the covariance is larger than $s^2(p_r) > 0.3$ the criteria weights are automatically changed in favor of the active loop closure criteria, see Tab. III.

Preference Function	
Linear: $P(d) = \begin{cases} 0 \text{ if } d \leq q \\ \frac{d-q}{r-q} \text{ if } q < d \leq r \\ 1 \text{ if } d > r \end{cases}$	Gaussian: $P(d) = \begin{cases} 0 \text{ if } d \leq 0 \\ 1 - e^{-\frac{d^2}{2 \cdot \sigma^2}} \text{ if } d > 0 \end{cases}$

TABLE II: Preference functions used for the experiments.

Experiment	Setup	Criteria				
		Robot-dependent			Map-dependent	
		c_{cost}	c_{doi}	c_{li}	c_{ll}	c_{IG}
1	S	0.55	0.2	0.05	0.05	0.15
2a	R	0.2	0.6	-	-	0.2
2b	R	0.4	0.4	-	-	0.2
2c	S	0.3	0.4	0.05	0.05	0.2
2d	S	0.3	0.1	0.2	0.3	0.1

2c: $s^2(\mathbf{p}_r) < 0.3$, 2d: $s^2(\mathbf{p}_r) > 0.3$, R=Real world, S=Simulation

TABLE III: Table stating the criteria parameters used for the experiments.

VI. Results and Discussion

Experiment 1: The focus of experiment 1 is the reduction in computation time. Fig. 4 compares the mean computation time for the processing steps: criteria calculation, unicriterion net flow calculation, and multi-criteria net flow calculation for an exploration experiment with and without our extensions. The subset extraction can speed up the decision process by up to 30 %. Fig. 7 shows that, independent of the total number of alternative goals, the subset contains about 70%(+-4,3%) of all goals. The single processing steps in Fig. 4 show that criteria evaluation is computationally most expensive. While the evaluation of the robot-dependent criteria (cost and DOI) is fast, the map-depended criteria (information gain and loop closure likelihood) take a long time. Applying the criteria classification saves 60% of the computation time. While the computation time reduction using the subset extraction is constant over time, the effect of the criteria classification increases with progressing exploration, see Fig. 7. Without the criteria classification, all criteria have to be evaluated for all exploration goals (green) at each exploration step. Using our criteria classification, the map- and environment-dependent criteria only have to be recomputed for new exploration goals and exploration goals located in areas the robot traversed, i.e., where the map changed (blue). Combining both, the criteria classification and the subset extraction, enables us to reduce the mean processing time of decisions by approx. 70%. The computation time stated in Fig. 4 is the mean over one exploration experiment stopped after 10 min. During our experiments, the minimum number of exploration goals was four and the maximum number was 94. We expect the number of exploration goals to further

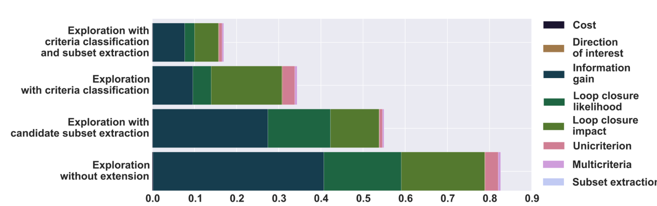


Fig. 4: Run time statistics showing the effect of our PROMETHEE II extensions & criteria classification: mean processing times for each step calculated from one exploration run that lasted 10 min. The processing times of the cost criteria and direction of interest criterion, as well as the time to extract the candidate subset are below 1 ms and thus not visible in the plot.

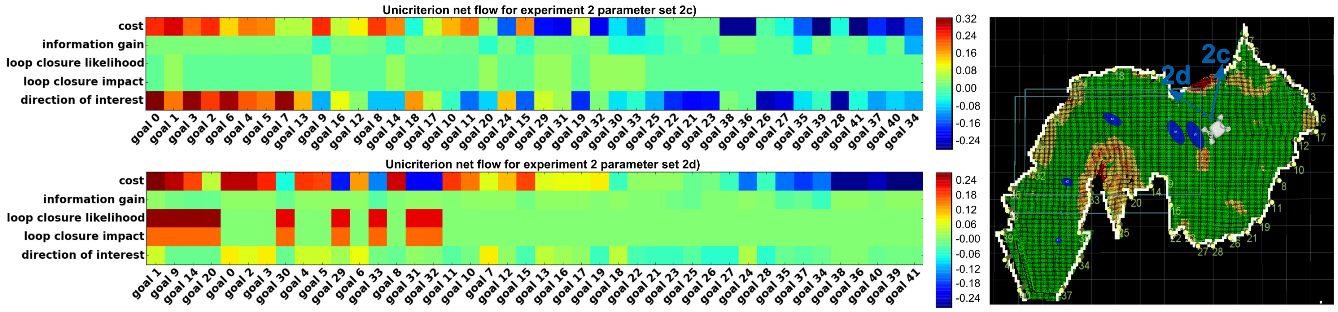


Fig. 5: Weighted unicriterion net flow $\phi_k(\cdot)w_k$ values for all goals of the exploration step shown in the map (left). The goals in the colormaps are sorted by their ranking (left best, right worst). At the shown exploration step, $s^2(p_r) > 0.3$ holds and the parameters are switched from 2c to 2d (see Tab. III).

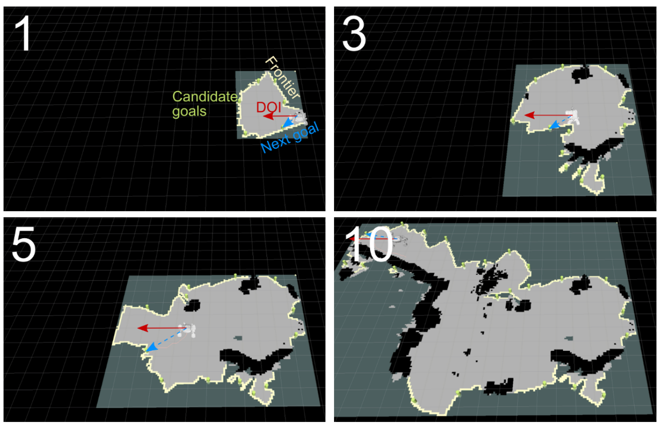


Fig. 6: Exploration steps of experiment 2a conducted in the real world outdoor environment shown in Fig. 3a. The images show the 2D occupancy grid map at each exploration step with the frontiers (yellow) and the exploration goals (green) at the frontiers. The direction of interest is marked with a red vector and the vector pointing to the next goal is colored in blue.

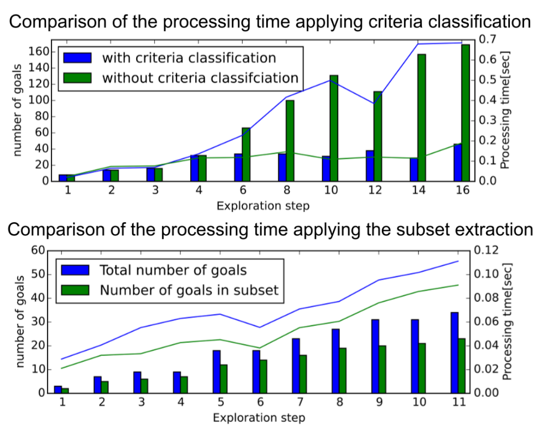


Fig. 7: Comparison of the processing times (lines) of the whole decision process with and without our extensions, in an experiment applying our criteria classification (top) and our subset extraction (bottom). Note, using the criteria classification, the number of goals (bars) sampled at a frontier were increased compared to the exploration applying the subset extraction.

increase for a real space exploration mission: Only reducing the computation time allows to apply enhanced decision making using a MCDM method such as PROMETHEE II.

Experiments 2: These experiments evaluate the suggested criteria applied to an exploration in a large outdoor environment. We conducted several experiments with four different parameter sets (see Tab. III, 2a - 2d). To evaluate the DOI criterion, we applied the parameter sets 2a and 2b in a real outdoor environment. Fig. 6 shows four exploration steps recorded while exploring with parameter set 2a. The DOI is set by the operator to $\mathbf{d_{oi}} = (1, 0, 0)$. With a strong weight on the $c_{d_{oi}}$ the rover favors goals lying towards the DOI. We found that applying the parameter set 2b, a balanced exploration is achieved. With the parameter sets 2c and 2d, we tested the exploration behavior applying all stated criteria. The map presented in Fig. 5 shows an exploration step where the parameters are switched from 2c to 2d as the positional covariance of the robot $s^2(\mathbf{r}_p)$ is larger than 0.3. The heatmap plots in Fig. 5 show the weighted unicriterion net flow values $\phi_k(\cdot)w_k$ using parameter set 2c (top) and 2d (bottom). The effect of the parameter change is clearly visible. Using set 2c, goals with a high value for c_{cost} and c_{doi} are favored, using set 2d, c_{doi} plays no role in the decision. The goals with the highest multi-criteria net flow are goals with a high value for c_{cost} and especially a high value for c_{ll} and c_{li}. At the final decision, the robot chooses to go to goal 1 instead of goal 0 in order to actively trigger a loop closure. This experiment impressively shows the advantage of using a MCDM: The exploration behavior can easily be adapted to the current mission requirements by adapting the criteria or their weights.

VII. Conclusion

We presented a Multi-Criteria Decision Framework designed to meet the challenges of autonomous exploration in space. To answer the central question *"where to go next?"*, we identified five important criteria. By prioritizing a direction, we guide the robot in a large environment, by actively triggering loop closures, we keep the localization uncertainty within limited bounds, and by considering the costs to go to a goal and the information gained at a goal, we maximize the exploration performance. To meet the limited resources of space rovers, we showed how to reduce the computation time of PROMETHEE II by approx. 70%: To reduce the number of evaluations, we introduce criteria classification and to reduce the number of goals to be compared, we extend PROMETHEE II to extract a subset of exploration goals. As the parameterization of the approach can be elaborate we plan to learn the exploration parameters in future work.

Acknowledgment

This work was supported by the Helmholtz Association, project ARCHES (contract number ZT-0033).

REFERENCES

[1] J.-P. Brans and P. Vincke, “Note—A Preference Ranking Organisation Method: (The PROMETHEE Method for Multiple Criteria Decision-Making),” *Management Science*, vol. 31, no. 6, pp. 647–656, 1985.

[2] P. Taillandier and S. Stinckwich, “Using the PROMETHEE multi-criteria decision making method to define new exploration strategies for rescue robots,” in *SSRR*, 2011, pp. 321–326.

[3] N. Basilico and F. Amigoni, “Exploration strategies based on multi-criteria decision making for searching environments in rescue operations,” *Autonomous Robots*, vol. 31, no. 4, p. 401, 2011.

[4] C. Stachniss, G. Grisetti, and W. Burgard, “Information Gain-based Exploration Using Rao-Blackwellized Particle Filters.” in *RSS*, 2005, pp. 65–72.

[5] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, “Information Based Adaptive Robotic Exploration,” in *IROS*, 2002, pp. 540–545.

[6] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, “An Experiment in Integrated Exploration,” in *IROS*, 2002, pp. 534–539.

[7] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, “Evaluating the Efficiency of Frontier-based Exploration Strategies,” *ISR Robotik*, 2010.

[8] H. H. Gonzalez-Banos and J.-C. Latombe, “Navigation Strategies for Exploring Indoor Environments,” *IJRR*, vol. 21, no. 10-11, pp. 829–848, 2002.

[9] S. Wirth and J. Pellenz, “Exploration Transform: A stable exploring algorithm for robots in rescue environments,” in *SSRR*, 2007, pp. 1–5.

[10] R. Sim and N. Roy, “Global A-Optimal Robot Exploration in SLAM,” in *ICRA*, 2005, pp. 661–666.

[11] R. Valencia, J. V. Miró, G. Dissanayake, and J. Andrade-Cetto, “Active Pose SLAM,” in *IROS*, 2012, pp. 1885–1891.

[12] A. Q. Li, R. Cipolleschi, M. Giusto, and F. Amigoni, “A semantically-informed multirobot system for exploration of relevant areas in search and rescue settings,” *Autonomous Robots*, vol. 40, no. 4, pp. 581–597, 2016.

[13] T. Dang, C. Papachristos, and K. Alexis, “Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics,” in *ICRA*, 2018, pp. 2526–2533.

[14] M. J. Schuster, S. G. Brunner, K. Bussmann, S. Büttner, A. Dömel, M. Hellerer, H. Lehner, P. Lehner, O. Porges, J. Reill, S. Riedel, M. Vayugundla, B. Vodermayer, T. Bodenmüller, C. Brand, W. Friedl, I. Grixa, H. Hirschmüller, M. Kaßecker, Z.-C. Márton, C. Nissler, F. Ruess, M. Suppa, and A. Wedler, “Towards Autonomous Planetary Exploration: The Lightweight Rover Unit (LRU), its Success in the SpaceBotCamp Challenge, and Beyond,” 2017.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, 2005.

[16] H. Lehner, M. J. Schuster, T. Bodenmüller, and S. Kriegel, “Exploration with active loop closing: A trade-off between exploration efficiency and map quality,” in *IROS*, 2017, pp. 6191–6198.

[17] M. Schuster, K. Schmid, C. Brand, and M. Beetz, “Distributed stereo vision-based 6D localization and mapping for multi-robot teams,” *Journal of Field Robotics (JFR)*, October 2018.

[18] M. J. Schuster, C. Brand, H. Hirschmüller, M. Suppa, and M. Beetz, “Multi-Robot 6D Graph SLAM Connecting Decoupled Local Reference Filters,” in *IROS*, 2015.

[19] C. Brand, M. J. Schuster, H. Hirschmüller, and M. Suppa, “Submap Matching for Stereo-Vision Based Indoor/Outdoor SLAM,” in *IROS*, 2015, pp. 5670–5677.

[20] M. Hellerer, M. J. Schuster, and R. Lichtenheldt, “Software-in-the-Loop Simulation of a Planetary Rover,” in *i-SAIRAS*, 2016.