

# **Task-Oriented Exploration: A Multi-Criteria Decision Making Approach for Robotic Exploration**

**Hannah, Lehner**

Vollständiger Abdruck der von der TUM School of Computation, Information and  
Technology der Technischen Universität München zur Erlangung einer

**Doktorin der Naturwissenschaften (Dr.rer.nat.)**

genehmigten Dissertation.

**Vorsitz:** Prof. Dr.-Ing Darius Burschka

**Prüfer\*innen der Dissertation:**

1. Priv.-Doz Dr. Rudolph Triebel
2. Prof. Dr. Teresa Vidal Calleja,  
University of Technology Sydney

Die Dissertation wurde am 29.06.2022 bei der Technischen Universität München eingereicht  
und durch die TUM School of Computation, Information and Technology am 31.10.2022  
angenommen.



# Abstract

In robotic planetary exploration missions, robots are deployed to autonomously explore and map the large and unstructured environments of planetary surfaces. While a robot should be able to execute a mission task mainly autonomously, for space exploration missions, it is important to have the opportunity to observe and adapt the robotic exploration task. Operators and scientists require to supervise the robot at the available communication time slots and understand the decisions made by the robot. For this we propose a generalized concept for robotic exploration based on Multi-Criteria Decision Making (MCDM) to model, implement and conduct exploration tasks. Our general formulation supports scientists by designing the autonomous exploration behavior of a robot to reach specific missions goals. In robotic exploration tasks, robots repeatedly decide where to move next. We define locations at the boundary to unknown areas - exploration goals - and locations in already visited areas - re-localization goals - to be the solution space of this decision problem. To model a certain exploration behavior, the goal locations are evaluated by a set of criteria and conditions. The criteria and condition values for each goal location are compared, applying a MCDM method to find the next goal location, which best matches the defined mission goal. Thereby, we introduce two novel multi-attribute utility functions and transfer the Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE II) to solve decision making in robotic exploration. To cope with the limited computational resources of space rovers, we extend the PROMETHEE II algorithm to decrease the required computational resources. Applying our generalized concept, we examine four exploration use cases, deduced from the Exploration Roadmap of the International Space Exploration Coordination Group (ISECG). In the first use case, the robot has to autonomously survey a region of interest. To tackle the trade-off between exploration efficiency and map quality, we implement an integrated exploration, which applies active loop closing to optimize an underlying SLAM graph. In our second use case, we implement a directed exploration to increase the scientific output while exploring a region of interest. It incorporates knowledge about the probability of detecting a feature of interest, i.e., a specific type of rock requested by the scientists. As our third use case, we implement an exploration behavior in the fashion of drive-by science, whereby the robot is directed to a predefined point of interest, while simultaneously gathering new information about the environment on its way. For our fourth use case, we apply the same concept to model a multi-robot exploration task, which coordinates a heterogeneous team of two robots. We demonstrate all four use cases on real or simulated space rover prototype hardware. In a total of more than sixty experiments, we evaluate our methods and analyze the implemented exploration behaviors.



# Zusammenfassung

Bei der Exploration von planetaren Oberflächen erkunden Roboter autonom fremde Umgebungen und bauen eine Karte dieser auf. Auch wenn ein Roboter in der Lage sein sollte, eine planetare Explorationsmission autonom durchzuführen, ist es unerlässlich, eine Möglichkeit zur Überwachung und gegebenenfalls zur Anpassung der Exploration zu haben. Operatoren und Wissenschaftler müssen den Roboter während der kurzen Zeitabschnitte, in denen eine Kommunikation zum Roboter aufgebaut werden kann, überwachen können. Wir schlagen ein allgemeines Konzept für robotische Exploration basierend auf multikriteriellen Entscheidungsverfahren vor, um verschiedene robotische Explorationsaufgaben zu modellieren, zu implementieren und auszuführen. Unsere allgemeine Formulierung ermöglicht es Wissenschaftlern, das Explorationsverhalten eines Roboters so zu gestalten, dass verschiedene zuvor definierte Missionsziele erreicht werden können. Bei der robotischen Exploration entscheidet ein Roboter immer wiederholend, welchen Zielpunkt er als Nächstes ansteuert. Der Lösungsraum dieses Entscheidungsproblems beinhaltet dabei Zielpunkte an der Grenze zu bis dahin unbekanntem Gebieten und Zielpunkte in Gegenden, die der Roboter bereits besucht hat. Um das Explorationsverhalten des Roboters zu modellieren, werden mehrere Kriterien und Konditionen über die Ziele ausgewertet. Um das nächst beste Ziel zu bestimmen, werden die Kriterien und Konditionen für die einzelnen Ziele mithilfe eines multikriteriellen Entscheidungsverfahrens verglichen. Wir stellen zwei neue multikriterielle Nutzenfunktionen vor und übertragen die bekannte Methode Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE II) auf das Entscheidungsproblem der robotischen Exploration. Die Rechenleistung eines Roboters, der für den Weltraumeinsatz konzipiert wurde, ist begrenzt. Darum erweitern wir PROMETHEE II, um die nötige Rechenleistung der Methode zu verringern. Wir leiten vier verschiedene Anwendungsfälle von der Exploration Roadmap der International Space Exploration Coordination Group (ISECG) ab und wenden unser allgemeines Konzept an, um diese Anwendungsfälle zu modellieren und zu untersuchen. Im ersten Anwendungsfall exploriert und vermisst der Roboter autonom eine unbekannte Region. Wir implementieren eine Explorationsstrategie, bei welcher aktiv nach Schleifenschlüssen im darunter liegenden SLAM Graphen gesucht wird, um diesen zu optimieren. Damit lösen wir den Konflikt zwischen einer effizienten Exploration und einer guten Kartenqualität. Im zweiten Anwendungsfall implementieren wir eine Explorationsstrategie, welche im "Vorbeifahren" wissenschaftliche Entdeckungen machen soll. Der Roboter muss eine vorgegebene Reihe von globalen Zielen nacheinander ansteuern und während der Fahrt zu den einzelnen Zielpunkten möglichst viele neue wissenschaftliche Informationen über die Region, welche er traversiert sammeln. Im dritten Anwendungsfall stellen wir eine Explorationsstrategie vor, welche es zum Ziel hat, den wissenschaftlichen Ertrag zu erhöhen, während der Roboter

## *Zusammenfassung*

eine Region exploriert. Dafür berücksichtigen wir die Wahrscheinlichkeit, ein Objekt von wissenschaftlichem Interesse an einem Ziel zu detektieren. Ein solches Objekt kann z. B. ein bestimmter Stein sein, der von Wissenschaftlern als wissenschaftlich relevant eingestuft wurde. In unserem fünften Anwendungsfall stellen wir eine Strategie vor, um ein heterogenes Team von Robotern zu koordinieren, welche gemeinsam eine Region explorieren. Dafür verwenden wir das gleiche allgemeine Konzept wie für die Anwendungsfälle, in denen nur ein Roboter alleine agiert. Wir demonstrieren alle vier Anwendungsfälle auf realen robotischen Systemen oder mithilfe einer Simulation. Insgesamt führen wir mehr als 60 Experimente durch, um alle Explorationsstrategien und unser allgemeines Konzept zu analysieren und zu validieren.

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Zusammenfassung</b>	<b>v</b>
<b>Contents</b>	<b>vii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>Acronyms</b>	<b>xiii</b>
<b>Math Symbols</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement and Objectives . . . . .	1
1.2 Scenario Description . . . . .	3
1.3 Challenges . . . . .	3
1.4 Approach . . . . .	5
1.5 Contributions . . . . .	8
1.6 Publications . . . . .	10
1.7 Outline . . . . .	12
<b>2 Exploration of Planetary Surfaces</b>	<b>15</b>
2.1 Concepts and Methods . . . . .	15
2.1.1 Multi-Criteria Decision Making . . . . .	15
2.1.2 Autonomous Exploration . . . . .	17
2.1.3 Online 3D Mapping . . . . .	20
2.2 Related Work . . . . .	22
2.2.1 Autonomous Exploration . . . . .	23
2.2.2 Drive-by Science . . . . .	24
2.2.3 Object Search . . . . .	24
2.2.4 Multi-robot Exploration . . . . .	26
<b>3 Exploration as Multi-Criteria Decision Making Problem</b>	<b>29</b>
3.1 Identification of the MCDM Problem in the Exploration Process . . . . .	29
3.2 General Formulation of Robotic Exploration as MCDM . . . . .	30

## CONTENTS

3.3	MCDM Methods for Robotic Exploration . . . . .	32
3.3.1	Multi-Attribute Utility Functions for an Integrated Exploration . .	33
3.3.2	Multi-Criteria Decision Making with Promethee II . . . . .	35
3.4	Summary and Discussion . . . . .	40
<b>4</b>	<b>Modeling Planetary Exploration Missions</b>	<b>43</b>
4.1	Sampling Goal Locations . . . . .	43
4.2	Use Case 1 - Autonomous Exploration . . . . .	45
4.2.1	Cost Criterion . . . . .	47
4.2.2	Information Gain Criterion . . . . .	47
4.2.3	Loop Closure Likelihood Criterion . . . . .	48
4.2.4	Loop Closure Impact Criterion . . . . .	49
4.3	Use Case 2 - Drive-by Science . . . . .	50
4.3.1	Direction of Interest Criterion . . . . .	51
4.4	Use Case 3 - Autonomous Search . . . . .	51
4.4.1	Features of Interest . . . . .	52
4.4.2	Detection of Features of Interest . . . . .	53
4.4.3	Feature of Interest Criterion . . . . .	54
4.5	Use Case 4 - Multi-robot Exploration . . . . .	57
4.5.1	Multi-robot Alignment Criterion . . . . .	58
4.5.2	Multi-robot Distance Criterion . . . . .	59
4.6	Criteria Classification . . . . .	59
4.7	Dissucssion and Summary . . . . .	60
<b>5</b>	<b>Implementation &amp; Integration</b>	<b>65</b>
5.1	Robot Hardware . . . . .	65
5.2	Exploration Framework . . . . .	66
5.3	Robot Simulation . . . . .	67
<b>6</b>	<b>Experimental Evaluation</b>	<b>71</b>
6.1	Use Case 1 - Integrated Exploration . . . . .	72
6.2	Use Case 2 - Drive-by Science . . . . .	77
6.3	Use Case 3 - Informed Exploration . . . . .	80
6.4	Use Case 4 - Multi-robot Exploration . . . . .	85
6.5	Multi-robot Exploration Field Tests at Mt. Etna . . . . .	89
6.6	Runtime . . . . .	90
6.7	Summary and Discussion . . . . .	92
<b>7</b>	<b>Conclusion</b>	<b>97</b>
7.1	Summary and Conclusion . . . . .	97
7.2	Future Work . . . . .	99
	<b>Bibliography</b>	<b>101</b>



# List of Figures

1.1	Space exploration use cases . . . . .	2
1.2	Martian environment . . . . .	5
1.3	Comparison of indoor and outdoor exploration . . . . .	6
1.4	Architecture overview . . . . .	7
2.1	MCDM stages . . . . .	16
2.2	Exploration process overview . . . . .	18
2.3	Schematic of frontiers . . . . .	19
2.4	Impressions of the underlying mapping system . . . . .	21
3.1	MCDM transfer to robotic exploration . . . . .	30
3.2	MCDM procedure for an autonomous exploration mission . . . . .	31
3.3	General illustration of a MCDM goal system . . . . .	32
3.4	Promethee II preference function . . . . .	37
4.1	Exploration goals . . . . .	44
4.2	Re-localization goals . . . . .	45
4.3	Illustration of use case 1 - autonomous exploration . . . . .	46
4.4	Goal system for use case 1 . . . . .	46
4.5	Wavefront path planning . . . . .	48
4.6	Drive-by science mission scenario . . . . .	50
4.7	Goal system for use case 2 . . . . .	50
4.8	Autonomous search mission scenario . . . . .	52
4.9	Goal system for use case 3 . . . . .	53
4.10	Features of interest on Mars . . . . .	54
4.11	Scientific camera . . . . .	55
4.12	Science product . . . . .	55
4.13	Polar histogram centered at the camera frame . . . . .	56
4.14	Multi-robot mission scenario . . . . .	57
4.15	Goal system for use case 4 . . . . .	58
4.16	Multi-robot distance function . . . . .	59
5.1	Lightweight rover units . . . . .	65
5.2	Exploration framework overview . . . . .	67
5.3	Simple simulation map example . . . . .	68
5.4	SiL simulator . . . . .	69
5.5	SiL simulator impressions . . . . .	70

## LIST OF FIGURES

6.1	Criteria hierachy FE . . . . .	71
6.2	Experiment environments: use case 1 . . . . .	73
6.3	Experiments use case 1: 3D translational error . . . . .	75
6.4	Experiments use case 1: mean number of actions . . . . .	76
6.5	Experiments use case 1: example occupancy grid maps . . . . .	76
6.6	Experiments use case 1: results outdoor environment . . . . .	77
6.7	Experiments use case 2: scenario . . . . .	78
6.8	Use case 2: exploration map examples . . . . .	79
6.9	Experiments use case 3: URSim landscape and setup . . . . .	81
6.10	Experiments use case 3: example of the sensor output . . . . .	82
6.11	Criteria hierarchy use case 3 simulation experiments . . . . .	82
6.12	Experiments use case 3: trajectories IFE . . . . .	83
6.13	Directed exploration action illustration . . . . .	84
6.14	Experiments use case 4: outdoor scenario . . . . .	85
6.15	Experiments use case 4: Results . . . . .	87
6.16	Experiments use case 4: 2D occupancy grid maps real world experiment . . . . .	88
6.17	Fiel tests - setup . . . . .	89
6.18	Field tests - 3D voxel map . . . . .	90
6.19	Runtime experiments: comparison of the mean run time . . . . .	91
6.20	Experiments use case 3: example of the sensor output . . . . .	92

# List of Tables

3.1	Preference functions . . . . .	38
4.1	Exploration criteria for robotic exploration missions . . . . .	62
6.1	Preference functions used in our simulation and real-world experiments . .	72
6.2	Experiments use case 1: Parameters . . . . .	74
6.3	Experiments use case 1: Results . . . . .	75
6.4	Experiments use case 3: Parameters . . . . .	81
6.5	Experiments use case 3: Results . . . . .	82
6.6	Experiments use case 4: Parameters . . . . .	86
6.7	Experiments use case 4: Parameters . . . . .	91



# Acronyms

AHP/ANP	Analytic Hierarchy/Network Process.
CALE	Combined Active Loop Closing and Exploration.
CPU	Central Processing Unit.
DE	directed exploration.
DLR	Deutsches Zentrum für Luft- und Raumfahrt.
DOI	Direction of Interest.
EKF	Extended Kalman Filter.
FOI	Feature of Interest.
FPGA	Field Programmable Gate Array.
GPS	Global Positioning System.
ICP	Iterative Closest Point.
IE	integrated exploration.
IFE	informed exploration.
IG	Information Gain.
ISECG	International Space Exploration Coordination Group.
LRU	Lightweight Rover Unit.
MADM	Multi-Attribute Decision Making.
MAUT/MAVT	Multi-Attribute-Utility/Value Theory.
MCDM	Multi-Criteria Decision Making.
MEPAG	Mars Exploration Program Analysis Group.
MODM	Multi-Objective Decision Making.
NAA	Naive Active Area.
POI	point of interest.
PROMETHEE	Preference ranking organization method for enrichment evaluation.

## Acronyms

PROMETHEE II	Preference ranking organization method for enrichment evaluation II.
ROI	Region of Interest.
ROS	Robot Operating System.
SALE	Separated Active Loop Closing and Exploration.
SaR	Search and Rescue.
SGM	Semi Global Matching.
SiL	Software in the Loop.
SLAM	Simultaneous Localization and Mapping.
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution.
ÉLECTRE	ELimination Et Choice Translating REality.

# Math Symbols

$A(t)$	active area at time t.
$C$	set of criteria.
$E[H'(\chi)]$	expected entropy.
$E[s_g(l)]$	expected uncertainty of the robot pose after a loop closure happend.
$F$	set of frontiers.
$F_c$	set of frontier cells.
$G$	set of goal locations.
$H(\chi)$	current entropy.
$P_c(X, Y, Z)$	reconstructed 3D certesian point in the camera frame.
$P_c^*$	reconstructed 3D point in the camera frame expressed in a spherical coordinate system.
$P_k : \mathbb{R} \rightarrow [0, 1]$	preference function.
$Q_o(\chi)$	propability of observability of a voxel.
$\alpha$	weight for balancing exploration and re-localization.
$\beta$	weight for balancing the liklehood and impact of re-localization.
$\chi$	voxel.
$\delta l_c$	propagated error on the depth component.
$\delta p$	image based error.
$\iota$	threshold explore or re-localize.
$\mathcal{D}$	distribution of the obstacle points in a submap.
$\mathcal{L}$	loop closure.
$\mathcal{X}$	set of voxel.
$\overline{d}_n$	distance to nearest neighbor.
$\phi^+(\mathbf{g})$	negative net-flow.
$\phi^+(\mathbf{g})$	positive net-flow.
$\pi(\mathbf{g}_i, \mathbf{x})$	multi-criteria preference degree.
$\pi_k(\mathbf{g}, \mathbf{x})$	unicriterion preference degree.
$\rho$	spherical coordinates radial distance component.
$\tau$	PROMETHEE II threshold for the subset extraction.
$\theta$	spherical coordinates azimuth component.
$\varphi$	spherical coordinates polar angle component.
$\mathbf{g}^*$	goal location with the highest rank.

## Math Symbols

$\mathbf{g}_{\text{ex}}$	exploration goal.
$\mathbf{g}_{\text{re}}$	re-localization goal.
$\mathbf{g} \in \mathbf{SE}(3)$	goal location.
$\mathbf{p}_r$	robot pose.
$\mathbf{v}_{\text{doi}}$	directional vector pointing from the robot to the direction of interest.
$\mathbf{v}_g$	directional vector pointing from the robot to a goal.
$\zeta$	past exploration step.
$a_s$	area of the bounding box of a submap.
$c : \mathbf{SE}(3) \rightarrow \mathbb{R}$	criterion.
$c_A$	multi-robot alignment criterion.
$c_D$	multi-robot distance criterion.
$c_{IG}$	information gain criterion.
$c_{\text{cost}}$	cost criterion.
$c_{\text{doi}}$	direction of interest criterion.
$c_{li}$	loop closure impact criterion.
$c_{ll}$	loop closure likelihood criterion.
$d_k(\mathbf{g}_i, \mathbf{g}_j)$	difference between two goal locations with respect to a criterion.
$fr$	frontier between unknown and known space in a 2D occupancy grid-map.
$l_c$	distance between a 3D Point in the camera frame and the origin of the camera frame.
$n$	number of points in a submap.
$n_o$	number of obstacle points in a submap.
$px$	pixel in an image.
$q$	indifference value.
$r$	preference value.
$s_g$	current uncertainty of the robot pose.
$t$	time.
$u(\mathbf{g})$	utility of visiting a goal location.
$u_1(\mathbf{g})$	utility of visiting a goal location after Stachniss et al. [1].
$u_2(\mathbf{g})$	utility of visiting a goal location after Gonzalez et al. [2].
$u_{\text{ex}}(\mathbf{g})$	utility describing the exploration performance of visiting a goal location.
$u_{\text{re}}(\mathbf{g})$	utility describing the re-localization performance of visiting a goal location.
$w_A$	weight of the multi-robot alignment criterion.
$w_D$	weight of the multi-robot distance criterion.
$w_{IG}$	weight of the information gain criterion.
$w_{\text{cost}}$	weight of the cost criterion.



$w_{doi}$	weight of the direction of interest criterion.
$w_{foi}$	weight of the feature of interest criterion.
$w_{li}$	weight of the loop closure impact criterion.
$w_{ll}$	weight of the loop closure likelihood criterion.



# 1 Introduction

To unlock clues about our solar system, the exploration of the Moon, Mars and asteroids are vital. As many destinations in space are inaccessible or too dangerous for humans, planetary exploration robotics is the key for future space applications. In this thesis, we investigate the robotic exploration problem of planetary surfaces, by which a robot has to repeatedly answer one important question: *where to move next?*

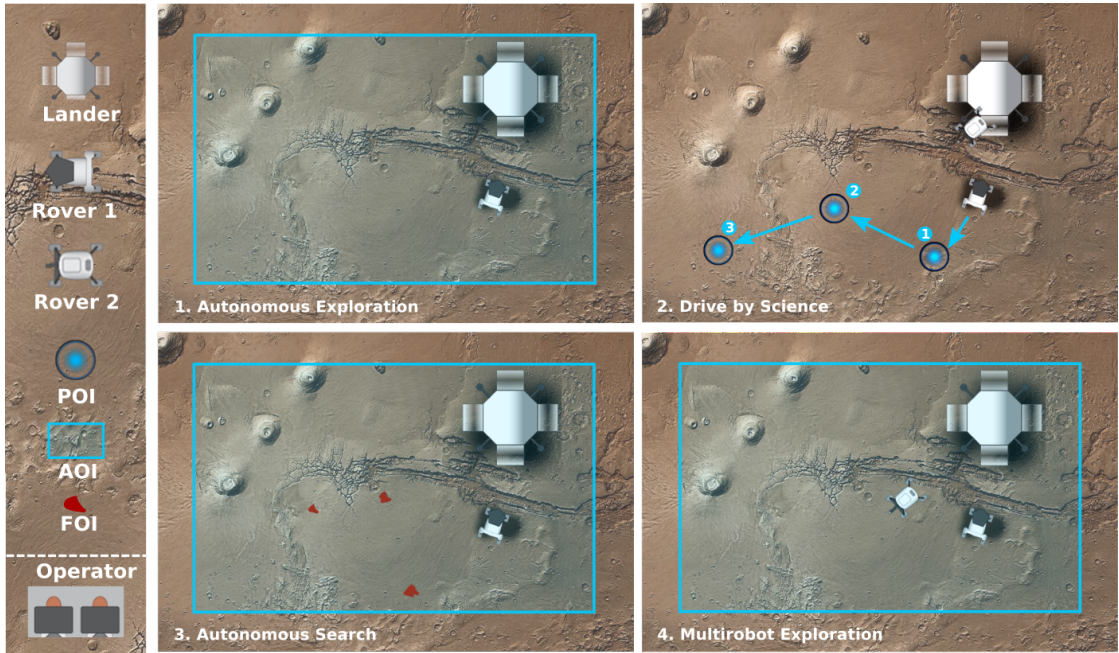
## 1.1 Problem Statement and Objectives

Traditionally, robotic investigations are operated remotely by human operators on Earth, which is difficult due to the delayed communication. To achieve most new scientific insights on a mission, a future robot should mainly act autonomously. For robotic exploration, that means deciding on its own *"where to move next?"*, to gain the most new information about the environment. The communication delay and small bandwidth, only allow for supervision by a human operator and for sending high-level commands. To enable supervision for robotic exploration tasks, we examine how to formulate robotic exploration as a general Multi-Criteria Decision Making (MCDM) Problem.

The answer to the question, *"where to move next?"*, depends on the mission objectives, i.e. varies for each exploration task. According to the Exploration Roadmap of the International Space Exploration Coordination Group (International Space Exploration Coordination Group (ISECG) ) [3] future robotic tasks include surveys and sample returns for science, as well as resource and environment assessment. We investigate four exploration use cases, which we illustrate in fig. 1.1:

1. **Autonomous exploration:** The survey of unknown areas is a basic task for a space rover. The goal is to build an accurate map of a Region of Interest (ROI), as illustrated in fig. 1.1, use case 1. The challenge is to explore, i.e. map the area, efficiently, while keeping a good map quality and an accurate self-localization. To ensure mission success, an accurate localization is vital to create a high-quality map, which can be used to plan further science investigations. In addition, to cope with the limited resources of a space rover, it is required to explore the environment efficiently.
2. **Drive-by science:** A common task for a space rover is to move to a point of interest (POI) defined by an operator. In a drive-by science mission, the goal is to maximize the information about the environment while moving towards the POI. Instead of moving on the direct path, the robot is allowed to take a longer, but

## 1 Introduction



**Figure 1.1:** Illustration of the four use cases we investigate in this thesis. Use cases 1-3 depict single robot use cases with one rover and use case 4 illustrates a multi-robot exploration. In use cases 1, 3 and 4 the exploration/search of/in a Region of Interest (ROI) is requested. In use case 2, a rover conducts exploration in a drive-by science fashion and has to visit several points of interest (POI) in a row.

more informative route. The challenge is to guide the robot in the desired direction while maximizing the information about the environment.

- 3. Autonomous search:** With an increasing level of autonomy, a future robotic task is to search for a Feature of Interest (FOI). For example, a rover has to find a certain geologic unit in an ROI, as illustrated in fig. 1.1, use case 3. The capability to find a FOI autonomously increases the scientific return of a space exploration mission significantly. Additionally, to explore a ROI the robot requires the capability to detect a FOI and to reason about its location. As the robot searches in an unknown environment, it is confronted with similar challenges as in use case 1. However, additionally a high search efficiency is desired.
- 4. Multi-robot exploration:** The key to more complex and efficient science investigations is teamwork. Multiple robots exploring a ROI together, lead to a significant increase in exploration efficiency. The challenge is to coordinate the robots, to most benefit from their cooperation.

Yamauchi et al. [4] formulate the central question of robotic exploration, as

*‘Given what you know about the world, where should you move to gain as much new information as possible?’*

This especially, describes the autonomous exploration task, we conduct in our first use case. However, we treat robotic exploration more generally and want to solve our other three use cases with an exploration approach as well. We reformulate the central question to be:

*‘Given what you know **and what you want to know** about the world, where should you move to gain as much new **valuable** information as possible?’*

Our formulation specifically addresses, the mission’s objectives and hence required exploration behavior to conduct a specific mission task.

In this thesis, we aim to transfer all four use cases to a robotic exploration problem and to model it with our proposed general exploration concept based on MCDM.

## 1.2 Scenario Description

To motivate our approach, we depict a robotic exploration mission to the surface of Mars, which is a compelling and accessible target for addressing the most crucial questions of our solar system and beyond [5]. The main scientific objectives and goals defined by the Mars Exploration Program Analysis Group (MEPAG) in the ‘Mars Science Goals, Objectives, Investigations and Priorities’ document [6] are: (1) Determine if Mars ever supported, or still supports, life; (2) understand the processes and history of climate on Mars; (3) understand the origin and evolution of Mars as a geological system; (4) prepare for human exploration. To find answers to these fundamental questions the composition of different rock samples or the distribution of rocks can give important clues. As illustrated in fig. 1.1 we consider two rovers that were brought together with a lander unit to the Martian surface. Both rovers can sense the environment and build with the retrieved sensor information a map of the environment. To localize itself within the map, the local sensor information is used. One of the rovers is equipped with a Scientific Camera and, the other with a robotic arm capable of taking rock samples. The robot with the scientific camera is assigned the tasks of the first three use cases. If a rock of interest is found in the third use case, the robot with the manipulator can be called to take a sample. The fourth use case is conducted by both robots together.

## 1.3 Challenges

In this section, we describe the challenges for robotic planetary exploration in general, and we identify the most compelling challenges for the four use cases described in the introduction, chapter 1. Thereby, we group the challenges into mission, environmental and system challenges.

### Mission Challenges

- **Communication:** The huge distance between an operator on Earth and a robot in space leads to delayed communication. Only short time slots to send a command

## 1 Introduction

or to receive a response are available. This leads to a planning and scheduling problem. An operator has to pre-plan investigations on the recorded images hours or a day earlier. Teleoperating or even commanding a rover from the Earth is a slow process due to limited communication bandwidth and delay.

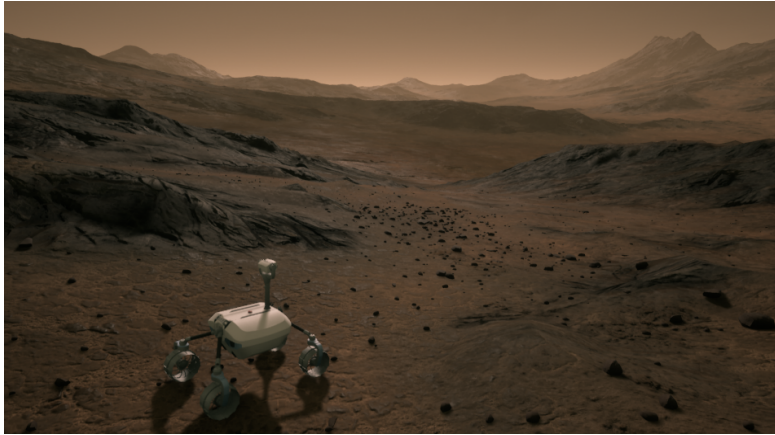
- **Changing mission requirements:** Depending on past discoveries the next mission task, e.g. one of the four use cases, for a robot is defined. On mission start, the operator defines the objectives and adjusts the exploration behavior accordingly. However, with the progressing mission, the situation and with that the requirements and objectives might change, which requires an adaptation of the exploration behavior. This is challenging itself but gets even more complex by considering the limited communication.
- **Conflicting objectives:** At each exploration action the robot has to decide, *where to move next?* To maximize the mission return, while avoiding mission failures, careful decision making is required. To guarantee mission success, especially for use cases 1 and 3, the robot has to re-localize itself by visiting previously sensed areas. This directly counteracts the main objective, gaining new information about the environment.

### Environmental Challenges

- **Large unstructured harsh environment:** Planetary surfaces are unstructured, large, and bear many hazards, as depicted in fig. 1.2. Deciding autonomously, *'where to move next?'*, becomes more complex the larger the area to be explored is, as more possible goal locations exist. The task of exploring large outdoor environments differs from the task of exploring bounded indoor environments, as shown in fig. 1.3. Unlike in indoor scenarios, in large outdoor environments the number of possible exploration actions grows fast with each exploration step. Whereas in indoor environments walls build a border and restrict the exploration directions, in outdoor environments large unbounded regions with only a few obstacles exist. To decide on the next goal, the rover needs to consider and compare hundreds of locations in opposite directions.
- **Scientific relevance:** The main goal of robotic exploration is, to maximize the information about an a priori unknown environment. However, in some use cases, e.g. use case 3, not any information about the environment but valuable information, i.e a certain FOI, is of interest. Thus, to fulfill the scientific objectives of a mission a robot has to prioritize a region to gain valuable information.

### System Challenges

- **Uncertainties:** How far is the next goal location? Is it safe to traverse the planned path? Is at this location really a FOI? These questions have to be considered when



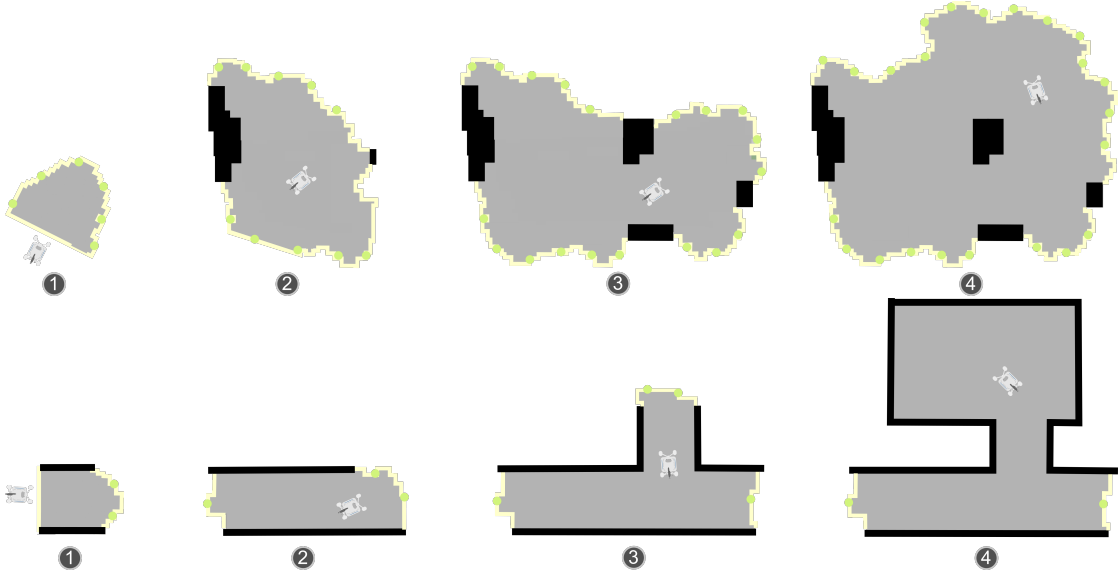
**Figure 1.2:** Photo-realistic simulation of a hazardous, large, and unknown martian-like landscape.

deciding on the next goal location. The local sensor information is the only source of information a robot has to make its decision, on ‘*where to move next?*’. This information is affected by sensor uncertainties, which propagate to the final decision and can lead to wrong decisions.

- **Limited hardware resources:** Robots for space exploration are complex systems, specially designed to withstand harsh conditions on planetary surfaces. The construction of such systems is restricted by these conditions and the fact that the robot is transported thousands of kilometers to its destination in space. Only a limited amount of Central Processing Unit (CPU) resources is available, which has to be distributed among thousands of processes running on an autonomous robot.
- **Supervision:** Although a higher level of autonomy can increase the scientific return of a planetary exploration mission, supervision by a human operator and scientists are important. In space, the final decision on, e.g. where to take a probe or where to explore is still in the hand of human scientists. Tight supervision, including observation and if necessary interference is requested by scientists. Decisive for scientists is also to understand the decision on the next exploration action made by an autonomous robot.
- **Test and Integration:** The development and test of exploration approaches is often limited by the availability of the hardware for space exploration missions. Field tests are expensive and allow no continuous test and development cycle.

## 1.4 Approach

To enable a human operator to model different exploration behavior, to conduct the tasks of the four use cases defined in the introduction, chapter 1, we present a general exploration concept based on MCDM. With our concept, it is possible to clearly outline a



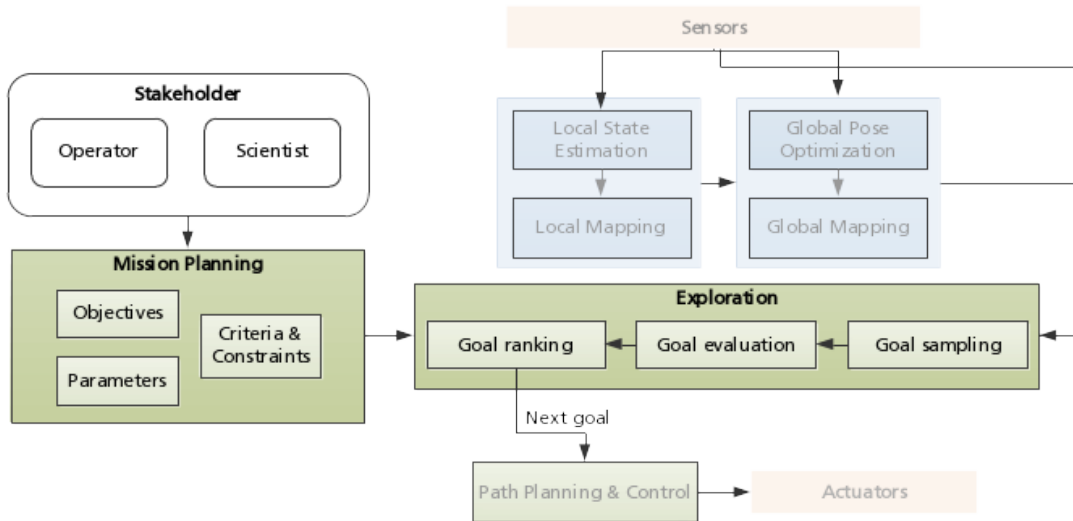
**Figure 1.3:** Illustration of frontiers (yellow) and exploration goals (green) for four consecutive exploration actions in an outdoor environment (top row) and an indoor environment (bottom row).

mission task as MCDM problem and transfer it to an exploration problem. By applying MCDM we offer a simple user interface for human operators to model, observe and adapt the exploration behavior of a robot in the short communication time slots between Earth and, for example, Mars.

In fig. 1.4, we present a general overview of the software architecture applied in this thesis and how the exploration is anchored. The mission objectives set by the stakeholders, in our case the operator and a scientist, define the exploration behavior. We achieve the required behavior, mainly by evaluating potential goal locations by different exploration criteria and constraints. An exploration criterion is any function that describes the quality of a goal location with respect to the objectives. For example, the distance to the goal location or the expected information at a goal location. An exploration constraint leads to an exclusion from the decision, of *‘where to move next?’*, if it does not hold for a goal location. For example, the operator can constrain the maximal distance to the next goal location to avoid a long traverse through the unknown environment. We use a state-of-the-art exploration approach: first, potential goal locations are sampled at the frontiers between the known and the unknown space, second, the goals are evaluated by the criteria and constraints and third, the goals are ranked. To rank the goals we apply multi-attribute utility functions and the well known MCDM Preference ranking organization method for enrichment evaluation II (PROMETHEE II) [7]. By applying PROMETHEE II we complete our general concept as it allows an operator to add or remove criteria online during an exploration process.

We apply our general concept to model the exploration behavior, required to conduct the tasks of the four use cases:





**Figure 1.4:** Overview of our architecture and integration of the robotic exploration. The stakeholders define in a mission planning phase the objectives, parameters, and exploration criteria. The exploration itself consists of the three steps: goal sampling, goal evaluation, and goal ranking. It depends on localization and mapping and requires a path planner to finally move towards the next goal location.

1. **Autonomous exploration:** We aim for a trade-off between a high exploration efficiency, a good map quality, and an accurate localization. For this, we present an integrated exploration approach, which actively triggers loop closures for pose optimization. Our integrated exploration is built upon a submap-based 6D SLAM system [8]. Loop closure constraints originate from pairwise submap matches, which allow the optimization of an underlying SLAM graph. During exploration, we employ the expected information gain as well as the robot’s localization uncertainty estimates to weigh exploration actions online. To evaluate goals with respect to loop closures, we introduce the two novel criteria loop closure impact and loop closure likelihood.
2. **Drive-by science:** To conduct a drive-by science mission with an exploration approach, we direct the robot towards the next POI, by incorporating the direction to the POI with our novel exploration criterion, which we call direction of interest criterion. To maximize the information on the environment, we consider the direction of interest criterion additionally to the information gained and the cost to move towards a goal.
3. **Autonomous search:** We introduce an informed exploration behavior, which incorporates the probability of detecting a predefined FOI when visiting a goal location. We equip the robot with the capability to detect a FOI in a camera image and store the direction towards a detected FOI in a robot-centric polar descriptor. Our main focus is on the exploration behavior, but additionally we

## 1 Introduction

present a preliminary approach for detecting different geologic units with a multi-spectral camera, which could be used in the future for detecting a FOI.

4. **Multi-robot exploration:** We present a leader-follower approach to coordinate two robots exploring a ROI together. The leader defines the exploration direction and the follower tries to drive in the same direction while keeping a certain distance from the leader. We achieve the following behavior by evaluating our novel exploration criteria *multi-robot alignment* and *multi-robot distance* for the goal locations of the follower robot. In our use case 4, the two rovers have different capabilities. One rover has a scientific camera and is able to detect a FOI, the other is equipped with a manipulator and can take a probe. Therefore, we choose an approach where both robots stay close to each other to complement each other when doing science investigations.

To cope with the limited CPU resources of a space rover, we introduce a classification of exploration criteria. The classification is based on the required update rate of a criterion for a goal location. In addition, we extend PROMETHEE II to reduce the runtime of the CPU-intensive outranking method.

Further, we adapted URSim[9] a Software in the Loop Simulator to be able to test and evaluate our exploration approach. It allows for short test and development cycles in photo-realistic environments, as shown in fig. 1.2.

## 1.5 Contributions

In this section, we state the contribution we accomplished to robotic exploration of planetary surfaces in general and on the four use cases we investigated in this thesis.

### Exploration as Multi-Criteria Decision Making Problem

- We formulate the exploration problem in general as a Multi-Criteria Decision Making problem and provide a concept on how to model different exploration behaviors to conduct various tasks.
- Based on the MCDM method PROMETHEE II we implemented an adaptable and flexible exploration framework, which allows an operator to model, observe and adapt the exploration behavior of a robot online.
- We reduce the high CPU resource consumption of PROMETHEE II by first extracting a subset of goal candidates, before evaluating the exploration criteria for all goals and pairwise comparison of all goals.
- We show how a simple classification of exploration criteria helps to reduce the runtime of an exploration behavior evaluating several criteria. Our classification divides the criteria in the three categories: robot-, map-, and environment-dependent, which is an indicator of how often a criterion value has to be recomputed for an existing goal location.

### Autonomous Exploration - Integrated Exploration

- We present an integrated exploration approach, which applies active loop closing in order to re-localize the robot, which improves its localization uncertainty. We define re-localization goals, at which it is likely to actively trigger a loop closure in the 6D graph SLAM system proposed by [8]. To decide whether the robot shall explore or revisit locations, we apply the current localization uncertainty estimate.
- We introduce two novel criteria, the loop closure likelihood and the loop closure impact, evaluating the likelihood and the impact of a potential loop closure.

### Drive-by Science - Directed Exploration

- We introduce the concept of directed exploration. By directing the robot into certain directions we enable an operator to use an exploration behavior to move towards a POI.
- We describe the new exploration criterion Direction of Interest (DOI), which evaluates if a goal is lying in the DOI defined by an operator. Combined with the information gain criterion, this allows us to conduct a drive-by science mission.

### Autonomous Search - Informed Exploration

- We introduce an informed exploration behavior, which enables the robot to efficiently find a FOI in a ROI. Instead of exploring the whole ROI the robot tries to detect a FOI in far range and prioritizes the direction, where the probability of finding the FOI is highest.
- We define the novel feature of interest criterion, which evaluates the probability of finding a FOI. We store the detection results in a polar descriptor, which is centered at the robot and describes the probability of finding a FOI in the direction of a goal location. Thereby, we consider the increasing distance error due to the stereo distance calculations.

### Multi-robot Exploration

- We present a leader-follower concept to coordinate two heterogeneous robots. The leader robot determines the exploration direction and the follower robot tries to move into the same direction while computing its own local optimal exploration goal. The approach is solely based on adding additional exploration criteria to the set of criteria of the follower robot and can be modelled with the same general exploration concept applied for the single robot use cases.

### Evaluation & Demonstrations

- We conducted in total more than 60 experiments to evaluate our different exploration behavior. Each use case presented in the introduction is demonstrated at

## 1 Introduction

least once on real space rover prototype hardware or in a high-fidelity Software in the Loop Simulator.

- Applying our integrated exploration, we could improve the mean localization error about 0.3m compared to the localization error occurring when applying an information gain based greedy exploration without active loop closing.
- We successfully conducted a drive-by science mission, where the robot visited three POI in a predefined order, applying our directed exploration behavior.
- We demonstrated the informed exploration and achieved a speed up of a factor of approximately 1.6 compared with an information gain based greedy exploration to find a FOI in an ROI.
- We demonstrated our multi-robot coordination approach in simulation, as well as on two real space rover prototypes. We show the intended leader’s follower-behavior and analyze its advantages and disadvantages.
- We proved that we could reduce the runtime with our extension to PROMETHEE II and our criteria classification approximately about 70%.

## 1.6 Publications

We published most contributions and methods described in this thesis already at international conferences. We cite the publications in the respective sections. In the following, we state a list of publications that contain key aspects of this thesis and publications that are related to this thesis.

### **First author publications on the key aspects of this thesis:**

- 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*, pages 6191–6198, 2017.
- H. Lehner, M. J. Schuster, T. Bodenmüller, and R. Triebel. Exploration of Large Outdoor Environments Using Multi-Criteria Decision Making. In *ICRA*, 2021.

### **First author publications on secondary aspects of this thesis:**

- M. Sewtz, H. Lehner, Y. Fanger, J. Eberle, M. Wudenka, M. G. Müller, T. Bodenmüller, and M. Schuster. URSim - A Versatile Robot Simulator for Extra-Terrestrial Exploration, March 2022.

The first authorship of the publication is shared between the first three authors. The author of this thesis contributed to the general concept of the simulation, as well as the robot integration and especially to the integration of the Lightweight Rover Unit (LRU).

- H. Kaufmann. Shadow-based matching for robust absolute localization during lunar landings. In *IEEE Aerospace Conference*, March 2014.

**Co-author publications related to this thesis:**

- M. J. Schuster, M. G. Müller, S. G. Brunner, H. Lehner, P. Lehner, R. Sakagami, A. Dömel, L. Meyer, B. Vodermayr, R. Giubilato, M. Vayugundla, J. Reill, F. Steidle, I. von Bargaen, K. Bussmann, R. Belder, P. Lutz, W. Stürzl, M. Smisek, M. Moritz, S. Stoneman, A. F. Prince, B. Rebele, M. Durner, E. Staudinger, S. Zhang, R. Pöhlmann, E. Bischoff, C. Braun, S. Schröder, E. Dietz, S. Frohmann, A. Börner, H.-W. Hübers, B. Foing, R. Triebel, A. O. Albu-Schäffer, and A. Wedler. The arches space-analogue demonstration mission: Towards heterogeneous teams of autonomous robots for collaborative scientific sampling in planetary exploration. *IEEE Robotics and Automation Letters*, 5(4):5315–5322, October 2020.
- M. J. Schuster, B. Rebele, M. G. Müller, S. G. Brunner, A. Dömel, B. Vodermayr, R. Giubilato, M. Vayugundla, H. Lehner, P. Lehner, F. Steidle, L. Meyer, K. Bussmann, J. Reill, W. Stürzl, I. von Bargaen, R. Sakagami, M. Smisek, M. Durner, E. Staudinger, R. Pöhlmann, S. Zhang, C. Braun, E. Dietz, S. Frohmann, S. Schröder, A. Börner, H.-W. Hübers, R. Triebel, B. Foing, A. O. Albu-Schäffer, and A. Wedler. The arches moon-analogue demonstration mission: Towards teams of autonomous robots for collaborative scientific sampling in lunar environments. In *European Lunar Symposium (ELS)*, 2020.
- M. J. Schuster, M. G. Müller, S. G. Brunner, H. Lehner, P. Lehner, A. Dömel, M. Vayugundla, F. Steidle, P. Lutz, R. Sakagami, et al. Towards heterogeneous robotic teams for collaborative scientific sampling in lunar and planetary environments. 2019.
- A. Wedler, M. Wilde, A. Dömel, M. G. Müller, J. Reill, M. Schuster, W. Stürzl, R. Triebel, H. Gmeiner, B. Vodermayr, K. Bussmann, M. Vayugundla, S. Brunner, H. Lehner, P. Lehner, A. Börner, R. Krenn, A. Dammann, U.-C. Fiebig, E. Staudinger, F. Wenzhöfer, S. Flögel, S. Sommer, T. Asfour, M. Flad, S. Hohmann, M. Brandauer, and A. O. Albu-Schäffer. From single autonomous robots toward cooperative robotic interactions for future planetary exploration missions. In *69th International Astronautical Congress (IAC)*, Proceedings of the 69th International Astronautical Congress (IAC). International Astronautical Federation (IAF), October 2018.
- 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. Vodermayr, T. Bodenmüller, C. Brand, W. Friedl, I. Grixia, 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.

- A. Wedler, M. Vayugundla, H. Lehner, P. Lehner, M. J. Schuster, S. G. Brunner, W. Stürzl, A. Dömel, H. Gmeiner, B. Vodermayr, B. Rebele, I. L. Grixia, K. Bussmann, J. Reill, B. Willberg, A. Maier, P. Meusel, F. Steidle, M. Smisek, M. Hellerer, M. Knapmeyer, F. Sohl, A. Heffels, L. Witte, C. Lange, R. Rosta, N. Toth, S. Völk, A. Kimpe, P. Kyr, and M. Wilde. First results of the robex analogue mission campaign: Robotic deployment of seismic networks for future lunar missions. In *68th International Astronautical Congress (IAC)*, volume 68 of *68th International Astronautical Congress (IAC)*. International Astronautical Federation (IAF), September 2017.

### 1.7 Outline

We describe our approach to planetary robotic exploration in seven chapters. In chapter 1 we introduce the topic, state the problem and state our approach and contributions. In chapter 2, we describe relevant concepts and methods to comprehend the thesis, as well as the related work. We explain our general exploration concept based on MCDM in chapter 3 and describe our approach for the four use cases in chapter 4. In chapter 5 and chapter 6 we state the implementation of our concept and methods and describe the experiments we conducted to evaluate our method. We summarize and conclude the thesis in chapter 7.

**Chapter 2 - Exploration of Planetary Surfaces:** In this chapter, we describe concepts and methods applied in this thesis and summarize the related work. It gives an introduction to the topics of autonomous exploration, mapping, navigation, and decision theory. We discuss the related work on autonomous single and multi-robot exploration as well as on visual search and opportunistic science.

**Chapter 3 - Exploration as Multi-Criteria Decision Making Problem:** we formulate the exploration problem as MCDM problem in general and specifically for planetary exploration missions. We identify the MCDM problem within the robotic exploration task and transfer the processes of one exploration action to it. We state, how we generate exploration goal locations, and how to rank the goal locations with a multi-attribute utility function or PROMETHEE II.

**Chapter 4 - Modeling Planetary Exploration Missions:** we model the exploration behavior for the four use cases introduced in section 1.1. In this chapter, we describe in detail the implementation of criteria used to model the required exploration behavior.

**Chapter 5 - Implementation & Integration:** we state the implementation of our exploration framework on the robotic system. We describe the real space rover prototype Hardware LRU used for the experiments and depict the software stack running on the system, which is required for the exploration. In addition, we present the Software in the Loop (SiL) Simulator URSim, which is used to conduct experiments in simulation.

**Chapter 6 - Experimental Evaluation:** we present and discuss the results of the experiments we conducted to evaluate our integrated exploration and our autonomous search, as well as the run time reduction on the decision making we achieved applying our criteria classification and extension to PROMETHEE II. Further, we depict the demonstration accomplished to show a drive-by science mission and multi-robot exploration.

**Chapter 7 - Conclusion:** we conclude by summarizing the thesis, and present challenges for future work.





## 2 Exploration of Planetary Surfaces

### 2.1 Concepts and Methods

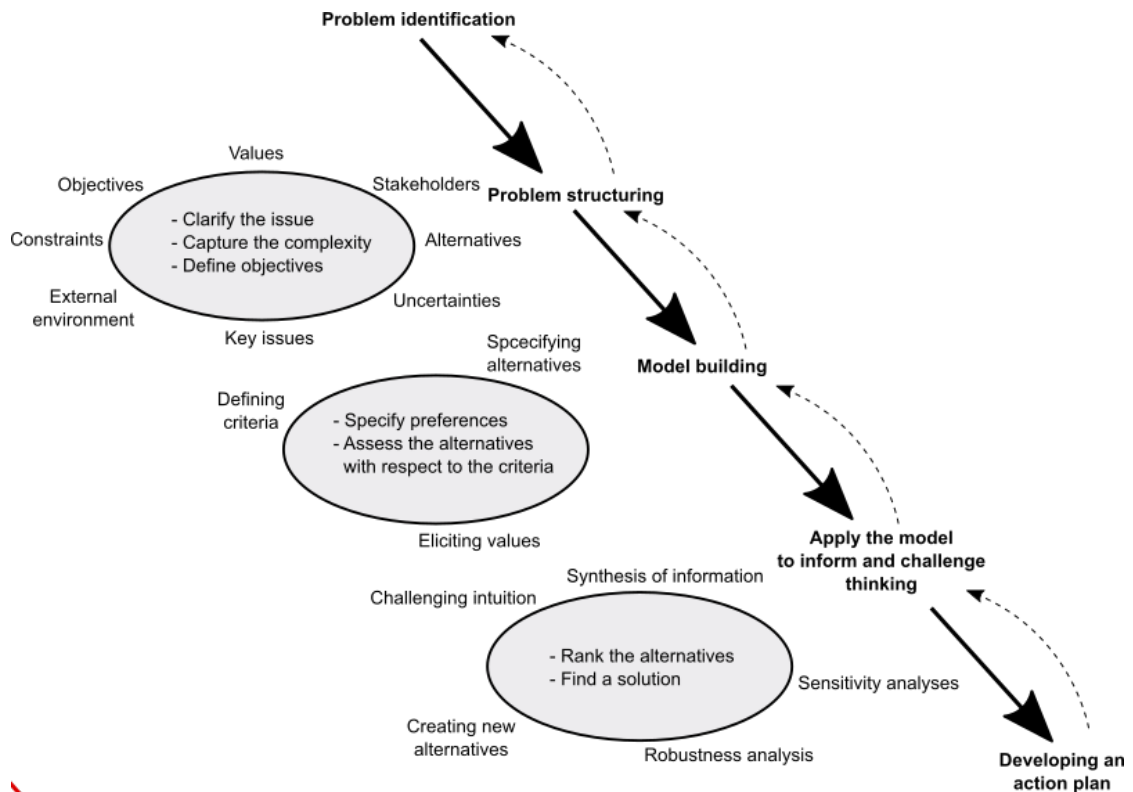
In this chapter, we introduce the main concepts and terms used throughout the thesis. First, we give an introduction to the main concepts of multi-criteria decision making, which general concept we transfer to robotic exploration in this thesis. Second, we state the general problem of robotic exploration and the general concept of exploration underlying our four use cases. Third, we describe the mapping on which our work is based on. The concept of multi-criteria decision making, robotic exploration and the described online 3D Mapping System are applied for each use case. Subsequently, we state the related work and pin down our own work to the literature. Thereby we present related work with respect to each use case.

#### 2.1.1 Multi-Criteria Decision Making

Making decisions is an important and critical component for autonomous robots. In robotic exploration, a robot is repeatedly confronted with the question, *‘where to move next?’*. At each exploration step, a robot has to choose the best action from hundreds of possible exploration actions. In this section, we state the basic definition of MCDM and common terms. How we formulate robotic exploration as MCDM problem, is detailed in chapter 3.

Making decisions is part of our daily lives. Facing a decision problem, one has to choose from a set of alternatives, the best solution. We are often confronted with complex decisions, with numerous alternatives and consequences difficult to grasp. Such complex decisions are often MCDM problems. According to Hwang et al. [24] a MCDM problem is present if, at least two conflicting criteria with incomparable units are present, multiple objectives exist and the goal is to decide for the best alternative or an optimal alternative. In fig. 2.1 we illustrate the main stages of MCDM after Belton and Steward [25]. After first, identifying the problem second, the problem is structured, to capture the complexity and to clarify the issue. Further, the objectives are defined in the problem structuring stage. Third, the criteria, preferences, and alternatives are identified and defined in the model building stage. Fourth, the model is applied, which leads to a ranking of the alternatives and a solution. For critical decisions, often a sensitivity and robustness analysis is applied to be sure the correct decision was made. To check the sensitivity of a solution, the model is updated several times and applied with varying preferences and criteria. The MCDM as described by Belton and Steward [25] is designed to help humans or a group of humans to solve a single decision problem. In

## 2 Exploration of Planetary Surfaces



**Figure 2.1:** Main stages of a MCDM adapted from Belton and Steward [25]. After identifying the problem on hand, a decision maker structures the problem by most importantly defining the objectives and alternatives. Only then, the model for decision making is build. Next, the model is applied to find a solution to the problem. The results can be used for example to refine the model.

chapter 3 we transfer MCDM to robotic exploration and explain how human operators benefit from the usage of MCDM.

For clarification, we state the most common terms of MCDM:

**Objectives:** describe the final state, which is desired. The main objective in turn can be expressed by several sub objectives. All objectives should be measurable.

**Alternatives:** are the different solutions to the decision problem. They are the potential subjects or actions the decision maker could choose. Each alternative excludes all other alternatives, such that the decision maker has to decide on one of the alternatives.

**Criteria:** are used to evaluate the alternatives, to find the alternative that best fits to approach the defined objectives.

**Preferences:** express the preference for one alternative over another to fulfill the objectives. An alternative is strictly preferred over another alternative if it dominates

the other alternatives for all objectives. Indifference expresses if choosing one alternative or another is irrelevant.

Many MCDM methods existing in literature were collected by Hwang et al. [24] in a survey. They can be classified into the two categories, Multi-Objective Decision Making (MODM) and Multi-Attribute Decision Making (MADM) [25]. For a MODM problem, the alternatives are not predetermined, the goal is to find the optimal solution to a decision problem. By contrast, with MADM one alternative from a discrete number of known alternatives is selected. Thereby, the solution is often a compromise, which best fits the preferences of the decision maker. In this thesis, we consider the decision ‘*where to move next?*’ of robotic exploration as MADM problem, as the goal locations represent a discrete number of alternatives. MADM methods have the advantage that the involved criteria can have different scales and measurement units, e.g. meter, seconds, energy, or even grades. Only, in the second step, the criteria are expressed in preference values which can be compared. MADM is further classified into value function approaches and outranking approaches.

The value function approaches calculate for each alternative an overall utility, by aggregating the criteria values. The alternative with the highest utility is the best alternative. Popular value function approaches are for example the Multi-Attribute-Utility/Value Theory (MAUT/MAVT) [26] and the Analytic Hierarchy/Network Process (AHP/ANP) [27]. The advantage of these methods is that they are very easy to understand and to model. However, setting up correct utility functions for each single criterion requires large datasets and a lot of time. Further, the preferences have to be very clear, which is often not the case. Outranking methods are based on a pairwise comparison of alternatives and have no underlying aggregate value function. It is possible to model weak preference and indifference, as well as to include redundant criteria information. Widely used outranking methods are for example ELimination Et Choice Translating REality (ÉLECTRE) [28], Preference ranking organization method for enrichment evaluation (PROMETHEE) [7] and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [24]. The preference for outranking approaches doesn’t have to be distinct, which allows for conflicting criteria. The result of an outranking method is a ranking of alternatives, which can be used to analyze the decision and the effect of the preferences on the solution.

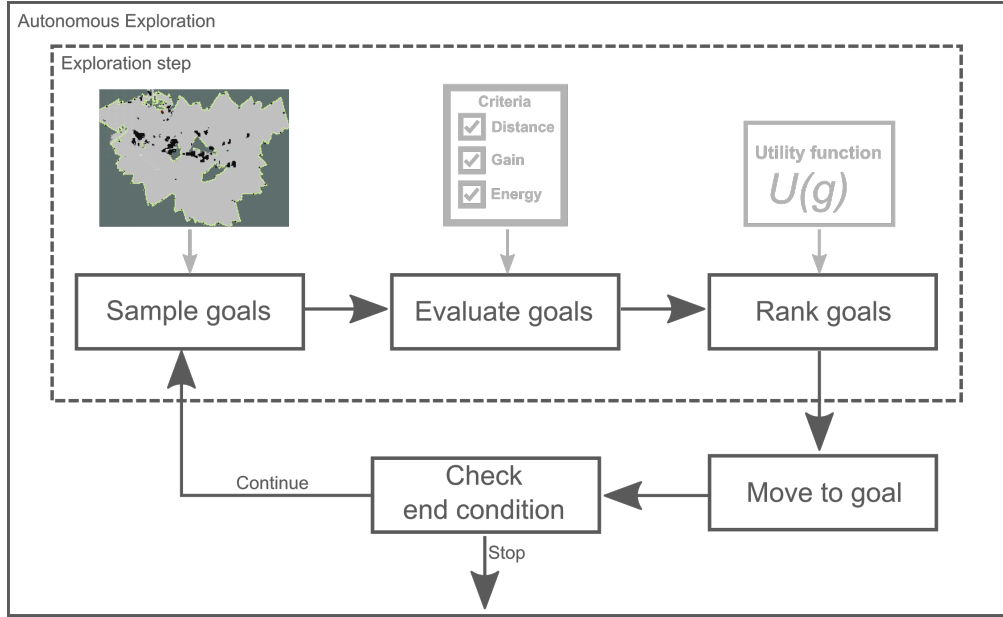
In this thesis, we apply and extend the outranking method PROMETHEE II [7] for decision making in robotic exploration. We detail the method, our extension, and application for exploration in section 3.3.2.

### 2.1.2 Autonomous Exploration

In their seminal work, Yamauchi et al. [4] describe the robotic exploration problem with: ‘*Given what you know about the world, where should you move to gain as much new information as possible?*’

An autonomous exploration rover has to answer this question repeatedly at each exploration action until the exploration mission is finished. In fig. 2.2 we illustrate the au-

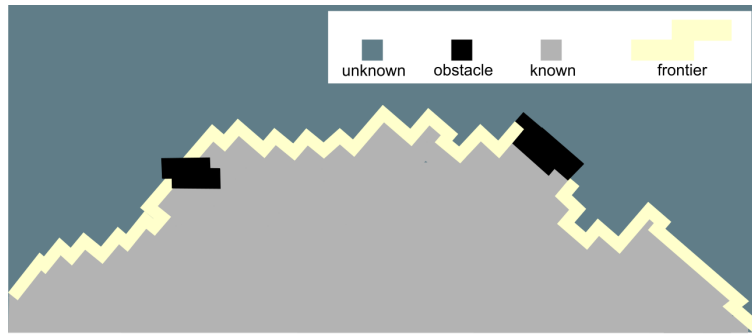
Autonomous exploration process and the single processes of one exploration action. First, potential goal locations, to which the robot could move, are sampled based on the current knowledge about the world, i.e. the current map. Throughout this thesis, these locations depict 3D locations in the global map frame and are named goals or goal locations. Second, the goals are evaluated with exploration criteria. Which criteria are applied depends on the mission task and the desired exploration behavior. Third, the goals are ranked by calculating the utility of reaching a goal based on the evaluated criteria. Finally, the robot moves to the goal with the highest utility value. The exploration mission ends if the defined end condition is reached. This could be a certain time budget or the full exploration of a ROI.



**Figure 2.2:** Overview of robotic exploration: In one exploration step, first, goal locations are sampled on the basis of the current map of the environment. Second, the goal locations are evaluated with exploration criteria. Third, the goal locations are ranked according to their utility. The robot moves to the goal with the highest utility and continues the exploration until an end condition is reached.

### Goal Sampling

Any 3D location in the area to be explored is a potential goal location. We express exploration goals as vectors  $\mathbf{g} \in \mathbf{SE}(3)$  of potential robot poses. However, usually a set of goals is sampled as it is not possible to evaluate and compare all possible goal locations. This, is even more important with the limited processing power of space rover hardware. We apply in this thesis the frontier strategy introduced by Yamauchi et al. [4], which is widely used. Goal locations are computed at the frontiers between known and



**Figure 2.3:** Illustration of a frontier (yellow line) between unknown (dark grey) and known (light grey) space in a 2d occupancy grid map with obstacles (black).

unknown space as illustrated in figure 2.3. In section 4.1 we describe the goal sampling method, we apply in this thesis, in detail.

### Goal Evaluation

In the second step, the set of goals is evaluated by computing exploration criteria. In this thesis, criteria are functions  $c : \mathbf{SE}(3) \rightarrow \mathbb{R}$ , that map an exploration goal to a scalar value, where higher values denote a higher preference for the particular goal regarding the respective criterion. Which criteria are applied depends on the mission task and the desired exploration behavior. Throughout this thesis, we describe eight exploration criteria, we apply to model the exploration behavior required to conduct the four use cases.

### Goal Ranking

In the third processing step of one exploration step, the decision on, ‘*where to move next?*’ is made. To rank the goals, it is common to use a utility function to correlate the single criterion values for each goal. In the following, we describe two widely used utility functions, which we used for comparison to analyse our own the utility functions and the MCDM method PROMETHEE II, which we investigate in this thesis. We reimplemented the utility function of Stachniss et al. [1] and the utility function of Gonzalez et al. [2]. Both functions use the distance from the robot to a goal as a cost criterion and apply the Information Gain (IG) to judge how much new information can be retrieved at a goal. Stachniss et al. [19] propose to use the difference between the information gain  $c_{IG}(\mathbf{g})$  and the weighted distance  $c_{cost}(\mathbf{g})$  to a goal in order to calculate the overall utility  $u(\mathbf{g})$  of moving to a goal. The goal with the highest utility value is chosen to be the best and thus the next goal the robot moves to.

$$u_1(\mathbf{g}) = c_{IG}(\mathbf{g}) - w_{cost} \cdot c_{cost}(\mathbf{g}) \quad (2.1)$$

Gonzalez et al. [2] propose to multiply the information gain  $c_{IG}$  with the exponential function of the distance  $c_{cost}$  to calculate the utility  $u_2(\mathbf{g})$  of reaching a goal.

$$u_2(\mathbf{g}) = c_{IG}(\mathbf{g}) \cdot e^{(-w_{cost} \cdot c_{cost}(\mathbf{g}))} \quad (2.2)$$

Both utility functions, use a constant to weight the cost of the motion against the expected information gain, thus a similar exploration behavior is achieved with both function. As the function of Stachniss et al. [19] is more easy to parametrize, we use the presented utility function  $u_1(\mathbf{g})$ , as baseline and compare it with our integrated and informed exploration behavior in chapter 6.

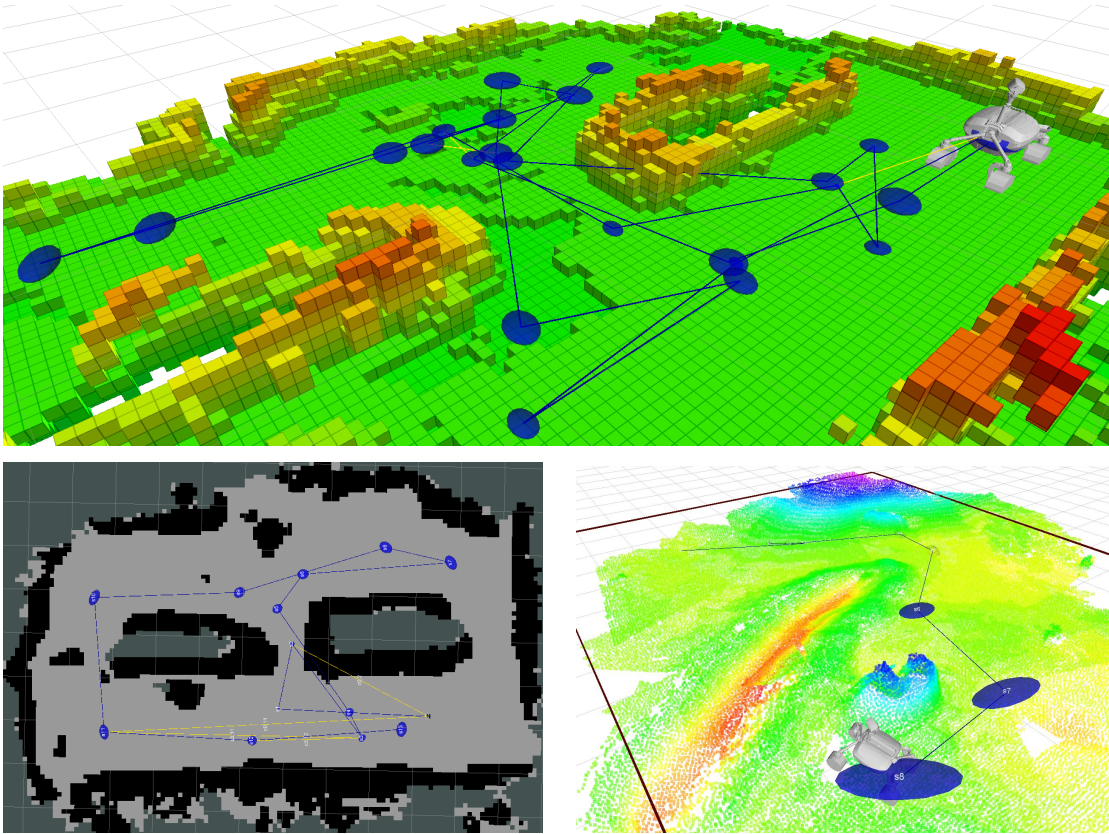
### 2.1.3 Online 3D Mapping

At each exploration action, the next exploration action is computed from the currently known information about the environment, usually stored in a map. An exploration approach and especially its implementation depends to some extent on the underlying mapping system and the available map representations. In this section we describe the 6D global localization and mapping system of Schuster et al. [8, 20] which we apply for Simultaneous Localization and Mapping (SLAM). It is based on the creation of local submaps and employs submap matches as loop closure constraints for global graph optimization. For details, please refer to the publications of Schuster et al. and Brand et al. [8, 20, 17].

#### Submap Generation

We perform dense stereo matching with the Semi Global Matching (SGM) [21] along the robot trajectories, estimated with a key-frame based Extended Kalman Filter (EKF) as local reference filter [22], and integrate the depth data in a submap. A submap is defined by its origin, which matches a local reference frame withing the local reference filter, and by its associated 3D data structure. The 3D data is stored as a colored 3D point cloud and 3D probabilistic voxel map. The 3D point cloud additionally stores a binary obstacle classification that is also used for obstacle avoidance. For the 3D probabilistic voxel map the open-source *OctoMap* library [23] is employed. It holds the information about occupied, free and unknown space, which we use to compute the expected information gain for our exploration criterion  $c_{IG}$ , detailed in section 4.2.

A new submap is created on two events: first, if the accumulated driven distance since the last submap was created reaches a defined maximum value, or second, each time the positional uncertainty of the robot grows above a defined threshold. In fig. 2.4 we visualize the global map, which is generated by merging the 3D data structures of all submaps. By projecting the obstacle and known points to a grid in the plane, a 2D occupancy grid-map representation, of the global map is created. Each map cell can have the state, free, unknown, or occupied. Free and occupied cells are already observed by the robot's sensor and thus known. A free cell is traversable, whereas an occupied cell contains an insurmountable obstacle for the robot. We apply the 2D occupancy grid-map for fast frontier computation in our thesis. For most planetary exploration



**Figure 2.4:** Impressions of the underlying mapping system. Top: Height-colored probabilistic voxel-grid map. Bottom left: 2D occupancy grid-map. Bottom right: 3D height colored point cloud. In each map, the underlying SLAM graph is depicted. The blue ellipsoids represent the positional uncertainty at the submap origin. Yellow edges represent loop closures between submaps.

scenarios, it is sufficient to use this 2D abstraction. Only overhanging rocks or caves can not be represented correctly however, we believe the exploration of caves should be regarded separately from general surface exploration. However, to calculate the IG we utilize the 3D probabilistic voxel-grid map, we show in fig. 2.4 top row.

### Submap Matching

In fig. 2.4 we show in each map the corresponding SLAM graph. The submap origins represent the nodes in the underlying sparse, undirected graph. The blue edges connecting these nodes originate from constraints between consecutive submaps, and the yellow edges result from loop closure constraints. These loop closure constraints are generated by pairwise matching between submaps and computation of the relative transformation between the submap origins together with uncertainty estimation. As the matching is based on the 3D geometry structure, more precisely the obstacle points, of the submaps, first submaps which are not sufficiently informative for matching are filtered out. A

submap has to contain a minimum number of 3D points and well distributed obstacle points. To find pairs of submaps for matching, the maps are sorted by their origins and overlapping 2d bounding boxes in the  $xy$ -plane. In addition, the overlap is used to rank the pair of submaps, as it is an indicator for the probability of a successful match. To match a pair of submaps, first, an initial alignment is computed by 3D feature matching. As proposed by Schuster et al. [20] we use the obstacle points as key points. Based on the current SLAM estimates, the most reasonable model is selected and refined with an Iterative Closest Point (ICP) algorithm. Finally, an uncertainty estimate is computed for the resulting 6D relative transformation between the two matching submaps and sent to the graph SLAM. The computed transformation is used as an additional constraint for global optimization.

### Multi-robot Mapping

In our fourth use case, we describe a multi-robot exploration. For this, we apply the decentralized multi-robot setup introduced by Schuster [8]. The above described localization and online 3D mapping are executed on board of all robots in a decentralized or distributed way. This ensures, at each time, an up-to-date pose estimate and map on all robots within the multi-robot team. A connection between the robots is made in the SLAM graph, by submap matches, robot detections, or landmark detections, which are also shared between the robots. Besides, the robot pose estimates, the 3D submaps are exchanged, and the latter is integrated into the own maps. In this thesis, additional exploration goals are exchanged between the robots in order to coordinate a team of robots during exploration. Our multi-robot coordination is described in detail in section 4.5. For more details on the multi-robot setup itself, please refer to Schuster [8].

## 2.2 Related Work

In this section, we present the related literature to our work on robotic planetary exploration and work out the existing gaps in the literature on the challenges of robotic exploration of extra-terrestrial surfaces.

Most authors [2, 29, 30, 1, 31] treat one single robotic exploration task. Usually, the mission's task is defined subconsciously and a utility function, specially designed to evaluate the goal locations, for the mission task, is set up. With such an approach, it is not possible to conduct different mission tasks and an operator has only very limited options to influence the implemented behavior.

Especially, for planetary exploration, we believe it is essential to contemplate robotic exploration as MCDM problem. Basilio and Amigoni [32] first introduced MCDM for robotic exploration. They applied it for Search and Rescue (SaR) scenarios, which have similar challenges as a planetary exploration scenario. To compute global utilities for varying exploration behavior, they apply the Choquet fuzzy integral. Similar to our approach, Taillandier and Stinckwich [33] utilize PROMETHEE II. The advantage of



PROMETHEE II, is that the difference of the criterion values is used to estimate the global utility, instead of the criterion values themselves. It allows estimating criteria with non-linear importance. Further, the difference in criterion values is transferred to a preference degree, which allows building a global utility from criteria with different units and characteristics. The application of MCDM methods for decision making, especially outranking methods, has the disadvantage of a long processing time, which is not copped by the work of Basilico and Amigoni [32] and Taillandier and Stinckwich [33]. Both, Basilico and Amigoni [32] and Taillandier and Stinckwich [33] apply MCDM methods only for the goal ranking step of exploration, however, they do not apply MCDM as a procedure to structure and formulate an exploration problem. We think this is important to really benefit from MCDM and to model different exploration behaviors to solve various missions tasks.

### 2.2.1 Autonomous Exploration

In use case 1, a robot has to explore an ROI. This depicts the basic exploration problem, which has the objective to create a complete map of a prior unknown region. A large number of exploration approaches exist, which can be classified into classical strategies and integrated strategies. Classical exploration strategies [4, 2, 1, 29, 30, 34] try to solve solely the area coverage problem, i.e. exploring the environment as fast as possible. Integrated exploration strategies [35, 36, 37, 38, 39, 40, 41, 42, 19], aim for an efficient exploration and simultaneously for a good map quality and an accurate localization. Considering the map quality and localization uncertainty is especially important for exploring large regions and missions, where a high level of safety is required, as a mission failure leads to a huge amount of costs.

The localization uncertainty and map quality can be improved, by decreasing the uncertainty accumulated by the robot movement [37, 36, 38] or by reducing the uncertainty [41, 31, 36, 35, 39], e.g. by actively seeking for loop closure events for re-localization. To evaluate the potential of generating a loop closure, when visiting a goal location, most authors only evaluate the likelihood of such an event [1, 41, 36]. Only Carrillo et al. [35] and we [10] consider additionally to the likelihood of a loop closure, the impact of a loop closure. Carrillo et al. [35] introduce a utility function based on the difference between the Shannon and the Re'nyi entropy of the current distribution of over maps. Although Carrillo et al. [35] achieve a high map quality, their approach has only a low exploration efficiency and the computation of the Renyi entropy leads to high computational costs.

Most strategies are designed for and tested in indoor environments [36, 35, 37, 38, 19, 41]. There exist only a few publications on exploration of outdoor environments [30, 29] and to the authors knowledge none was evaluated and tested in a large outdoor environment with a real robot. Almost all, approaches are only tested with simple 2D simulations [36, 38, 39, 29, 41].

## 2 Exploration of Planetary Surfaces

The survey of Garaffa et al. [43] investigates the current literature on learning methods for robotic exploration. They especially survey reinforcement learning, by which robots automatically learn skills from environment interaction. Most approaches for single robot exploration are end-to-end methods, e.g. [44, 45, 46, 47, 48], which act as a black box, as they directly return robot control actions [43]. The few existing two stage approaches [49, 50, 51] first, decide on the next goal location applying reinforcement learning and second, compute a path to the goal with common path planning techniques. This still means the decision process on *where to move next?* is a complete black box for a human operator. For space exploration missions, scientists and human operators are adamant that a persistent possibility of supervising a robot exists. Further, space exploration differs strongly from the domains where reinforcement learning is already successfully applied. An agent has to act from the beginning autonomously and a safe exploration is required, there is no time for learning in the real world. All methods reviewed by Garaffa et al. [43], perform the training phase through simulation. As the existing data of planetary surfaces is very limited and no exact models of the environment to be expected exists, this is especially for planetary exploration missions currently an invincible challenge. Garaffa et al. [43] come to the conclusion, that the small performance improvements of reinforcement learning methods for single robot exploration, do not justify their application over classical exploration methods, due to their high computational costs and long training periods. We share this opinion, especially for planetary exploration, considering the lack of training data and the requirement to understand the decision a robot made and to supervise a robot by human operators.

### 2.2.2 Drive-by Science

Our second use case represents a drive-by science mission, often also called science on the fly or opportunistic science. The main goal is to move to a defined POI, at which new science investigations are planned. The idea is to use the time during the traverse to increase the scientific information gained while moving towards the POI [52]. Some authors, even propose to interrupt the traverse for science investigation, as an opportunity arises. In the small space exploration community, only a few approaches to drive-by science were published. However, among these, no real exploration strategy can be found, but only informative path planning methods. It can be distinguished between methods planning a path based on prior knowledge [53, 54], e.g. from satellite data, and methods adapting a path continuously [55, 52, 56], while traversing the environment based on the newly gained information. The disadvantage of considering drive-by science as path planning problem, is that it requires prior knowledge and the re-planning often requires complex computations. We suggest to use an exploration approach, to set up a mission in a drive-by science fashion.

### 2.2.3 Object Search

In use case 3, the robot has to find a FOI in an unknown ROI, which is similar to the robotic task of object search. The object search problem is a common robotic task,

service robots have to search and fetch household items, search and rescue robots have to identify hazards or locate victims and industrial robots have to fetch, for example, tools or materials. Recently, the object search problem got more attention in the robotic community, whereby the context of the object search differs and thus the methods. A bunch of work exists where the environment is known in advance [57, 58, 59, 60, 61, 62, 63, 64, 65, 63, 66, 67, 68, 69], only a few authors consider the search task in a-priori unknown environments [70, 71, 72, 73, 74, 32, 75, 76].

As in our use case, a robot has to find a defined FOI in a prior unknown environment, we examine the literature on object search in unknown environments in the following in detail. We distinguish between direct search methods and informed search methods. Direct search [77, 73, 78, 32, 76] assumes that the target object lies with equal probability at any location and selects viewpoints based on the environment geometry until the object is found, thus it is similar to the area coverage problem, respectively the classical exploration problem [77].

Informed search methods [72, 70, 71, 76, 74, 75, 79] incorporate search knowledge to reason about an object’s location. Rasouli et al. [79, 75] utilize visual saliency to guide the search. They build a saliency map during the search and update the probability of detecting the object at a location based on this map [79, 75]. To build the saliency map they explore visual clues in the current images. Shubina and Tsotsos [76] and Cipolleschi et al. [74] integrate hints about the target object location as a probability distribution. Additionally, Ciopolleschi et al. [74] use these hints to prioritize areas and to coordinate multiple robots. Areas with a high probability of detecting the object are explored by several robots, whereas areas with less probability of detecting the object are neglected or only explored by one robot. Aydemir et al. [72] and Joho et al. [70, 71] present approaches that make use of indirect search knowledge. The approach of Joho et al. [70] is based on spatial relations, gained from usual object arrangements, e.g. milk is in a fridge, and object attributes, e.g. avocado is a fruit, to find a product in an a-priori unknown supermarket. They use a maximum entropy model, which models the conditional distribution over possible locations of the target object given all observations made so far. The parameters of the model are learned by maximizing the data likelihood using a gradient ascent. In a follow up work Joho et al. [71] compare their strategy with a reactive search strategy, which uses only local information to decide where to search next, instead of all observations made so far. They learn a decision tree from optimal search paths, which include positive and negative decisions. The decisions are again derived from object co-occurrences. One drawback of Joho et al.’s work is that they rely on RFID tags to detect a target object. Thus, the target object has not to be in the field of view of the visual sensor but only has to be in a certain range around the robot. This simplifies the search and makes the approach impractical for applications other than supermarkets. Aydemir et al. [72] present an active visual search method, which utilizes uncertain semantics of the environment and a probabilistic model of the search environment to prioritize the search. The robot is equipped with general world knowledge about indoor spaces to be able to exploit spatial semantic relations. In detail,

they use place-object relations, e.g. a mug is likely to be found in a kitchen. Their robot is able to reason about the category of a room and to connect these to the target object. They propagate the probability of detecting an object in a certain room to the 3D voxel space of the room, which is then used for detailed view planning. The described informed search method, are all designed for use cases in human structured indoor environments, and can't be applied for finding features of interest in a planetary exploration scenario. In a space exploration scenario, a FOI is usually not detectable by a visual saliency algorithm, as it does not stand out in the camera image. Also, spatial relations can not be defined for features on planetary surfaces. Only hints about the location of a FOI could be given by a human scientist, in order to prioritize regions. Nevertheless, we believe it is important to incorporate the probability of finding a FOI in the decision *'where to move next?'* to increase the mission return.

### 2.2.4 Multi-robot Exploration

In use case four, two robots collaborate to efficiently explore a ROI. Using a team of robots increases the performance and robustness significantly. Several coordination strategies for a team of robots were proposed over the last years. A common state of the art approach is to assign frontiers to the robots [4, 80, 81, 82, 83]. In their seminal work Yamauchi et al. [4] sequentially compute a cost-utility function for each robot and frontier to assign the frontiers. Burgard et al. [81] propose to use the Hungarian Method to improve the assignment of the frontiers. Bautin et al. [82] propose to first rank the goal locations, by determining how many robots are closer to a goal than the currently considered robot.

Another, common approach for coordination, is to split the environment into small parts, which are assigned to the robots [84, 85, 86]. For instance, Kemna et al. [85] and Nieto-Granda et al. [86] use a dynamic Voronoi partitioning method and Solanas and Garcia [84] the K-means clustering algorithm to divide the environment.

Reactive behavior based coordination approaches were proposed by Julia et al. [87], Lau [88], Arkin and Diaz [89] and Rooker and Birk [90]. Different behaviors are combined to compute the desired heading vector for a robot. For example, Lau [88] apply the three behavior, avoid obstacles, go to frontier and avoid other robots. The coordination of the robots is implemented by the behavior avoid other robots. Julia et al. [91] additionally apply the behaviors, improve imprecise landmarks and go to unexplored zones. To guarantee consistent communication between a team of robots, Arkin and Diaz [89], as well as Rooker and Birk [90] implement a behavior, that limits the distance, that robots, in a team, can separate from each other. Rooker and Birk [90] additionally suggest to change the behavior, from exploring to meet with each other, to avoid deadlocks. These deadlocks, happen when a robot can not move towards a frontier without violating the distance constraint to other robots. The behavior based coordination strategies only implement weak coordination, which depends on the structure of the environment [87] and performs only well in simple environments with few obstacles.

In our scenario (see section 1.2), we consider a team of two heterogeneous robots. Enhanced tasks, as autonomous science investigations, require a local collaboration between the robots. For example, in our case, one rover is equipped with a scientific camera to detect a FOI and the other has a manipulator to take samples. Our setup is similar to the one of Manjanna et al. [92], who measure the chlorophyll distribution of water, using an explorer detecting suitable locations and a sampler measuring the chlorophyll. Andre and Bettstetter [93] investigate local collaborations for physical interactions, e.g. clearing a block path with combined strength. They, especially, treat the question, when to collaborate and with whom to collaborate [93]. Although, we do not explicitly consider local collaboration in this thesis, but solely the co-exploration of a ROI, we keep our heterogeneous team in mind and present a coordination approach, where the robots keep close to each other, to quickly react on a request for support by another robot.



## 3 Exploration as Multi-Criteria Decision Making Problem

In this chapter, we present our general exploration concept based on MCDM. We show the MCDM problem within the robotic exploration task and, explain how an exploration behavior can be modelled by a human operator, especially for planetary exploration missions. Further, we describe two MCDM methods to rank goal locations.

### 3.1 Identification of the MCDM Problem in the Exploration Process

In robotic exploration, a robot has to repeatedly answer the central question, ‘*where to move next?*’. We show that this decision problem is a MCDM problem, by expressing the main characteristics of MCDM (see section 2.1.1) in the sense of exploration:

**(1) At least two criteria** - To successfully conduct the challenging task of exploring large unknown environments, a robot has to consider its next action carefully. An efficient exploration requires at least two criteria. It is state of the art, to consider the distance from the robot to the goal and the information that is gained when moving to the goal. As the localization error accumulates over time with no Global Positioning System (GPS) system available, it is further common to consider if a robot could re-localize itself at a goal. In this thesis, we even apply five criteria at the same time to model an exploration behavior to conduct use case 3.

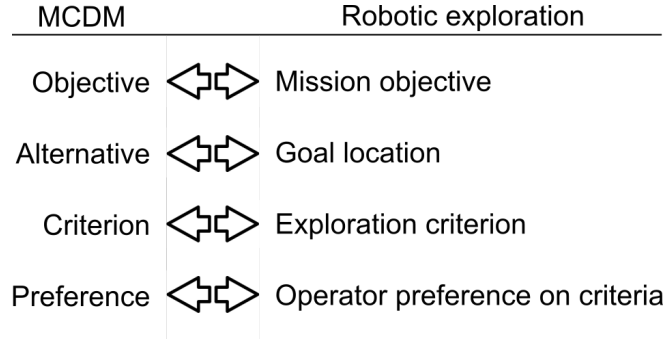
**(2) Multiple objectives and conflicting criteria** - The exploration tasks induced by a planetary exploration mission, all include multiple objectives. For example, in use case 1, we aim for an efficient exploration, an accurate localization and a good map quality. As the robot usually has to interrupt the exploration process to re-localize itself, the exploration efficiency, localization accuracy, and map quality represent conflicting objectives.

**(3) Incomparable Units** - The attributes formulated to describe the criteria, e.g. the path length to a goal, the expected information gain, or the re-localization probability, all have different units and can’t be compared directly.

**(4) Design/Selection** - The main objective of robotic exploration is to decide ‘*where to move next?*’ Thus, the robot has to select a next action to best fulfill the objectives of an exploration task.

### 3 Exploration as Multi-Criteria Decision Making Problem

To structure a decision problem with a MCDM method, the problem is commonly described by objectives, alternatives, criteria, and preferences. In fig. 3.1 we highlight the transfer of common terms from MCDM to robotic exploration.



**Figure 3.1:** We transfer the common terms of a MCDM to the terms of robotic exploration. The mission objectives, exploration criteria and operator preferences directly map to the objectives, criteria and preferences of a MCDM. The alternative solution of a MCDM correspond to the sampled goal locations.

The main objective describes the final state to be reached. It can be expressed by several sub-objectives, which should be measurable. For planetary exploration the main objective arises from the mission task itself, e.g. for use case 1, the main objective is to explore a ROI. Depending on the main objective, we can derive different sub-objectives, e.g. mapping a large area and reducing the localization uncertainty. The exploration criteria used to evaluate the potential goal locations can be directly transferred to MCDM. To reach the objectives, the goal locations are evaluated by these criteria. The preferences depict the preference of one alternative over another regarding the criteria, i.e. fulfilling the respective objective. In exploration the human operator can set the preference to influence the decision making. It influences which goal location, an autonomous robot prefers over another.

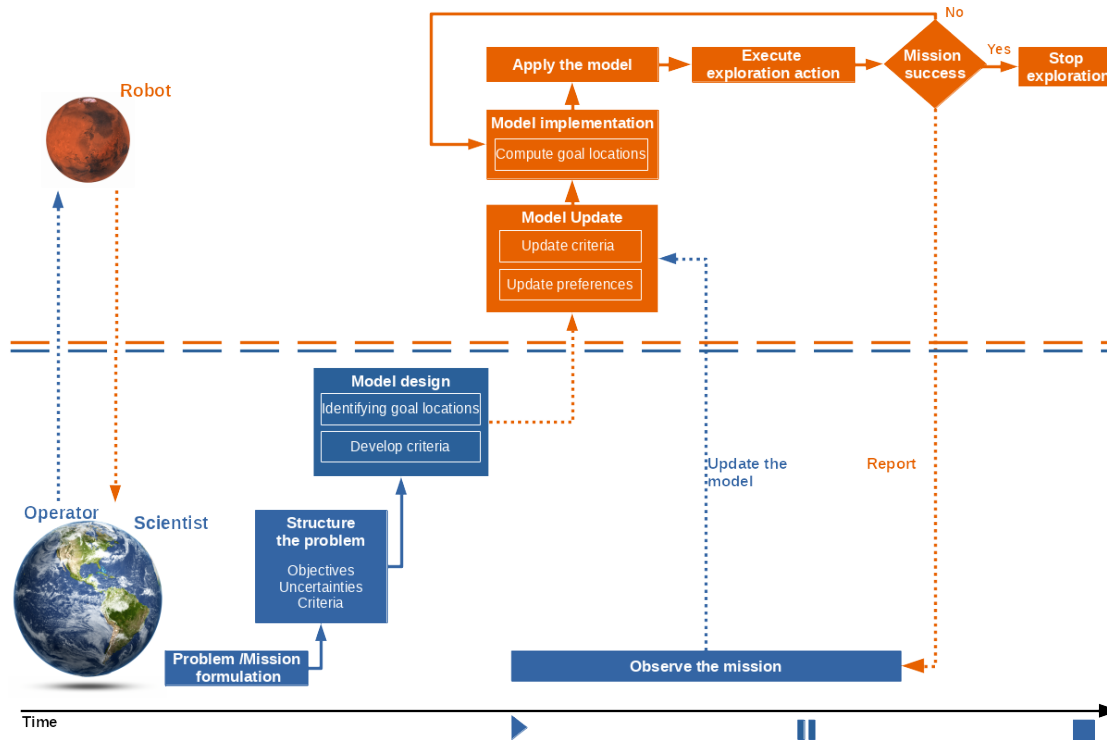
## 3.2 General Formulation of Robotic Exploration as MCDM

In fig. 3.2 we illustrate how we transfer the stages of MCDM, stated in section 2.1.1, to robotic exploration of planetary surfaces. The blue processes are conducted by the operator and scientists on Earth before the mission starts. The orange processes run online onboard a robot on e.g. Mars. The communication between the operator on earth and the robot is limited to short communication time slots, which is highlighted by the dotted errors.

**Problem formulation and structuring:** Starting from the high level mission task, the problem is structured by the scientist and the operator by identifying the challenges, requirements, and mission sub-objectives. Further, potential uncertainties are recognized, and the decision space is defined. To structure and model a mission



### 3.2 General Formulation of Robotic Exploration as MCDM



**Figure 3.2:** MCDM procedure for an autonomous exploration mission. Blue processes are conducted by the operator and scientists on Earth. Orange processes run online on-board a robot on e.g Mars. Solid connections between the processes represent a short communication time and dotted connections a delayed communication in the short available communication time slots. The x-axis depicts the time and mission progress. Before the exploration task is executed the operator structures the problem and design the model. During task execution the model is repeatedly implemented on the robot, applied and an exploration action is executed, until mission success. After each exploration action the robot reports the progress and the operator can update the model if required.

task we suggest first building a hierarchical goal system, as illustrated in figure 3.3. It shows the coherence between the main objective, sub-objectives, and exploration criteria. The goal system, should further state if a criterion should be maximized or minimized and its unit.

**Model design and model implementation:** In MCDM the second stage is the model building stage, which includes the identification of alternatives and their computation. This is not applicable for robotic exploration, as not one single decision, but repeatedly, for each consecutive exploration action a decision has to be made. Therefore, we split the model building stage into two stages, model design on Earth and model implementation on the robot. In the model design stage, we define the sampling method for goal locations, develop the criteria to evaluate them, and

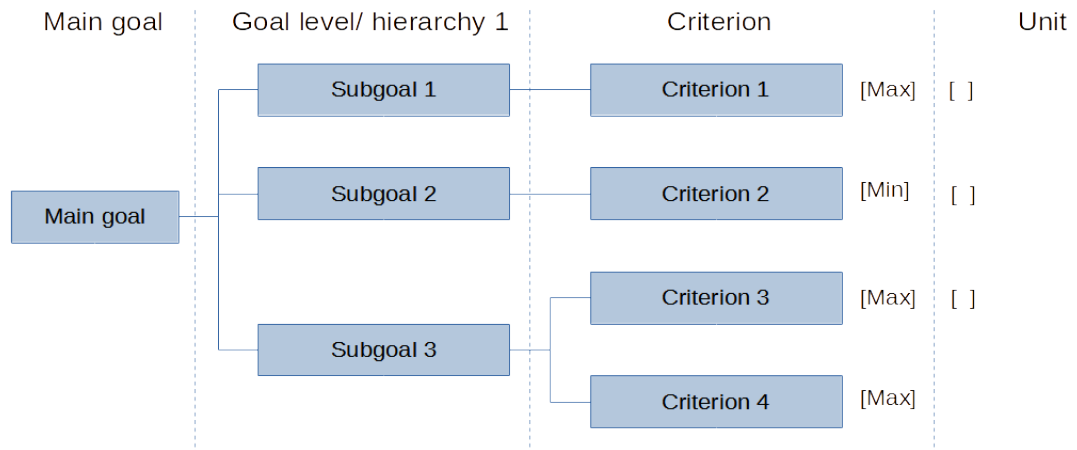
### 3 Exploration as Multi-Criteria Decision Making Problem

postulate their preferences. In the model implementation stage, the goal locations are computed on the basis of the current map of the robot. As the preferences, criteria, and also the goal locations highly depend on the mission objectives, the model has to be designed and implemented for each use case.

**Applying the model:** Finally, the model is applied, i.e. the criteria are evaluated for each goal location and a ranking is built with a MCDM method. The solution to the decision problem is the goal with the highest rank.

**Execution of an exploration action:** Now, the robot approaches the chosen goal location, by planning a path through the environment and following it until the goal location is reached.

**Reporting:** After execution of the chosen exploration action, the mission progress is checked. We sent a report on the last exploration action and the mission progress to the operator on earth. The operator can analyze the past exploration actions and update the model if required. The robot updates the criteria and preferences in the model design stage, immediately after receiving it and applies the updated model for the decision on the next exploration action.



**Figure 3.3:** General illustration of a goal system to structure an exploration problem. The main goal is subdivided in to several subgoals. Each subgoal is described by measurable criteria.

### 3.3 MCDM Methods for Robotic Exploration

In this section, we introduce two MCDM methods, we suggest for ranking the goal locations, which is part of *applying the model* in the MCDM procedure. First, we introduce in section 3.3.1 two Multi-Attribute Decision Making (MADM) utility functions

especially designed for use case 1, which implement an integrated exploration behavior. Second, we describe in section 3.3.2 the outranking approach Preference ranking organization method for enrichment evaluation II (PROMETHEE II) and our extension, to save computation time. PROMETHEE II can be applied in general for any of the use cases or even any exploration problem to rank goal locations.

### 3.3.1 Multi-Attribute Utility Functions for an Integrated Exploration

In this section, we describe two novel utility functions, which we first published in Lehner et al. [10]. The utility functions implement an integrated exploration behavior, which balances the exploration efficiency with the map quality and localization accuracy, as required for use case 1.

The Combined Active Loop Closing and Exploration (CALE) approach, tries to find the optimal goal, which maximizes the exploration efficiency as well as the re-localization performance simultaneously. All evaluation criteria are incorporated in a single utility function.

The Separated Active Loop Closing and Exploration (SALE) approach, first decides based on the current localization uncertainty of the robot, if the robot should explore or re-localize itself. Based on this decision, either goals to explore or to re-localize are sampled and evaluated, due to their exploration performance or their re-localization performance.

In this section, we describe the utility function for ranking goal location in general. Details on, where goal locations for exploration and re-localization are sampled are described in chapter 4. The same holds for the exploration criteria applied to evaluate the goals, a detailed description, can be found in section 4.2.

SALE allows only one type of action, exploration or re-localization, to be considered at a time. The robot first decides if it can explore unknown environment or if it requires to re-localize itself. The decision to explore or to re-localize is made based on the norm of the current covariance  $|cov(\mathbf{p}_r)|$  of the robot pose  $\mathbf{p}_r$ . If the covariance of the pose is larger as the threshold  $\iota$  the robot has to interrupt the exploration in favor of re-localizing itself. Depending on this decision, the set of goal locations either includes exploration goals or re-localization goals, and the utility for an efficient exploration eq. (3.2) or the utility for the re-localization performance eq. (3.3), is computed.

$$u(\mathbf{g}) = \begin{cases} u_{ex}(\mathbf{g}) & \text{if } \iota \geq |cov(\mathbf{p}_r)| \\ u_{re}(\mathbf{g}) & \text{if } \iota < |cov(\mathbf{p}_r)| \end{cases} \quad (3.1)$$

To find the goal, which has the greatest exploration performance, we apply the utility function introduced by Stachniss et al. [1] (see section 2.1.2, eq. (2.1)).

$$u_{ex}(\mathbf{g}) = c_{IG}(\mathbf{g}) - w_{cost} \cdot c_{cost}(\mathbf{g}) \quad (3.2)$$

It balances the information gain  $c_{IG}$  with the cost  $c_{cost}$  of moving to a goal. Details on the calculation of the information gain criterion (see section 4.2.2) and cost criterion

### 3 Exploration as Multi-Criteria Decision Making Problem

(see section 4.2.1) are given in the next chapter, where we explain how to model the exploration behavior for each single use case. To find the goal having the greatest re-localization performance, we consider two criteria, which evaluate the likelihood  $c_{ll}$  of a re-localization and its impact  $c_{li}$ .

$$u_{re}(g) = ((\beta \cdot c_{ll}(\mathbf{g})) + (1 - \beta) \cdot c_{li}(\mathbf{g})) - w_{cost} \cdot c_{cost}(\mathbf{g}) \quad (3.3)$$

The weight  $\beta$  balances the likelihood and the impact of re-localization. Similar to the utility function for an efficient exploration eq. (3.2) we subtract the weighted cost criterion  $c_{cost}$  to limit the distance the robot has to drive to a goal. For details on the calculation of the loop closure likelihood criterion and the loop closure impact criterion please refer to section 4.2.3 and 4.2.4.

We use a simple additive weighting approach to calculate the re-localization utility  $u_{re}(g)$ , based on a linear combination of  $c_{ll}$  and  $c_{li}$ , which is a well known method for ranking alternatives based on multiple attributes. Alternatively, it would be possible to multiply the two criteria to avoid the assumption that there is a fixed trade off between the two criteria, which can be expressed with linear preference.

CALE searches for the optimal goal which satisfies all four criteria. We define the utility of reaching a goal  $u(\mathbf{g})$  as:

$$u(\mathbf{g}) = (\alpha \cdot u'_{re}(\mathbf{g}) + (1 - \alpha) \cdot u'_{ex}(\mathbf{g})) - c_{cost}(\mathbf{g}) \quad (3.4)$$

with  $\alpha = |cov(\mathbf{p}_r)|/\iota$

where  $u'_{re}$  and  $u'_{ex}$  are similar to  $u_{re}$  and  $u_{ex}$  without considering the costs  $c_{cost}$ , as it is subtracted from the sum of  $u'_{re}$  and  $u'_{ex}$ , i.e.:

$$u'_{re}(\mathbf{g}) = (\beta \cdot c_{ll}(\mathbf{g})) + (1 - \beta) \cdot c_{li}(\mathbf{g}) \quad (3.5)$$

$$u'_{ex}(\mathbf{g}) = c_{IG}(\mathbf{g}) \quad (3.6)$$

Depending on the ratio of the norm of the currently estimated covariance of the robot pose  $|cov(\mathbf{p}_r)|$  and the threshold  $\iota$ , either exploration actions or loop closing actions, are weighted higher.

CALE distinguishes itself by searching for goals that are suitable for both, exploration and re-localization. This can speed up or slow the exploration process, depending on the parameters. It has the advantage that we do not need to define a fixed threshold, at which the robot is forced to interrupt the exploration process immediately.

The advantage of SALE is, that it is easier to understand the behavior of the robot, which simplifies the choice of the exploration parameters for human operators during mission planning. In addition, the exploration is only interrupted if it is really necessary. CALE usually starts earlier to revisit known areas, if the opportunity to drive to a close re-localization goal is given. With SALE, it might happen that the robot has to drive back long distances to reach a re-localization goal worth visiting. This is further evaluated and discussed in section 6.1.

### 3.3.2 Multi-Criteria Decision Making with Promethee II

The presented utility functions are especially designed for an integrated exploration behavior required to conduct use case 1. It is not possible to apply the same utility functions to find a solution for the other use cases. In this section, we describe the well-known MCDM method PROMETHEE II [7], which we suggested in our publication Lehner et al. [11] as a general decision making method for robotic exploration. The outranking method is based on pair wise comparison of goal locations and has no underlying value aggregation function. Any exploration behavior can be modeled with PROMETHEE II, only by adding or removing criteria and adapting the preferences. Further, we describe our extension to PROMETHEE II to save run time, which is important for planetary exploration applications with limited CPU resources.

#### Brief Introduction to PROMETHEE II

PROMETHEE II is an outranking method, based on pairwise comparison of alternatives, i.e. in our case the goal locations. Let  $G = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n\}$  be a set of goal locations with the goal locations being a vector  $\mathbf{g} \in \mathbf{SE}(3)$  of potential robot poses. The exploration criteria to rate and compare the goal locations in the set  $G$  are stored in the set of criteria  $C = \{c_1, c_2, \dots, c_m\}$ . Higher values of these real-valued functions  $c : \mathbf{SE}(3) \rightarrow \mathbb{R}$  denote a higher preference for the particular goal location regarding the respective criterion.

First, all goal locations  $\mathbf{g} \in G$  are compared pairwise for each criterion  $c_k \in C$ . Hence, let

$$d_k(\mathbf{g}_i, \mathbf{g}_j) = c_k(\mathbf{g}_i) - c_k(\mathbf{g}_j) \quad \forall k = 1, \dots, m \quad (3.7)$$

denote the difference between two goal locations  $\mathbf{g}_i$  and  $\mathbf{g}_j$  with respect to criterion  $c_k$ .

Second, the difference  $d_k(\mathbf{g}_i, \mathbf{g}_j)$  is mapped to the unicriterion preference degree applying a preference function  $P_k : \mathbb{R} \rightarrow [0, 1]$  defined for each criterion  $c_k$ .

$$\pi_k(\mathbf{g}_i, \mathbf{g}_j) = \begin{cases} 0 & \text{if } d_k(\mathbf{g}_i, \mathbf{g}_j) < 0 \\ P_k(d_k(\mathbf{g}_i, \mathbf{g}_j)) & \text{else} \end{cases} \quad (3.8)$$

The preference functions are modeled before the exploration starts in the model design stage (see fig. 3.2). We describe the most common preference functions in detail in the next section 3.3.2. Summing up the weighted unicriterion

$$\pi(\mathbf{g}_i, \mathbf{g}_j) = \sum_{k=1}^m \pi_k(\mathbf{g}_i, \mathbf{g}_j) \cdot w_k \quad (3.9)$$

with the weights  $w_k \in [0, 1]$  and  $\sum_{k=1}^m w_k = 1$  leads to the multi-criteria preference degree  $\pi(\mathbf{g}_i, \mathbf{g}_j)$ .

Based on the preference values, the positive and negative multi-criteria net-flows can be calculated. They describe the advantages and disadvantages of one goal location to another. The positive net-flow  $\phi^+$  is the normalized sum of the preference values for the

### 3 Exploration as Multi-Criteria Decision Making Problem

pair comparisons  $\pi(\mathbf{g}_i, \mathbf{g}_j)$ . It indicates how a goal location is outranking all other goal locations.

$$\phi^+(\mathbf{g}_i) = \frac{1}{n-1} \sum_{\mathbf{x} \in G \setminus \mathbf{g}_i} \pi(\mathbf{g}_i, \mathbf{x}) = \sum_{k=1}^m \phi_k^+(\mathbf{g}_i) w_k \quad (3.10)$$

The negative net-flow  $\phi^-$  aggregates the opposing pairs  $\pi(\mathbf{g}_j, \mathbf{g}_i)$ :

$$\phi^-(\mathbf{g}_i) = \frac{1}{n-1} \sum_{\mathbf{x} \in G \setminus \mathbf{g}_i} \pi(\mathbf{x}, \mathbf{g}_i) = \sum_{k=1}^m \phi_k^-(\mathbf{g}_i) w_k \quad (3.11)$$

The value shows how a goal location is dominated by the other goal locations. A value of 1 indicates that the goal location is dominated by all other goal locations. The best goal location has a high positive net-flow value and a low negative net-flow value.

The above defined multi-criteria net-flows  $\phi^+$  and  $\phi^-$  can also be disaggregated, which results in the positive  $\phi_k^+$  and respectively negative unicriterion net-flow  $\phi_k^-$ :

$$\phi_k^+(\mathbf{g}_i) = \frac{1}{n-1} \sum_{\mathbf{x} \in G \setminus \mathbf{g}_i} \pi_k(\mathbf{g}_i, \mathbf{x}). \quad (3.12)$$

$$\phi_k^-(\mathbf{g}_i) = \frac{1}{n-1} \sum_{\mathbf{x} \in G \setminus \mathbf{g}_i} \pi_k(\mathbf{x}, \mathbf{g}_i). \quad (3.13)$$

To build a complete ranking of the goal locations the multi-criteria net-flow  $\phi(\mathbf{g})$  is computed by

$$\phi(\mathbf{g}_i) = \phi^+(\mathbf{g}_i) - \phi^-(\mathbf{g}_i) = \sum_{k=1}^m \phi_k(\mathbf{g}_i) w_k \quad (3.14)$$

with the unicriterion net-flow

$$\phi_k(\mathbf{g}_i) = \phi_k^+(\mathbf{g}_i) - \phi_k^-(\mathbf{g}_i) \quad \text{with} \quad \phi_k(\mathbf{g}_i) \in [-1, 1]. \quad (3.15)$$

To illustrate the meaning of the net-flow let us consider two goal locations  $g_i$  and  $g_j$ :

- if  $\phi^+(g_i) \geq \phi^+(g_j)$  and  $\phi^-(g_i) \leq \phi^-(g_j)$ : goal location  $g_i$  outranks goal location  $g_j$ ,
- if  $\phi^+(g_i) = \phi^+(g_j)$  and  $\phi^-(g_i) = \phi^-(g_j)$ : goal location  $g_i$  is indifferent from goal location  $g_j$ ,
- otherwise goal location  $g_i$  and goal location  $g_j$  are incomparable.

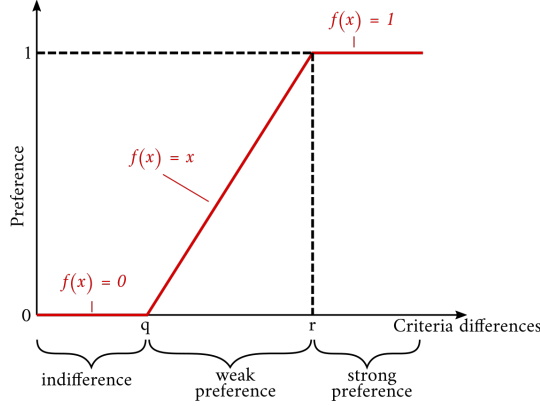
The final decision, the solution to the problem, is the goal location with the highest rank  $\mathbf{g}^*$

$$\mathbf{g}^* = \operatorname{argmax}_{\mathbf{g} \in G} \phi(\mathbf{g}). \quad (3.16)$$

The possibility to separate the multi-criteria net-flow in unicriterion net-flows for a single criterion gives the operator relevant information to understand the decision, which can be useful to adapt the criteria, their preference functions, and weights.

### Developing Criteria

In the model design stage (see fig. 3.2) the criteria to evaluate the goal locations are developed. In this section, we describe how a criterion and its corresponding preference function are set up for PROMETHEE II. A preference function  $P_k(d_k(\mathbf{g}_i, \mathbf{g}_j))$ , indicates how the difference  $d(\mathbf{g}_i, \mathbf{g}_j)$  between two goal locations for one criterion  $c_k$  is mapped to a preference degree between  $g_i$  and  $g_j$ .

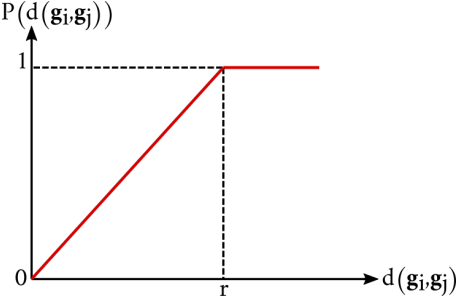
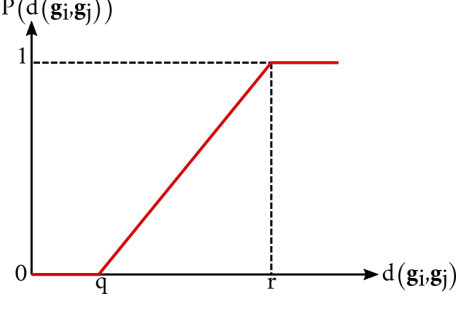
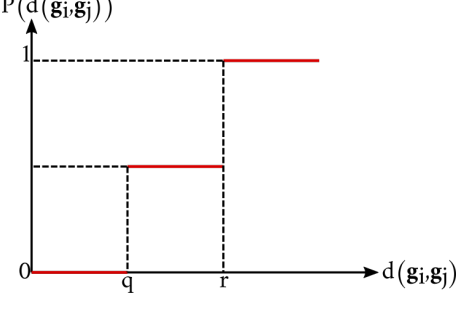
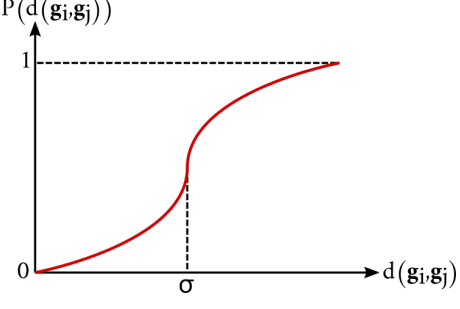


**Figure 3.4:** General description of a preference function. With  $q$  being the indifference value and  $r$  being the preference value.

In fig. 3.4 we illustrate the general structure of a preference function. The y-axis plots the preference value  $P_k(d_k(\mathbf{g}_i, \mathbf{g}_j)) \in [0, 1]$  and the x-axis plots the difference  $d_k(\mathbf{g}_i, \mathbf{g}_j)$  between the criterion values of two goal location. A preference value of  $P_k(d_k(\mathbf{g}_i, \mathbf{g}_j)) = 0$  implies indifference. For example, it doesn't matter to the decision maker, if the distance between the robot and a goal location is 3m or 3.1m. The difference does not lead to a preference for the goal location closer to the robot. Strict preference is indicated by a preference value of  $P_k(d_k(\mathbf{g}_i, \mathbf{g}_j)) = 1$ . For example, consider a goal location at 3 m distance and a goal location that is 20 m away from the robot. The difference of 17 m is quite large and the robot should strictly prefer the closer goal location. A value between  $0 < P_k(d_k(\mathbf{g}_i, \mathbf{g}_j)) < 1$  leads to a weak preference, with an increasing difference and thus increasing preference. To summarize:

- $P(g_i, g_j) = 0$  : indifference between  $g_i$  and  $g_j$ .
- $P(g_i, g_j) \approx 0$  : weak preference of  $g_i$  over  $g_j$ .
- $P(g_i, g_j) \approx 1$  : strong preference of  $g_i$  over  $g_j$ .
- $P(g_i, g_j) = 1$  : strict preference of  $g_i$  over  $g_j$

The indifference value  $q$  marks the point until indifference between two goals is obtained, and the preference value  $r$  marks the point at which strict preference applies. In other words,  $q$  is an indifference threshold indicating the largest deviation, which is considered as negligible by the operator, while  $r$  is the preference threshold, stating the smallest deviation, which is considered as sufficient to generate a full preference of one goal over another by the operator.

Preference function	Definition and description
<p style="text-align: center;"><b>Linear function</b></p> 	$P(d(\mathbf{g}_i, \mathbf{g}_j)) = \begin{cases} 0 & \text{if } d(\mathbf{g}_i, \mathbf{g}_j) \leq 0 \\ \frac{d(\mathbf{g}_i, \mathbf{g}_j)}{r} & \text{if } 0 \leq d(\mathbf{g}_i, \mathbf{g}_j) \leq r \\ 1 & \text{if } d(\mathbf{g}_i, \mathbf{g}_j) > r \end{cases}$ <p>With increasing difference <math>d(\mathbf{g}_i, \mathbf{g}_j)</math> the preference increases linearly until reaching the preference value <math>r</math>. Difference values larger than the preference value <math>r</math> lead to a strict preference of <math>\mathbf{g}_i</math> over <math>\mathbf{g}_j</math>.</p>
<p style="text-align: center;"><b>Linear function with indifference</b></p> 	$P(d(\mathbf{g}_i, \mathbf{g}_j)) = \begin{cases} 0 & \text{if } d(\mathbf{g}_i, \mathbf{g}_j) \leq q \\ \frac{d(\mathbf{g}_i, \mathbf{g}_j) - q}{r - q} & \text{if } q < d(\mathbf{g}_i, \mathbf{g}_j) \leq r \\ 1 & \text{if } d(\mathbf{g}_i, \mathbf{g}_j) > r \end{cases}$ <p>Similar to the linear preference function this preference function defines the weak preference range as linear function. However, the operator defines additionally an indifference range for differences smaller than the indifference value <math>r</math>.</p>
<p style="text-align: center;"><b>Step function</b></p> 	$P(d(\mathbf{g}_i, \mathbf{g}_j)) = \begin{cases} 0 & \text{if } d(\mathbf{g}_i, \mathbf{g}_j) \leq q \\ \frac{1}{2} & \text{if } q < d(\mathbf{g}_i, \mathbf{g}_j) \leq r \\ 1 & \text{if } d(\mathbf{g}_i, \mathbf{g}_j) > r \end{cases}$ <p>With the step function the operator can define an indifference, weak preference and strict preference range. The weak preference range is defined as a single step with equal preference value which is usually set to <math>P(d(\mathbf{g}_i, \mathbf{g}_j)) = 0.5</math>.</p>
<p style="text-align: center;"><b>Gaussian function</b></p> 	$P(d(\mathbf{g}_i, \mathbf{g}_j)) = \begin{cases} 0 & \text{if } d \leq 0 \\ 1 - e^{-\frac{d(\mathbf{g}_i, \mathbf{g}_j)^2}{2 \cdot \sigma^2}} & \text{if } d(\mathbf{g}_i, \mathbf{g}_j) > 0 \end{cases}$ <p>The preference function, based on the Gaussian normal distribution, is monotonically increasing until the turning point <math>\sigma</math>. No strict preference is defined. To define the turning point <math>\sigma</math> it is common to use 50% between the difference of the best and worst criterion value.</p>

**Table 3.1:** Table presenting the most common preference functions for PROMETHEE II.



### 3.3 MCDM Methods for Robotic Exploration

For criteria which can be measured and expressed by a quantitative number the linear preference with or without indifference is a good choice. To choose  $q$  and  $r$  we suggest first calculating the best criterion value and the worst criterion value for a goal location. If this is not possible, as there is no upper limit for the criterion value, the operator has to choose the preference and indifference threshold based on his experience. Starting from these corner cases,  $q$  and  $r$  can be refined empirically. For this, the operator can apply the decision making several times, while varying the preference and indifference values  $q$  and  $p$  and performing an sensitivity analysis, to detect if the chosen values would change the result and another goal has the highest multicriteria preference degree.

In table 3.1 we summarize the most common preference functions used for PROMETHEE II. It is not a complete list, an operator may define any function that best fits a criterion.

#### Reduction of Goals for PROMETHEE II

The main disadvantage of using an outranking method like PROMETHEE II is the required processing power. This is especially critical for planetary exploration missions, where the CPU resources are limited, and all methods have to run online onboard the robot. The pair wise comparison of hundreds of goal locations for several criteria is CPU-intensive as, the time complexity of PROMETHEE II is  $\mathcal{O}(mn^2)$ .

We suggest reducing the number of goal locations to be compared, to improve the run time. In Lehner et al. [11] we first introduced an extension to PROMETHEE II, by which first a subset of goal locations is extracted, before second the multi-criteria preference degree is exclusively calculated for the goal locations in the subset. The goal locations in the subset are selected upon the unicriterion net-flow  $\phi_k(\cdot)$  of one criterion. For this, we derive a threshold  $\tau$  for the unicriterion net-flow  $\phi_k(\cdot)$  of this criterion. All goal locations with a unicriterion net-flow larger than the threshold  $\phi_k(\cdot) > \tau$  are included in the subset and all goal locations with a unicriterion net-flow lower than the threshold  $\phi_k(\cdot) < \tau$  are excluded, i.e. neglected during the further decision process. For the goal locations in the subset the remaining criteria are computed and compared to find the best goal location with the highest multi-criteria preference degree among them.

The idea behind calculating the threshold  $\tau$  is, that a goal location, with a very low unicriterion net-flow value for a single, but strongly weighted criterion has no chance to be the best goal location, nevertheless how large its unicriterion preference values are for the other criteria. To find the threshold  $\tau$  we compute the unicriterion net-flow value for the criterion  $c_m$ , which has the strongest weight among all criteria  $w_m = w_m^{max} > w_m \forall w_k \in W \setminus w_m$ , for all goal locations. Let's assume two goal locations  $\mathbf{g}_a$  and  $\mathbf{g}_b$  with the following properties: Goal location  $\mathbf{g}_a$  has the highest unicriterion net-flow value

$$\phi_m(\mathbf{g}_a) = \phi_m^{max} > \phi_k(\mathbf{g}_j) \quad \forall \mathbf{g}_j \in G \setminus \mathbf{g}_a, \quad (3.17)$$

for the criterion  $c_m$  and the smallest possible value  $\phi_k(\mathbf{g}_a) = -1$  for all other criteria. Alternative  $\mathbf{g}_b$  has the highest possible value  $\phi_k(\mathbf{g}_b) = 1$  for all other criteria  $k \neq m$ . We

### 3 Exploration as Multi-Criteria Decision Making Problem

aim for the minimum unicriterion net-flow value  $\phi_m(\mathbf{g}_b)$  for goal location  $\mathbf{g}_b$ , which is required to hold:

$$\phi(\mathbf{g}_b) > \phi(\mathbf{g}_a) \quad (3.18)$$

For this we substitute eq. (3.14) for the unicriterion values for goal location  $\mathbf{g}_a$  and  $\mathbf{g}_b$  3.14.

$$\sum_{k=1}^{m-1} 1 \cdot w_k + w_m \phi_m(\mathbf{a}_j) > \sum_{k=1}^{m-1} -1 \cdot w_k + w_m \phi_m^{max}. \quad (3.19)$$

and solve for the minimum value  $\phi_m(\mathbf{g}_b)$  for the threshold criterion  $c_m$  to find the threshold  $\tau$ :

$$\tau = \frac{1}{w_m} \left( \sum_{k=1}^{m-1} (-w_k) + w_m \cdot \phi_m^{max} - \sum_{k=1}^{m-1} w_k \right) \quad (3.20)$$

The relation  $\phi(\cdot) > \phi(\mathbf{g}_a)$  holds for goal locations with a higher unicriterion net-flow value  $\phi_m(\cdot) > \tau$  than  $\tau$ , one of those goals is the goal with the highest multi-criteria preference value. For all goal locations with  $\phi_m(\cdot) < \tau$  it holds  $\phi(\cdot) < \phi(\mathbf{g}_a)$  thus, they can be neglected in the further processing.

To calculate the threshold  $\tau$ , we suggest computing  $\tau$  based upon the criterion with the strongest weight  $w_m$ . Further, a dominant goal location with a high unicriterion net-flow value should exist for the criterion, in order to prune many goal locations. To relax this assumption, it would also be possible to combine several criteria and calculate the threshold based on two or more criteria. However, although using more criteria for the threshold calculation could prune more goals, the processing time to calculate the threshold itself increases.

Our method guarantees that the best goal location is among the selected subset, however, as PROMETHEE II is an outranking method, i.e. each goal comparison influences the ranking, it is possible that the exclusion of goal locations from the final decision making can lead to changes in the final ranking. As this only happens when the difference between the multi-criteria preference degree of two goal locations is very small, i.e. the goal locations are equivalent, we assume this as neglect able.

## 3.4 Summary and Discussion

In this chapter, we detailed our general concept of robotic exploration based on MCDM. In contrast to Basilico and Amigoni [32], who first introduced MCDM for exploration, we not only use a MCDM method for decision making, but transfer the whole procedure of MCDM to robotic exploration. We present a complete concept for modelling exploration tasks, especially for planetary exploration. Our approach considers the short communication time slots available to send commands or receive data from the robot in space. To create a ranking among the goals, we presented two different decision making methods. First, we presented the utility functions CALE and SALE, first introduced in

Lehner et al. [10]. Second, we described how we utilize PROMETHEE II. The utility functions are specially designed to model an integrated exploration, as required to conduct use case 1. Both find a trade-off between exploration efficiency, map quality and localization accuracy. Disadvantages of these utility functions are their complexity and non-flexibility. A human operator, might have difficulties parametrizing the functions and understanding the found solution. In contrast to PROMETHEE II the utility functions can not be applied for the other use cases. With PROMETHEE II any criterion can be added to the decision process to model the desired exploration behavior to conduct any use case. Although Taillandier and Stinckwich [33] proposed to use PROMETHEE II, they similar to Basilico and Amigoni [32] lack the general formulation of robotic exploration as MCDM. Further, Taillandier and Stinckwich [33] do not consider the high resources required for the pair-wise comparison, which we solve by extending PROMETHEE II with our subset extraction method.



## 4 Modeling Planetary Exploration Missions

In this chapter, we model the four robotic exploration use cases: (1) autonomous exploration (section 4.2), (2) drive by science (section 4.3), (3) autonomous search (section 4.4), (4) multi-robot exploration (section 4.5). We model each use case using the general MCDM exploration concept introduced in section 3.2. First, we structure the problem, second, we conduct the model design stage and identify alternatives and develop criteria for each use case. In section 4.1, we detail how we sample goal locations, which corresponds to the model implementation stage. Further, we present a classification of exploration criteria in section 4.6, after describing the use cases. Finally, we conclude the chapter by summarizing and discussing our approach.

### 4.1 Sampling Goal Locations

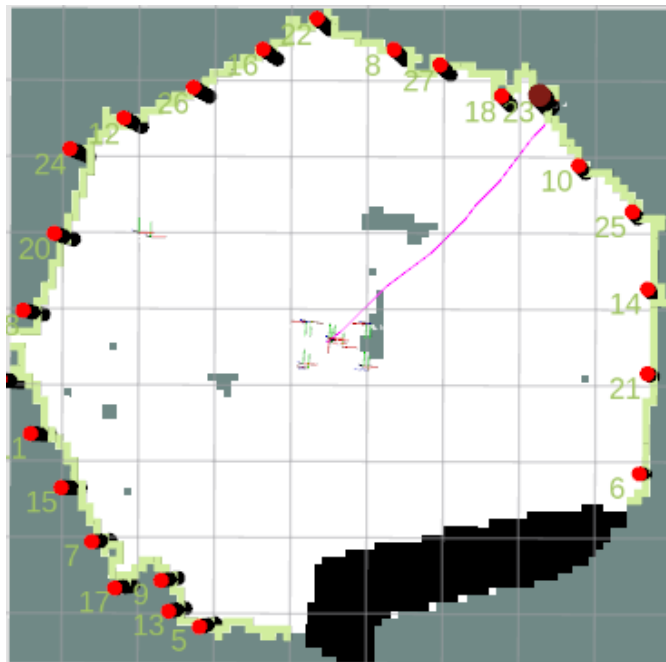
In this section, we describe our methods to sample goals. Let  $\mathbf{g} \in G$  be the set of goal locations, with each goal location being a vector  $\mathbf{g} \in \mathbf{SE}(3)$  of a potential robot pose. We sample exploration goals  $\mathbf{g}_{ex}$  and re-localization goals  $\mathbf{g}_{re}$ . At exploration goals a robot can gain new information about the environment. Re-localization goals represent locations a robot visited before, thus a re-localization can be performed.

#### Exploration Goals

To sample exploration goals, we apply a frontier-based algorithm [4]. The exploration goals are placed at so called frontiers, which build the boundary between unknown and known space, as illustrated in fig. 4.1. Let's consider a 2D occupancy grid map, for which each map cell can have the state *free*, *occupied*, or *unknown*. Cells with the state *free* or *occupied* are known to the robot, i.e. have already been observed by the robot's sensors. A free cell is traversable by the robot, whereas an occupied cell indicates an obstacle. To detect frontiers, we first identify all frontier cells, then, in a second step, group the frontier cells and, third, post-process the generated frontiers. To efficiently detect the frontier cells we apply an approach similar to the Naive Active Area (NAA) method introduced by [94]. This approach only detects frontier cells in the so called active area, which is defined by the scans taken by the robot since the last frontier update. In our case, the active area at time  $A(t)$  is defined by the submaps the robot traversed since the last exploration step  $t^{-1}$ . The active area  $A(t)$  is then the bounding box around these submaps. The cells outside the active area are assumed to have the same status as at the previous exploration step  $t^{-1}$ . Thus, frontier cells outside the active area can be added to the current set of frontiers cells  $F_c(t)$  without updating them. The cells in the active area  $A(t)$  are re-evaluated and the newly detected frontier cells are added to the unordered

set of frontier cells  $F_c(t)$ , too. Following this, we group and sort the frontier cells to build sorted and connected frontier lines. We loop through the set of unordered frontier cells  $F_c(t)$  until we find a start cell. A start cell is defined as frontier cell, having exact one frontier cell as neighbor. Beginning at the start cell we repeatedly find the nearest neighbor until another start cell, i.e. the end of the frontier is reached. This procedure is conducted until all frontier cells in the unordered set  $F_c(t)$  are processed. Each detected frontier  $fr$  is finally added to the current set of frontiers  $F(t)$ . Finally, we refine the frontiers in a post-processing step and place exploration goals  $\mathbf{g}_{\text{ex}}$  at the middle of each frontier  $fr \in F(t)$ . In scenarios with few obstacles, frontiers can occur that are too long to be observed from a single point of view. We recursively split long frontiers until no single frontier is longer than 5 m. We determined this limit from our sensor's observation range and its decreasing quality of measurements for larger distances.

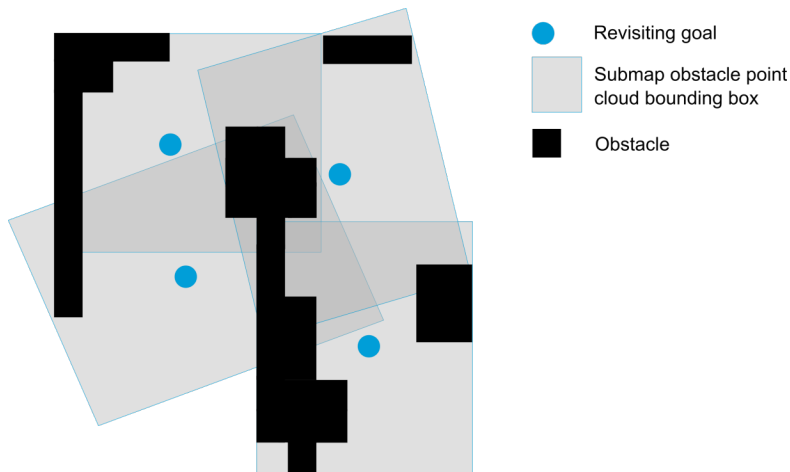
The NAA approach does not account for changes in the map outside the active area. However, this could happen when a loop closure is conducted, by which the map is optimized. We recognize loop closures and apply a naive frontier detection over the whole map at the next exploration step, to also update the frontier cells outside the active area.



**Figure 4.1:** 2D occupancy grid-map showing the frontier between known and unknown space, with sampled exploration goals along the frontiers. One exploration goal is calculated for each frontier, however long frontiers are split in smaller frontiers to sample more exploration goals. The pink line illustrates the current path of the robot (depicted with the robot transformation tree) to the chosen exploration goal (larger goal with dark red color).

## Re-localization Goals

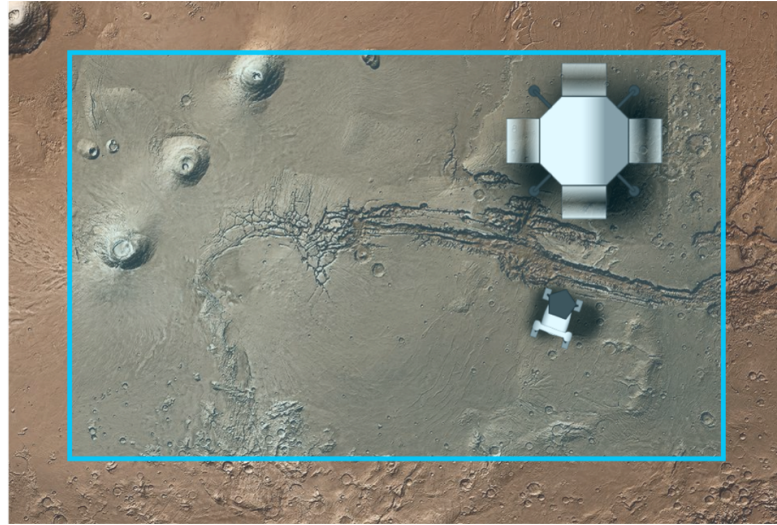
In our underlying mapping and localization system, re-localization is based on the optimization of the SLAM graph by a loop closure constraint. As described in section 2.1.3, we perform a pairwise matching between submaps to generate loop closure constraints. To generate a loop closure, the robot has to traverse areas already sensed. By visiting a previously generated submap, a new submap overlapping with this existing submap is generated. By matching the new and the old submap, a loop closure can be created. The pairwise matching between submaps is based on the 3D geometric structure, more precisely the obstacle points. To achieve a high overlap between the obstacle points of the existing submap and the submap to be created at the re-localization goal, we sample the re-localization goals in the middle of the bounding box of the obstacle points of the existing submap, as illustrated in fig. 4.2.



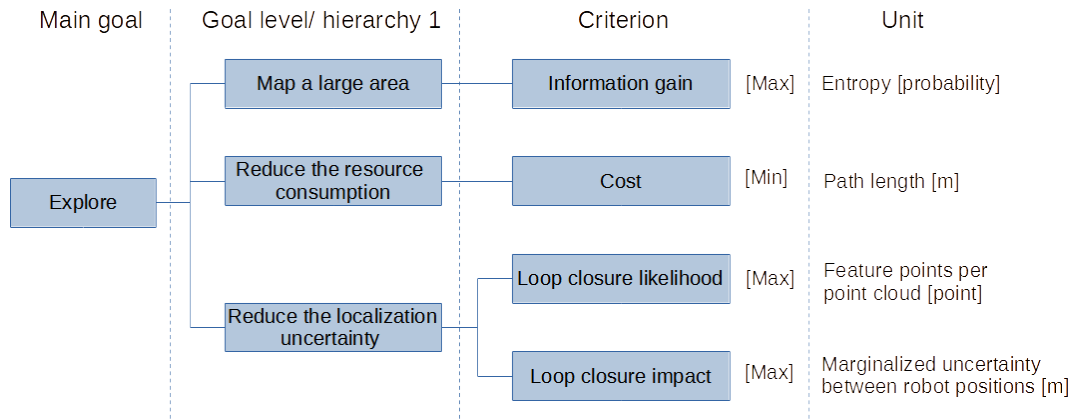
**Figure 4.2:** Illustration of the placement of the re-localization goals. We place re-localization goals (blue circles) in the center of the bounding box of the obstacle points (grey rectangle).

## 4.2 Use Case 1 - Autonomous Exploration

In use case 1, a robot has to explore a prior unknown ROI, as illustrated in fig. 4.3. The objective of the mission task is to build a map of the environment of the ROI, which can be used to plan further science investigations. For this, a complete map with high quality is required. Further, it is important to keep the resource consumption within the limited mission budget.



**Figure 4.3:** Illustration of use case 1 - autonomous exploration. The goal of an autonomous exploration mission is to survey the environment. For this the robot explores a predefined ROI. The mission is finished if the area within the ROI is mapped.



**Figure 4.4:** Goal system for use case 1 - autonomous exploration. The main goal is to explore the ROI, which can be subdivided in the three subgoals, mapping a large are, while reducing the resource consumption and the localization uncertainty. To evaluate each goal  $g$  for the subgoals, we apply the criteria  $c_{IG}$ ,  $c_{cost}$ ,  $c_{li}$  and  $c_{li}$

From the requirements and the main objective, we derive the objectives for a single exploration action, illustrated in fig. 4.4. At each action, the robot should map a large prior unknown area, try to limit the necessary resources, and to reduce the localization uncertainty. In order to gain new information, the robot has to move towards unknown areas, therefore, we sample exploration goals, as described in section 4.1. To reduce the localization uncertainty, the robot has to re-localize itself, which is only possible in regions already visited. Thus, we additionally sample re-localization goals.



We model an integrated exploration behavior by applying four criteria to achieve the sub-objectives. To measure the new information which can be gained by an exploration action, we evaluate the information gain criterion  $c_{IG}$  (section 4.2.2). To keep the resources at each step low, we consider the distance from the current robot position to a goal location, with the cost criterion  $c_{cost}$  (section 4.2.1). For re-localization, we actively trigger loop closure to optimize the underlying SLAM graph of our mapping system (see section 2.1.3). We apply the two criteria loop closure likelihood  $c_{ll}$  and loop closure impact  $c_{li}$  to evaluate the re-localization performance at a goal. In the following, we detail the implementation of the four criteria.

### 4.2.1 Cost Criterion

The cost criterion  $c_{cost}(\mathbf{g})$  describes the costs required to move to a goal location starting from the current robot position. We define the costs as the length  $len$  of the shortest traversable path from the current robot position  $\mathbf{p}_r$  to the goal  $\mathbf{g}$  [11].

$$c_{cost}(\mathbf{g}) = len_{min}(\mathbf{g}, \mathbf{p}_r) \quad (4.1)$$

To calculate a discrete approximation of the shortest path we apply a wavefront planner [95], using the wavefront expansion algorithm. We used the simple wavefront planner, as it gives the shortest path between two points and it is easy to implement. To estimate the costs of moving to a goal the estimation with wavefront planner is sufficient. Starting at the current robot position the nearest neighbors (8 neighborhood) in a circle around the robot are analysed. If a neighbor cell is traversable, the path costs are increased by one. At each step, the circle is expanded and the more distant neighbors are analyzed. For our application, it is not necessary to find a path to the goal, only the length of the path is of interest. We benefit from this fact and expand the wave until the last goal is reached. Thus, by only expanding the wavefront once, we can calculate the path length for all goals as shown in fig. 4.5.

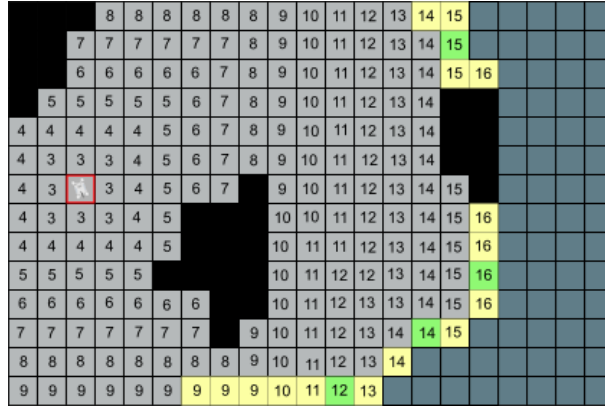
### 4.2.2 Information Gain Criterion

The most important criterion for an efficient exploration is the information gain  $c_{IG}$ . It is a measure of how much new information a robot gains at a goal location.

We estimate the information gain from the probabilistic 3D voxel map provided by the underlying mapping system (see section 2.1.3). Let  $\chi \in \mathcal{X}(\mathbf{g})$  be a voxel in  $\mathcal{X}$  in the neighborhood of a goal  $\mathbf{g} \in G$ . Then, the  $IG$  is the sum of the differences between the current entropy  $H(\chi)$  and the expected entropy  $E[H'(\chi)]$  at each voxel  $\chi \in \mathcal{X}(\mathbf{g})$ . We calculate the current entropy  $H(\chi)$  and the expected entropy  $E[H'(\chi)]$  according to [96].

$$c_{IG}(\mathbf{g}) = IG = \sum_{\chi \in \mathcal{X}(\mathbf{g})} H(\chi) - E[H'(\chi)] \quad (4.2)$$

For a more reliable information gain we take into account that the probability of observability  $Q_o(\chi)$  decreases with an increasing distance from the stereo camera sensor,



**Figure 4.5:** Illustration of the result of the wavefront expansion algorithm. Starting at the robot position (marked red) the wave is circularly expanded through the known space (light gray) until all goals (green) at the frontiers (yellow) to the unknown space (dark gray) are reached. Obstacles are marked black.

as described in our publication Lehner et al. [10]. The probability observability  $Q_o(\chi)$  is defined as

$$Q_o(\chi) = \prod_{\chi' \in \mathcal{X}_{ray}(\chi)} p(\chi') \quad (4.3)$$

with  $\mathcal{X}_{ray}(\chi)$  being the set of voxel, which intersect the casted ray and being nearer to the origin than  $\chi$  and  $p(\chi)$  as the posterior of  $\chi$ . Applying the probability of observability  $Q_o(\chi)$ , we extend eq. (4.2) to [10]

$$c_{IG}(\mathbf{g}) = IG = \sum_{\chi \in \mathcal{X}(\mathbf{g})} H(\chi) - Q_o(\chi) \cdot E[H'(\chi)] \quad (4.4)$$

For the estimation, we simulate a horizontal 360-degree sensor swipe with a fixed angular resolution and a 40 degree vertical FoV [10] in our experiments (section 6.1). Only the measurements in a range of 4m are included, as the stereo uncertainty grows at farther distances, as shown by Brand and Schuster [97].

### 4.2.3 Loop Closure Likelihood Criterion

In our integrated exploration approach, a robot can re-localize itself by generating loop closures. The loop closure likelihood criterion  $c_{ll}$  evaluates how likely it is, that a loop closure is generated when visiting a goal location. We define the loop closure likelihood  $c_{ll}$  as a heuristic measure

$$c_{ll}(\mathbf{g}) = p(\mathcal{L}(\mathbf{g}, \mathbf{g}_\zeta)) \quad (4.5)$$

with  $p(\mathcal{L}(\mathbf{g}, \mathbf{g}_\zeta))$  being the probability, that a loop closure  $\mathcal{L}$ , between the new goal location  $\mathbf{g}$  and a goal visited in the past  $\mathbf{g}_\zeta$  at exploration step  $\zeta$ , happens.

The implementation of the heuristic depends on how loop closures are computed in the underlying mapping framework. Our work is based on the 6D Graph SLAM introduced

by Schuster et al. [8], briefly described in section 2.1.3. Loop closure constraints are generated by pairwise matching of submaps. In turn, the matching is based on the 3D geometric structure of the submaps, whereas the obstacle points are employed as 3D features. Thus, a successful submap match requires two overlapping submaps with a large number of feature points, which are equally distributed. The heuristic of the loop closure likelihood describes in our case the matching quality of the existing submap at a goal location. We consider the distribution of feature points  $\mathcal{D}$  and the number of obstacle points  $n_o$  in the existing submap. We compute the distribution  $\mathcal{D}$  by applying a nearest neighbor analysis, which provides a numerical value describing the extent to which a set of points is clustered or uniformly spaced. To compute  $\mathcal{D}$  we first calculate for each obstacle point in a submap the distance to its nearest neighbour, i.e. the closest obstacle point. Then  $\mathcal{D}$  is defined as follows

$$\mathcal{D} = \frac{\overline{d_n}}{0.5\sqrt{n_o/a_s}} + \mathcal{D} \quad (4.6)$$

where  $\overline{d_n}$  is the mean value of the computed nearest neighbour distances and  $a_s$  is the area of the submap. Then the loop closure likelihood criterion is computed by the ratio between the number of obstacle points  $n_o$  in a submap and the total number of points  $n$  in a submap, to account for the number of obstacle points and the distribution of the obstacle points  $\mathcal{D}$  [10].

$$c_{ll}(\mathbf{g}) = \frac{n_o}{n} + \mathcal{D} \quad (4.7)$$

A uniform distribution of feature points and a large number of feature points compared to all points, increase the loop closure likelihood.

#### 4.2.4 Loop Closure Impact Criterion

The loop closure likelihood is a measure of the success of an intended loop closure but does not consider the effect of the constraint on the SLAM graph. Usually, re-localization requires the robot to revisit goal locations visited earlier. As this interrupts the exploration of the ROI we believe it is essential to take the reward of the loop closure constraint into account. In our publication Lehner et al. [10] we first introduced, the novel criterion loop closure impact. We define it as the difference between the current uncertainty  $s_{\mathbf{g}}$  and the expected uncertainty  $E(s_{\mathbf{g}}(\mathcal{L}))$  after optimization when closing the loop  $\mathcal{L}$ .

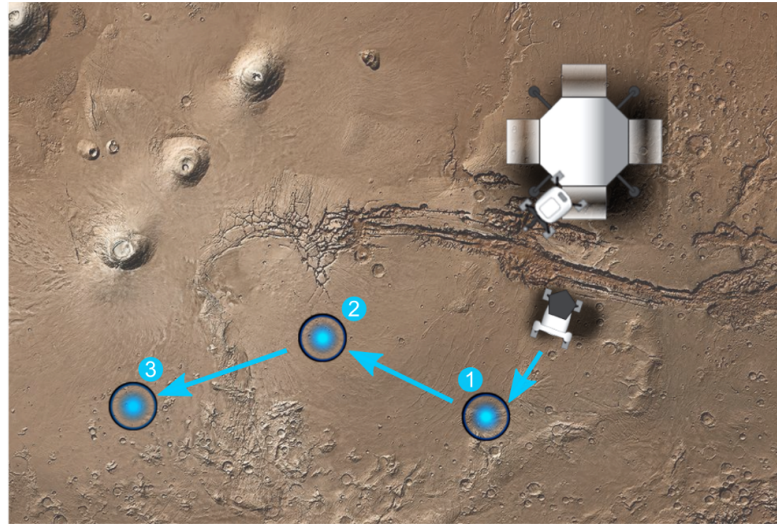
$$c_{li}(g) = s_{\mathbf{g}} - E(s_{\mathbf{g}}(\mathcal{L})) = s^2(\mathbf{p}_r) - s^2(\mathbf{g}_i) \quad (4.8)$$

With  $s_{\mathbf{g}}$  being the uncertainty of the submap at the current robot position  $s^2(\mathbf{p}_r)$  and  $E(s_{\mathbf{g}}(\mathcal{L}))$  being the uncertainty at the submap goal location  $s^2(\mathbf{g})$

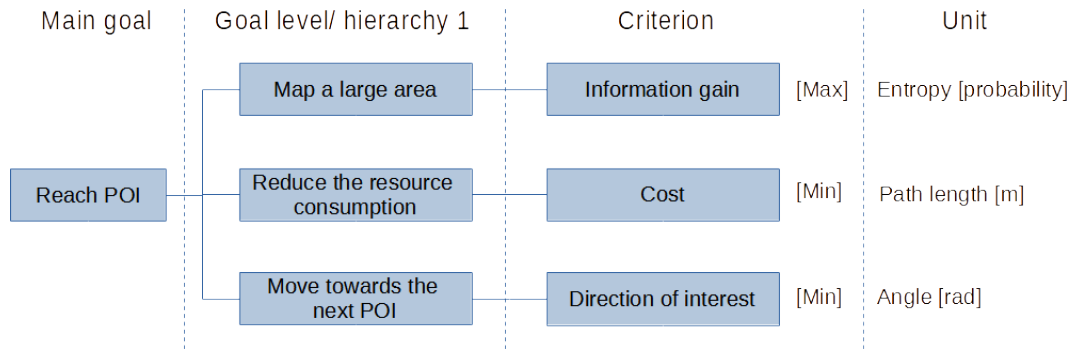
The loop closure impact  $c_{li}$  increases with increasing difference between the current uncertainty  $s_{\mathbf{g}}$  and the expected uncertainty  $E(s_{\mathbf{g}}(l))$ . That, usually inclines that the existing submap is quite old, thus farther away from the current robot position.

### 4.3 Use Case 2 - Drive-by Science

Use case 2, illustrated in fig. 4.6, depicts the common task of driving to a predefined POI. Instead of driving directly to the POI, often drive-by science or opportunistic science methods are applied, which try to increase the information about the environment, while moving to the POI. That means, the robot is allowed to depart from the planned path to sense a large part of the environment on its way.



**Figure 4.6:** In the drive-by science mission, the robot has to visit several POI in sequential order. The mission is finished, after reaching the last POI. In this example three POI are defined.



**Figure 4.7:** Goal system for use case 2 - drive-by science. Additionally, to the subgoals, mapping a large area, while reducing the resource consumption, we define the subgoal, moving towards a predefined global goal (POI). We evaluate the latter, with the direction of interest criterion  $c_{doi}$ .

We propose to apply a robotic exploration in order to conduct a mission in the fashion of a drive-by science. In fig. 4.7 we present the goal system, which depicts a directed

exploration behavior, required to conduct a drive-by science missions with robotic exploration. Starting from the main objective, reaching the next POI, we define three sub-objectives. To reach the POI, the robot has to move in the direction of the POI at each exploration action. To gain the most information about the environment while moving to the POI the robot should map a large part of the environment at each exploration step. Further, it is important to keep the resources within the mission budget, e.g. limiting the distance the robot has to drive.

We only sample exploration goals and no re-localization goals, as revisiting re-localization goals contradicts the main objective of moving to a POI. Other than in use case 1, the map created during the traverse is not necessarily used to plan further science investigation. To approach the objectives, we apply the three criteria, information gain criterion  $c_{IG}$ , cost criterion  $c_{cost}$ , and direction of interest criterion  $c_{doi}$ .

With the information gain criterion  $c_{IG}$ , the goal locations are evaluated for their potential of mapping a large unknown part of the environment. The cost criterion evaluates the distance to the goal locations and is thus a measure for the required energy, i.e. resources. The direction of interest criterion  $c_{doi}$  evaluates if a goal location is lying into the direction of the POI. In the following, we describe the novel direction of interest criterion  $c_{doi}$  in detail. For the information gain criterion  $c_{IG}$  and the cost criterion  $c_{cost}$  please refer to section 4.2.2 and section 4.2.1.

### 4.3.1 Direction of Interest Criterion

To direct the robot toward the POI, we present the novel exploration criterion direction of interest  $c_{doi}$ , which we first introduced in our publication Lehner et al. [11]. It can be used to direct the robot into any predefined direction, which we call direction of interest (DOI), while exploring. The direction of interest criterion  $c_{doi}(\mathbf{g})$  is defined as the angle between the directional vector  $\mathbf{v}_{doi}$  pointing into the current DOI and the directional vector  $\mathbf{v}_{\mathbf{g}}$  pointing from the current robot position  $\mathbf{p}_r$  to the goal  $\mathbf{g}$  [11]:

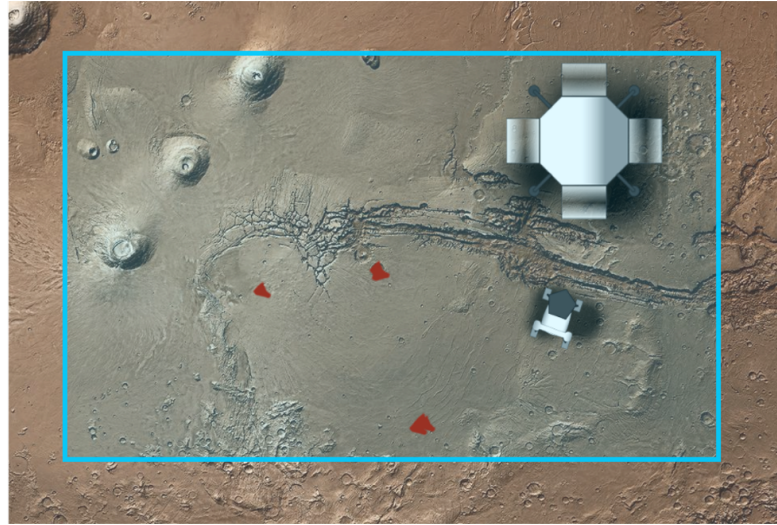
$$c_{doi}(\mathbf{g}) = \cos^{-1} \left( \frac{\mathbf{v}_{doi} \circ \mathbf{v}_{\mathbf{g}}}{|\mathbf{v}_{doi}| \cdot |\mathbf{v}_{\mathbf{g}}|} \right) \quad (4.9)$$

## 4.4 Use Case 3 - Autonomous Search

In use case 3, a robot has to search in a prior unknown ROI for a Feature of Interest (FOI), as illustrated in fig. 4.8.

To find the FOI in the ROI we could apply the exploration behavior modelled for use case 1 (see section 4.2). However, in contrast to an exploration of a ROI, as in use case 1, the main objective is not to build a complete map, but to find a FOI. Eventually, the robot would find the FOI while building a complete map of the ROI. However, this approach is not efficient, in the worst case the robot detects the FOI with its last sensor swipe.

In fig. 4.9 we present the goal system, which structures the exploration problem of use case 3. Except for the new sub-objective, finding a FOI, the goal system is similar to



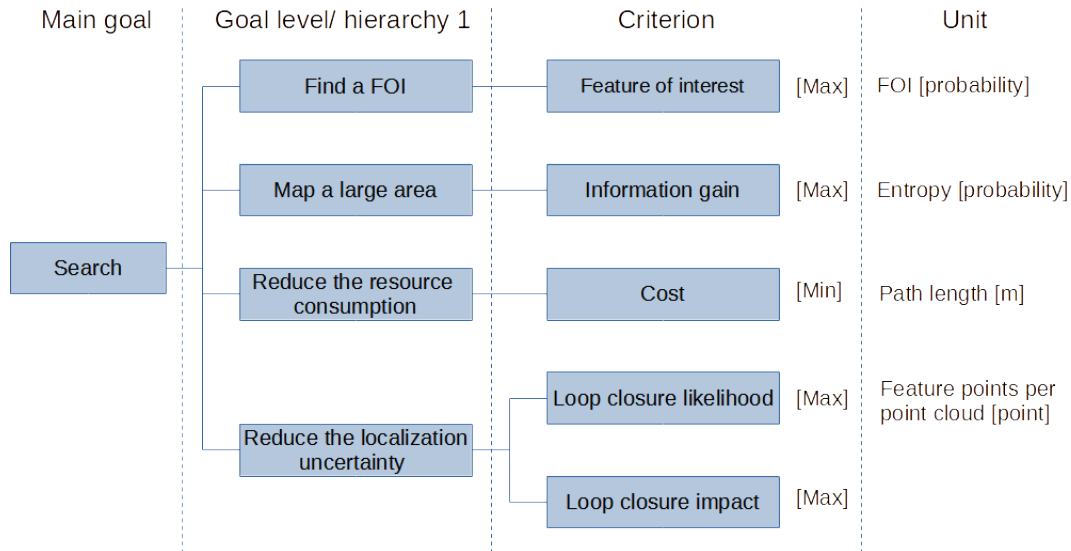
**Figure 4.8:** The goal of an autonomous search mission is to find a FOI in a ROI. The robot explores the ROI until one instance of the FOI is detected.

the one of use case 1. We sample goal locations for exploration and for re-localization. The exploration goals, induce new information about the environment, which increases the probability of detecting a FOI. As the search for a FOI in a large ROI can require a long time, it might be necessary for the robot to re-localize itself, thus we also sample re-localization goals. To approach all sub-objectives, we evaluate the information gain criterion  $c_{IG}$ , the cost criterion  $c_{cost}$ , the loop closure likelihood criterion  $c_{ll}$ , the loop closure impact criterion  $c_{li}$ , and the novel feature of interest criterion  $c_{foi}$ . We call the modeled exploration behavior informed exploration. The robot is equipped with a detector and the knowledge of how to detect a FOI.

In the following, we state, which geologic features are of interest (section 4.4.1), how we detect these (section 4.4.2) and, how we evaluate the potential of detecting a FOI with our feature of interest criterion  $c_{foi}$  (section 4.4.3).

#### 4.4.1 Features of Interest

To motivate our approach, we describe in this section, which features are of interest at Mars and which conclusion scientists can draw from these. NASA's Perseverance is exploring the Jezero Crater since the beginning of 2021. Its main tasks are to study geology and astrobiology, collect samples and prepare for humans to enter Mars. To reveal the geologic history of the Jezero Crater, Perseverance detects and analyses geologic features of interest. A FOI could be an individual rock of a certain type, a novel geologic feature, outcrops, or the contact area between different geologic units. In fig. 4.10 we show the rocks "Máaz" and "Rochette" which were selected by scientists to be of high interest. The escarpments, short scapsrs, shown in figure 4.10 are outcrops that are of importance, as older rock layers are exposed. Further, they present a contact



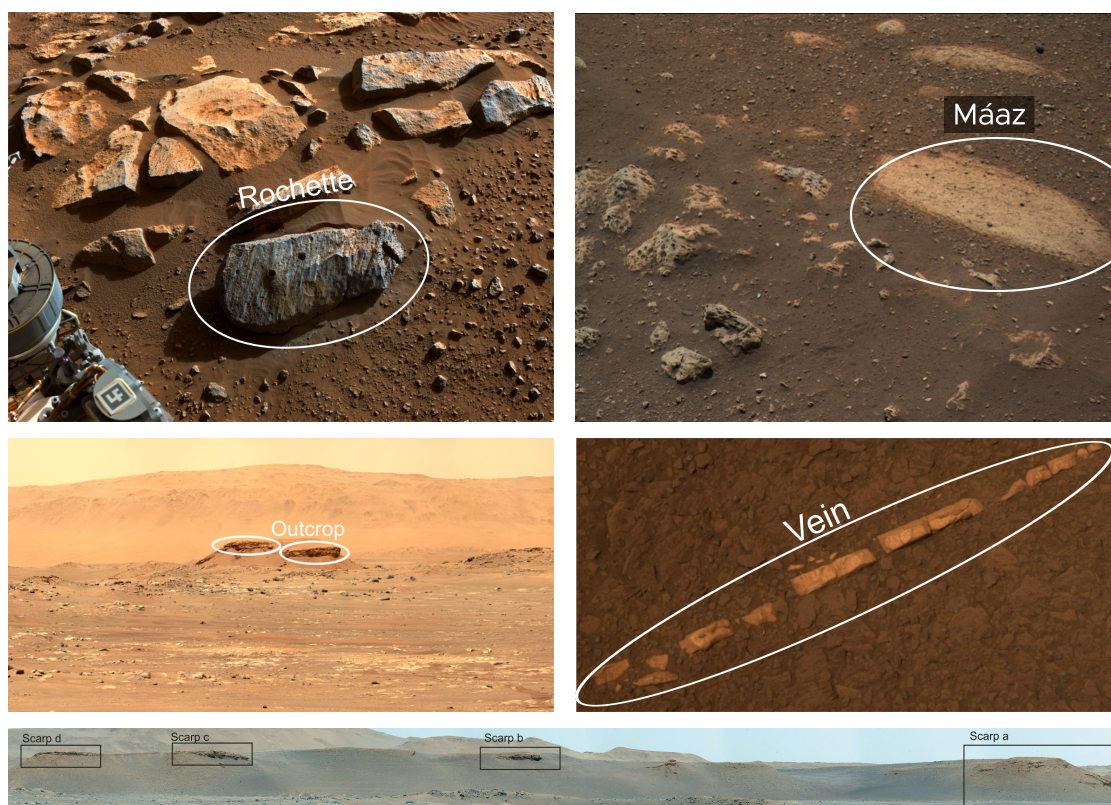
**Figure 4.9:** Goal system for use case 3 - autonomous search. We add the subgoal find feature of interest to the express the search task and introduce the criterion feature of interest  $c_{foi}$ . It is a measure for the probability of detecting a FOI when moving to a goal  $g$ .

zone between two geologic units, where one unit is underlying another. An investigation of the erosion process of the overlying unit, helps to understand the evolution of the landscape.

#### 4.4.2 Detection of Features of Interest

To detect a FOI we perform a full 360° camera swipe before computing the next goal location. On each image captured, we apply an image detector to identify a FOI. For the identification, any appropriate image detector could be used.

In our publication, Schuster et al. [15], we present preliminary tests on using a scientific camera, the Science Cam shown in fig. 4.11, to detect specific geologic features. The author of this thesis elaborated on this part in the publication. The concept of the Science Cam, is similar to the PanCam of the ExoMarsRover, presented in [98]. Our Science Cam consists of two wide-angle cameras (LWAC, RWAC) in a stereo setup with spectral filter wheels, one narrow-angle camera, and one thermal camera. The filter wheels for the LWAC and RWAC are each comprised of three color filters (100 nm bandwidth) and six narrow band (10 nm bandwidth) geology filters (*Geo1-Geo12*), across a range of 440 – 660 nm for LWAC and 720 – 1000 nm range for RWAC. To detect and distinguish different rock types from the moon analogue site on Mt. Etna we combined images captured with different filters. As the detection of a FOI is beyond the scope of this thesis, we only present preliminary results of a science product in fig. 4.12. We combined



**Figure 4.10:** Feature of interest (FOI) in a robotic Mars Mission. Top right: a rock called Rochette, analysed by NASA's Perseverance. Top left: a rock called Máaz, which was analysed by the SuperCam of Perseverance. Middle right: Landscape image showing an interesting outcrop. Middle left: A mineral vein of scientific interest detected by Curiosity. Bottom: Several escarpments in the Jezero Crater. Image Credit: NASA/JPL-Caltech.

several filters found empirically to create the science product:

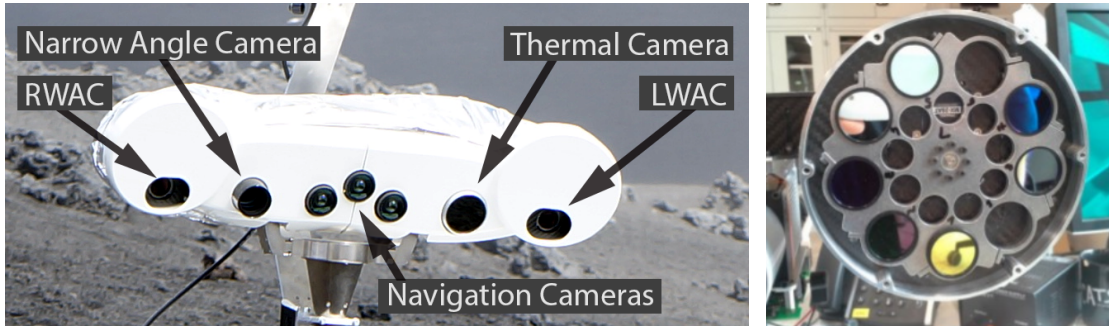
$$\text{RGB} = \left( \text{Geo4}, \frac{\text{Geo1} - \text{Blue}}{\text{Geo1} + \text{Blue}}, \frac{\text{Geo6} - \text{Geo1}}{\text{Geo6} + \text{Geo1}} \right) \quad (4.10)$$

As highlighted by figure fig. 4.12 the rocks can be discriminated in the science product, but look similar in the visible range.

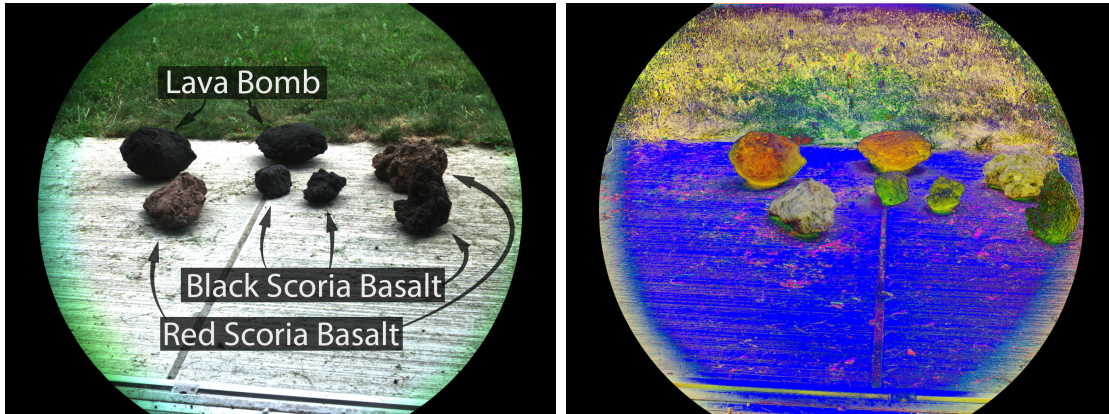
#### 4.4.3 Feature of Interest Criterion

Let us imagine a robot has the task to find a lava bomb as shown in fig. 4.12. To enable an autonomous search for a lava bomb, we include the results of the feature detection in the decision process on 'where to move next?'. To evaluate the probability of finding a FOI, when moving towards a goal location, we introduce our novel feature of interest criterion  $c_{foi}$ .





**Figure 4.11:** Left: ScienceCam with LWAC (Manta G-145B NIR), RWAC (Manta G-145B NIR), narrow-angle camera (Manta G-505C) and thermal camera (Xenics Serval-640); Right: Left filter wheel with spectral bandpass filters. Image Credit: Schuster et al. [15].

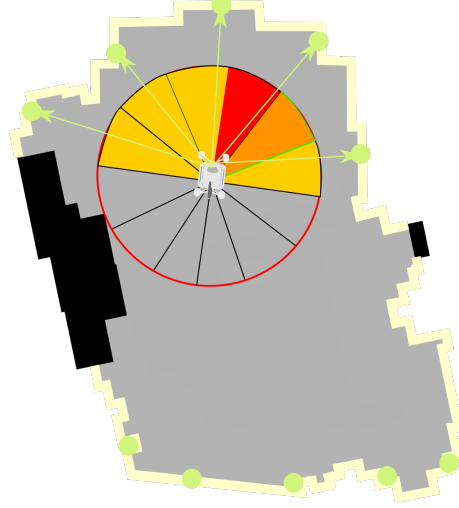


**Figure 4.12:** Left: RGB image of volcanic rocks from Mt. Etna. Right: generated science product. In the science product the lava bomb and the black scoria basalt can be distinguished clearly. Image credit: Schuster et al. [15].

It describes the probability of detecting a FOI in a certain direction. We compute, a polar interest descriptor, which encapsulates this probability for each possible movement direction from the robot’s current position. The computation of the polar interest descriptor includes four steps: First, we apply a FOI image detector on each image captured during the sensor swipe. We suggest using an image detector, which returns the probability for each pixel of containing a FOI. Second, we reconstruct the 3D points of the ‘pixels of interest’ and transform them into a spherical coordinate system. Third, we construct a polar histogram storing the probability of detecting a FOI in bins, which describe the possible movement directions.

For each pixel  $px$  with a probability higher than 80% of showing a FOI we reconstruct the Cartesian 3D coordinates  $P_c(X, Y, Z)$  in the camera frame using a synchronized stereo image pair. For our local mapping, described in section 2.1.3, we only include the 3D points in 4 m range from the camera sensor. To be able to detect features of interest

in the far range we include all points in a range of 20 m. As the reconstruction error from stereo pairs increases with increasing distance [99], we include the uncertainty of the 3D reconstructed point in our calculations of the polar interest descriptor.



**Figure 4.13:** The polar histogram with its bins centered at the camera frame, that the azimuth  $\theta$  and distance  $\rho$  can directly be plotted to the histogram.

As stated by Hirschmüller [99] the error in  $Z$  is proportional to  $Z^2$  if  $X = 0$  and  $Y = 0$ . Whereas, the error on  $X$  and  $Y$  are linearly dependent on  $Z$ . As the largest error in 3D reconstruction is on the depth component, i.e. the distance  $l_c$  between the point  $P_c$  to the camera sensor,

$$l_c = \sqrt{X^2 + Y^2 + Z^2} \quad (4.11)$$

we are interested in the error on  $l_c$ . Following Hirschmüller [99], the error in the distance can be approximated with:

$$\delta l_c \approx \delta p \cdot \frac{Z \cdot l_c}{f \cdot t} \sqrt{2} \quad (4.12)$$

To easily read out the direction to the features of interest, we transform the 3D cartesian points  $P_c(X, Y, Z)$  to spherical coordinates  $P_c^*(\rho, \theta, \varphi)$  with:

$$\rho = l_c = \sqrt{X^2 + Y^2 + Z^2} \quad (4.13a)$$

$$\varphi = \arctan \frac{\sqrt{X^2 + Y^2}}{Z} \quad (4.13b)$$

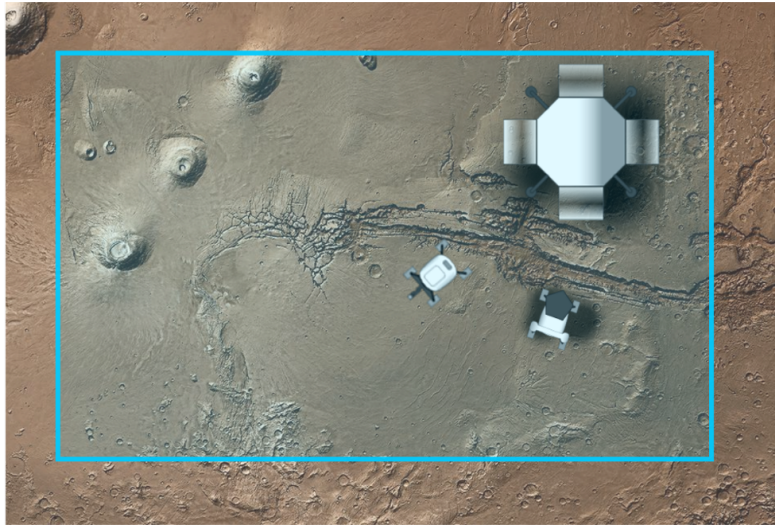
$$\theta = \begin{cases} \arctan \frac{X}{Y} & \text{if } X \geq 0 \\ \arctan \frac{X}{Y} + \pi & \text{if } X < 0 \\ \frac{\pi}{2} & \text{else} \end{cases} \quad (4.13c)$$

Where  $\theta$  is the azimuth,  $\varphi$  is the polar angle and  $\rho$  is the radial distance. To fill the polar histogram we only take into account the azimuth  $\theta$  of the spherical coordinates and the distance  $\rho$ , including its estimated error  $\delta\rho$ . We consider the distance, as it greatly influences the quality of the feature detection. For each bin, we sum up the probability for each point within the bin, of showing a FOI. Thereby, we reduce the probability of being a FOI by multiplying it with the distance error. To take the error of the estimated distance to the points into account, we reduce the probability of a point showing a feature of interest by multiplying it with the error  $\delta\rho$ . In fig. 4.13 we illustrate, how we evaluate the goal locations with the feature of interest criterion  $c_{foi}(\mathbf{g})$ . We calculate the directional vector  $\mathbf{v}_{\mathbf{g}}$  pointing from the current robot position to the goal location  $\mathbf{g}$  and transform it into the camera frame. The value of the feature of interest criterion  $c_{foi}$  can then directly be read out from the polar histogram.

$$c_{foi}(g) = p(bin(\mathbf{v}_{\mathbf{g}})) \quad (4.14)$$

## 4.5 Use Case 4 - Multi-robot Exploration

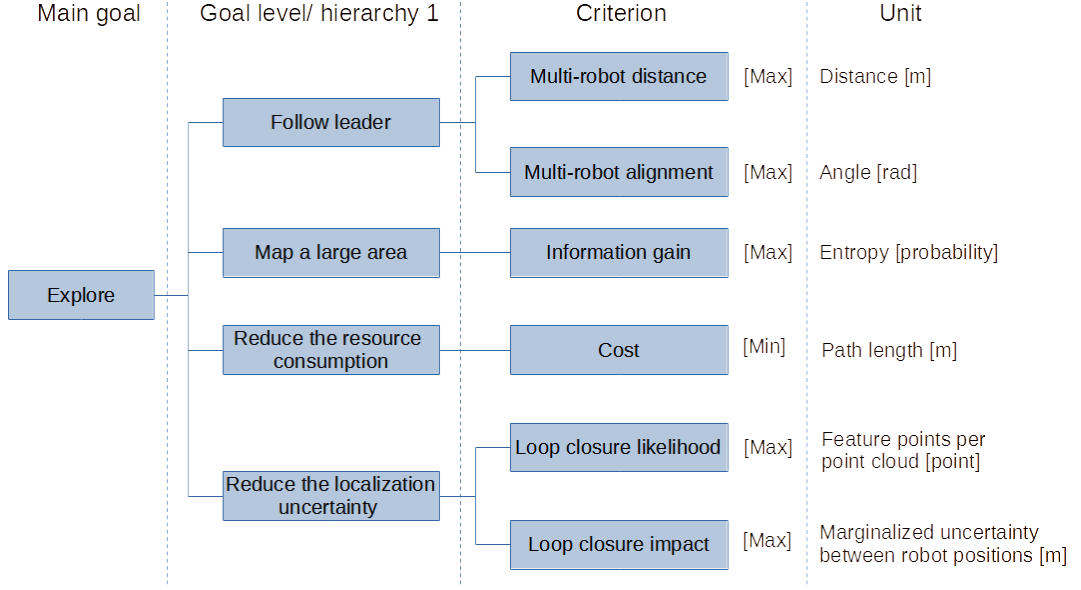
Use case 4, illustrated in fig. 4.14, represents a multi-robot exploration of a ROI. We present a leader-follower coordination strategy for two robots, applying the similar general exploration concept based on MCDM we introduced in chapter 3, and we applied to model the exploration behavior to conduct use case 1, 2 and 3.



**Figure 4.14:** The goal of an autonomous exploration mission is to survey the environment. Two robots exploring together a ROI can increase the exploration efficiency significantly.

In our leader-follower approach, the leader robot determines the exploration direction and the follower robot tries to move in the same direction. In our use case 4, the two robots have different capabilities. One rover has a scientific camera and is able to detect

#### 4 Modeling Planetary Exploration Missions



**Figure 4.15:** Goal system of the follower robot in use case 4 - multi-robot exploration.

a FOI, the other, is equipped with a manipulator and can take a probe. Therefore, we choose an approach, where both robots stay close to each other, to complement each other at science investigations.

Our leader-follower coordination is solely based on the exchange of the current goal location of each robot. The leader, as well as the follower, compute their own local exploration goals. The goal system of the leader is similar to the goal system for a single robot exploring an ROI, as presented in use case 1 (see fig. 4.4).

The goal system of the follower robot, presented in fig. 4.15, includes the additional sub-objective *follow leader*. To follow the leader, we introduce two new criteria, multi-robot distance  $c_D$  and multi-robot alignment  $c_A$ , which we detail in the next section.

##### 4.5.1 Multi-robot Alignment Criterion

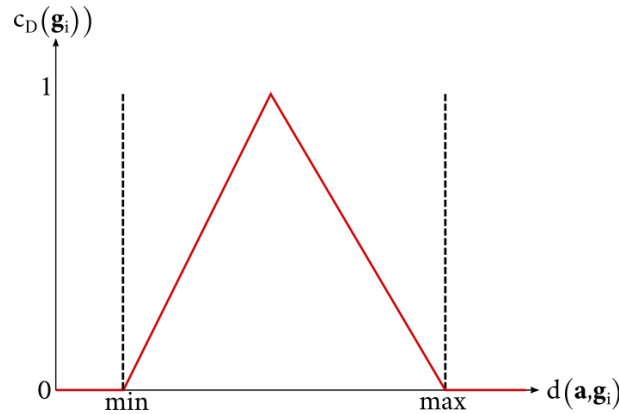
The multi-robot alignment criterion  $c_A$  implements a ‘following’ behavior. To move in the same direction as the leader robot, the follower robot receives the current position and the next exploration goal location of the leader. The multi-robot alignment criterion  $c_A$  describes the difference between the movement direction  $\mathbf{v}_l$  of the leader and the directional vector  $\mathbf{v}_f$  of the current position of the follower robot and a potential goal location  $\mathbf{g}_f$  of the follower.

$$c_A(\mathbf{g}) = \arccos\left(\frac{\mathbf{v}_l \cdot \mathbf{v}_f}{|\mathbf{v}_l| |\mathbf{v}_f|}\right) \quad (4.15)$$

### 4.5.2 Multi-robot Distance Criterion

In order to maximize the new information gained and to ensure that the robots don't interfere with each other, we apply the multi-robot distance criterion  $c_D$ , which evaluates the distance between the leader and follower robot. For this, the operator has to define a minimal, maximal, and optimal distance between the goal locations of the two robots. With the function  $f_D()$ , shown in figure 4.16, we transfer the distance between the goal location of the leader  $\mathbf{g}_l$  and a potential location of the follower  $\mathbf{g}_f$  to the multi-robot distance criterion  $c_D(\mathbf{g})$ .

$$c_D(\mathbf{g}) = f_D(d(g_f, g_l)) \quad (4.16)$$



**Figure 4.16:** To transfer the distance between the leader and the follower robot a function is set up by the defining the minimal, maximal and optimal distance between them. Distances below the minimal and above the maximal distance get a score of 0 and the optimal distance a score of 1. In between we use a linear function to rate the distance between the robots.

## 4.6 Criteria Classification

To conduct the four use cases, we developed several criteria, to model the required exploration behavior, section 4.2-section 4.5. Each goal location has to be evaluated for each criterion. For example, for use case 3 - autonomous search, five criteria have to be evaluated. This evaluation, is a CPU-intensive task, for example, especially performing the ray cast to calculate the information gain from the 3D structure of the environment requires a large amount of resources. As CPU resources on space robots are often very limited, we suggested in our publication, Lehner et al. [11], a simple, but very effective criteria classification. The idea is, that it is not necessary to recalculate, for example, the information gain, for each goal at each exploration action. The information gain, at a goal only changes when the map at the goal location is modified. This is the case when the area is sensed by the robot or the map is optimized by a loop closure event. Both cases, rarely happen when exploring large regions. We found that the exploration

criteria can be classified in three general classes, which reflect how often a criterion has to be recomputed for a goal location.

- **Robot-dependent:** A criterion value changes when the robot moves through the environment. Thus, robot-dependent criteria have to be recomputed after each robot movement, i.e. at each exploration step. The following criteria applied in this thesis are robot-dependent:
  1. cost  $c_{cost}$
  2. direction of interest  $c_{doi}$
  3. loop closure impact  $c_i$
  4. multi-robot alignment  $c_A$
  5. multi-robot distance  $c_D$
- **Map-dependent:** A criterion value only has to be recomputed, when the map at the goal location is modified. At each exploration action, the robot only senses a small part of the environment, thus most parts of the map do not change. After a first estimation of a map-dependent criteria, the values have to be re-computed only for a few goal locations at each exploration actions. The following criteria applied in this thesis are map-dependent:
  1. information gain  $c_{IG}$
  2. loop closure likelihood  $c_{ll}$
- **Environment-dependent:** A criterion value, for a goal location never has to be re-computed, assuming the environment is not changing. For example, the scientific interest of a goal location remains the same unless the mission objectives change. Nevertheless, we relax the assumption that the environment-dependent criteria values never change for a goal location, by allowing an update of the criteria values on certain events, e.g. on the request of the operator. In this thesis, we present the feature of interest criterion  $c_{foi}$ , which we classify as an environment-dependent criterion. However, we describe the criterion with a polar descriptor and finally evaluate the scientific interest in the direction between the current robot position and a goal location. On the one hand, the classification of the environment and the description with the polar descriptor can be classified as environment-dependent. On the other hand, the polar descriptor is computed relative to the robot center and thus, the orientation of the polar descriptor has to be updated with the robot movement.

## 4.7 Dissucssion and Summary

In this chapter, we presented the realization of four relevant use cases for robotic planetary exploration missions. For each use case, we modelled an exploration behavior, utilizing our general concept for robotic exploration based on MCDM. In the following,

we discuss the exploration behaviors for each use case with respect to the related work, stated in section 2.2.

- **Use case 1 - integrated exploration for an autonomous exploration:** We presented an integrated exploration strategy, which applies active loop closing for re-localization during the exploration process, similar to [36, 41, 35]. For re-localization we, additionally to exploration goals, sample revisiting goals. These revisiting goals are placed at existing submaps, considering the loop closure generation approach of our underlying 3D global mapping system. Instead of solely, evaluating the likelihood of a loop closure event, when visiting a revisiting goal, e.g. as [1, 41, 36], we introduce our novel criterion loop closure impact, which evaluates the impact of a potential loop closure on the graph optimization. This is important, as revisiting means interrupting the exploration process, consequently, the impact of such an event is important.
- **Use case 2 - directed exploration to set up a drive-by science mission:** We model a directed exploration behavior, which allows applying robotic exploration in the fashion of drive-by science. Our main goal is to direct the robot towards a predefined POI. As we apply an exploration approach, we can apply the information gain criterion  $c_{IG}$  in order to increase the information gain, while moving towards the POI. However, compared to the drive-by science approaches utilizing informative path planning [53, 54], we are not able to maximize the information gain during the traverse of the environment. We require no a-priori knowledge of the environment, can react quickly to the changing structure in the environment and have low computational costs compared to methods, which apply continuous re-planning [55, 52, 56]. To allow for science investigation on the way, we could additionally apply our feature of interest criterion  $c_{foi}$ , we introduced for use case 3.
- **Use case 3 - informed exploration to search for a FOI:** We consider the object search problem, as an exploration problem and model an informed exploration behavior to find a FOI in an unknown ROI. Our strategy can be classified as an informed search method with respect to the related work (see section 2.2.3). To detect a FOI we equip the robot with an object detector. We present preliminary results on detecting and distinguishing different types of rocks, with a Scientific Camera with several multi-spectral filters. At each exploration step, we perform a camera swipe and try to detect a FOI. The detection results are stored in a polar histogram, describing the probability of finding a FOI in a certain direction. Unlike, Shubina and Tsotsos [76], who save the probability of finding an object in a probabilistic map over the whole area, we decided against a global representation. Our reactive approach requires less memory. For space exploration missions it is not possible to apply enhanced reasoning, by considering object co-occurrences, as for example Aydemir et al. [72] utilize.

- Use case 4 - leader follower multi-robot coordination:** We coordinate two heterogeneous robots co-exploring a ROI. The coordination is solely implemented by evaluating two additional criteria for the goal locations of the follower robot. Our approach is best compared with the existing reactive behavior based approaches [87, 100, 89, 90], although our method is not reactive. However, we similarly apply careful decision making at each exploration step. Similar to [92], we consider that our robots have different capabilities and have to collaborate for potential science investigations. In our leader-follower coordination strategy, the leader explores freely the ROI by applying the integrated exploration strategy presented for use case 1. The follower also applies the integrated exploration strategy but additionally is requested to hold an optimal distance to the leader and to move in the same direction as the leader. To achieve this coordination, the robots exchange the exploration goals and the follower evaluates the criteria multi-robot alignment  $c_A$  and multi-robot distance  $c_D$ . Other, than [92], our follower robot also explores the environment and thus contributes to the map of the environment. Behavior based coordination strategies, where found to work only in simple environments [87]. Although, planetary surfaces induce several challenges, the structure of the environment is compared to an indoor environment simpler.

Criterion	Description	Class	Use case
cost $c_{cost}$	Cost to pay to reach a goal	robot-dependent	Use case 1-4
information gain $c_{IG}$	New information that can be gained at a goal location	map-dependent	Use case 1-4
loop closure likelihood $c_{ll}$	Likelihood that a loop closure event is successful	map-dependent	Use case 1,3,4
loop closure impact $c_{li}$	Impact of the loop closure constraint on the map optimization	robot-dependent	Use case 1,3,4
direction of interest $c_{doi}$	Directs the robot into a predefined direction	robot-dependent	Use case 2
feature of interest $c_{foi}$	Probability that a FOI lies in the direction to a goal location	environment-dependent	Use case 3

**Table 4.1:** Table summarizing the identified exploration criteria for a robotic space exploration mission.



To model the exploration behaviors to conduct our use cases, we applied several criteria, summarized in table 4.1. As the evaluation of the criteria is computationally expensive, e.g. especially for the information gain, we introduced a novel criteria classification, which states, how often a criterion has to be recomputed for a goal location. Whereas robot-dependent criteria have to be recomputed at each exploration step for all goal locations, map-dependent criteria have to be recomputed occasionally on map change and environment-dependent criteria never have to be re-evaluated.

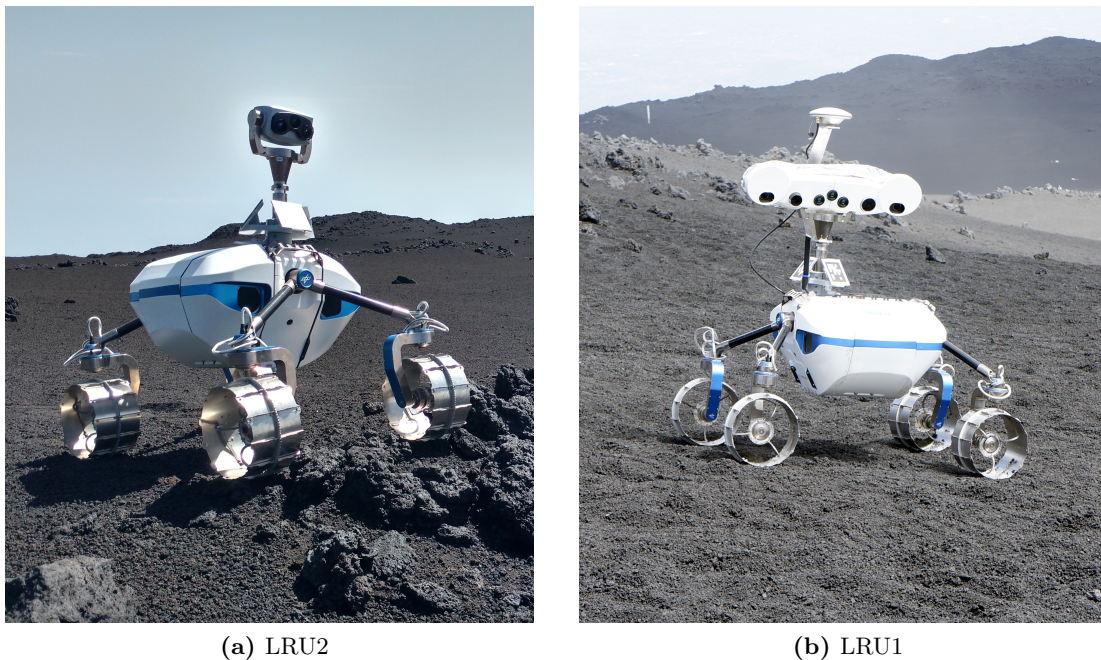


## 5 Implementation & Integration

In this chapter we describe, the implementation of our general exploration approach onboard a space rover prototype. Further, we introduce the robot hardware platform and three different simulation environments, we applied for our experiments.

### 5.1 Robot Hardware

We demonstrate our general exploration concept and its application by conducting real-world experiments with the two lightweight rover units, LRU and LRU2, shown in fig. 5.1.



**Figure 5.1:** Real space rover prototypes, LRU2 (left) and LRU (right). Both rovers have four individually controlled and powered wheels. The Pan/tilt camera head consists of two cameras in a stereo setup and a third RGB camera. LRU is additionally equipped with scientific cameras. LRU2 has a manipulator and a carrier on its back.

Both rovers have four individually controlled and powered wheels, which allows navigating through the rough terrain of planetary surfaces. Each rover is equipped with a

## 5 Implementation & Integration

stereo camera system, with a baseline of 9cm and a Xsens MTi-10 IMU. Dense stereo matching is performed on a Spartan 6 LX75 Field Programmable Gate Array (FPGA). All other computations are done on an Intel Core i7-3749QM CPU (2.70 GHz).

LRU is equipped with a scientific camera consisting of two wide-angle cameras (LWAC, RWAC) in a stereo setup with spectral filter wheels, one narrow-angle camera, and one thermal camera. The filter wheels for the LWAC and RWAC are each comprised of three color filters (100 nm bandwidth) and six narrow band (10nm bandwidth) geology filters (*Geo1-Geo12*), across a range of 440 – 660 nm for LWAC and 720 – 1000 nm range for RWAC. LRU2 is equipped with a robotic arm, the jaco2 manipulator of Kinova, with a special docking interface developed at Deutsches Zentrum für Luft- und Raumfahrt (DLR). The docking interface allows attaching different tools, e.g. a robotic hand to grab stones or a shovel to take a sand probe. Further, LRU2 can carry payload boxes on its back, which can hold scientific instruments or can serve as sample return box.

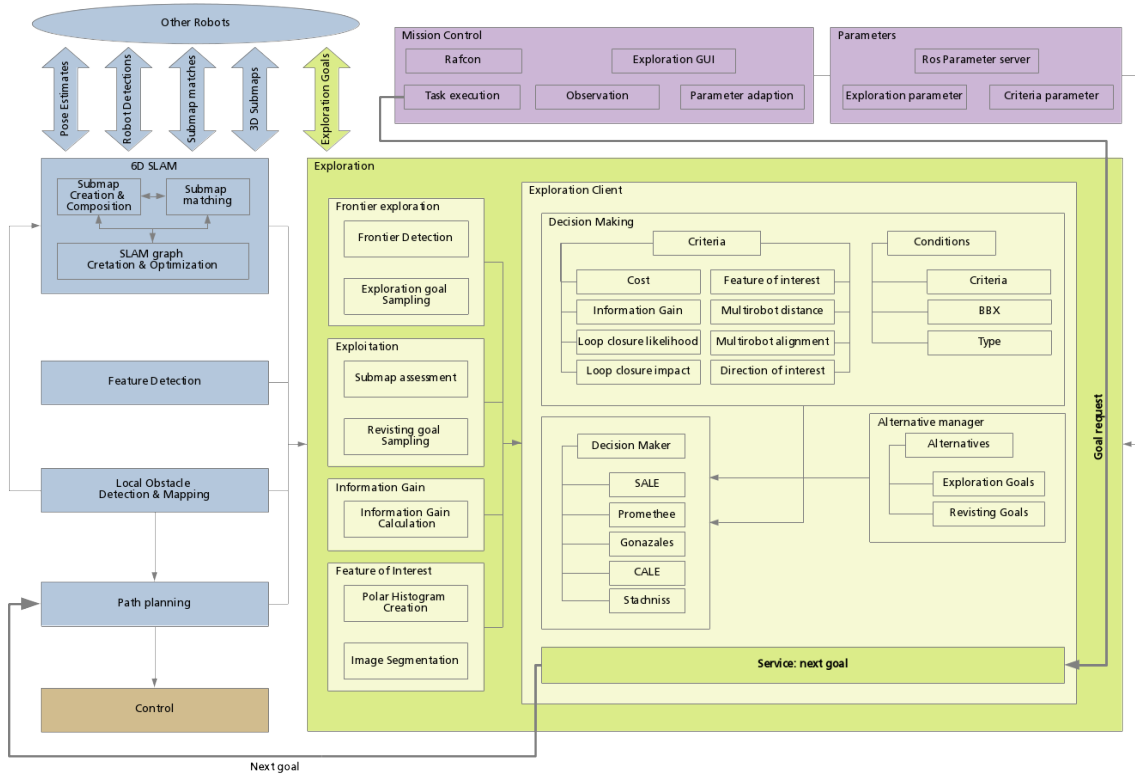
### 5.2 Exploration Framework

In fig. 5.2 we show an overview of our software architecture. The exploration module depends on a 6D Localization and mapping [8], a local obstacle detection and mapping module [97], a feature detection module, as well as on a path planning module. For details on the global mapping components, please refer to section 2.1.3.

The exploration module consists of five separate nodes, which communicate by applying the Robot Operating System (ROS). To decide, *where to move next?*, the exploration client implements a service to request the next best goal. The frontier exploration node, implements our frontier based goal sampling method for exploration goals. The exploitation node implements the sampling of re-localization goals at the existing submaps. The exploration client node, stores the exploration goals and re-localization goals in the so called alternative manager, which also handles the update of goal locations. To evaluate the goal locations several criteria and conditions are implemented. Each criterion or condition is derived from the corresponding parent classes. Criteria like the information gain and feature of interest require complex and are implemented in separate nodes. The computed criteria values can be requested via service calls. To rank the goals our framework implements several decision maker, the outranking method PROMETHEE II described in section 3.3.2 and the multi attribute utility functions SALE and CALE detailed in section 3.3.1, as well as the state of the art utility functions introduced by Gonzales and Latombe [2] and Stachniss et al. [1] and the utility function based on the nearest frontier method introduced by Yamauchi et al. [4].

To start and observe an exploration mission, we apply the mission planning software RAFCON [102], by which tasks are described in hierarchical state machines.

To define an exploration behavior to conduct an exploration task, similar to our use cases, the parameters defining the criteria are defined in a configuration file. We provide a GUI to observe the exploration process and adapt the criteria parameters if necessary.



**Figure 5.2:** Overview of our exploration framework. In this thesis we investigated and developed the exploration components (green), we additionally utilized the mission control software RAFCON to operate and start our robotic exploration. Our exploration component is based on the mapping and navigation pipeline (blue) introduced by Brand et al. [20] and Schuster et al. [101, 8]. To explore, repeatedly a goal request is sent to a service, which computes the next best goal location.

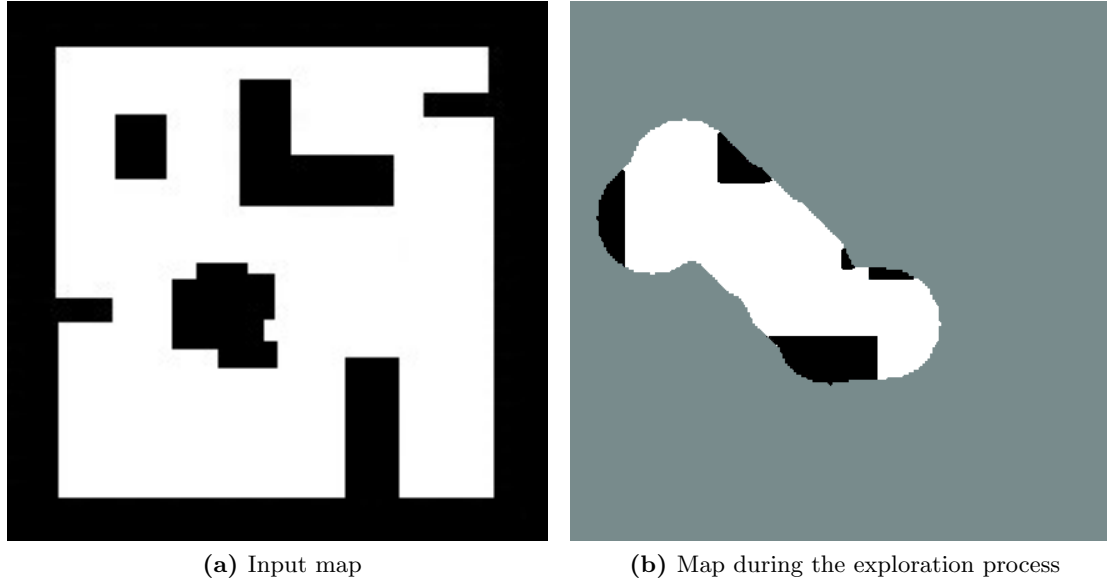
## 5.3 Robot Simulation

To develop and test our methods for robotic planetary exploration, we applied one simple 2D simulation and two different high-fidelity Software in the Loop (SiL).

### Simple exploration simulation

To test and evaluate our multi-robot coordination approach (see section 4.5), we implemented a simple 2D simulator. A black and white image, as illustrated in fig. 5.5, whereby black represents obstacles and white represents free space, are used as environment representation. The map, is revealed when the robot moves through the environment, in a circle around the robot. As we do not simulate the real robot sensors, only the distance criterion  $c_{cost}$ , the direction of interest criterion  $c_{doi}$ , multi-robot criteria  $c_A$ ,  $c_D$  can be computed as on the system. In our simple simulation we approximate the

information gain, by counting the number of unknown grid cells around a goal location.



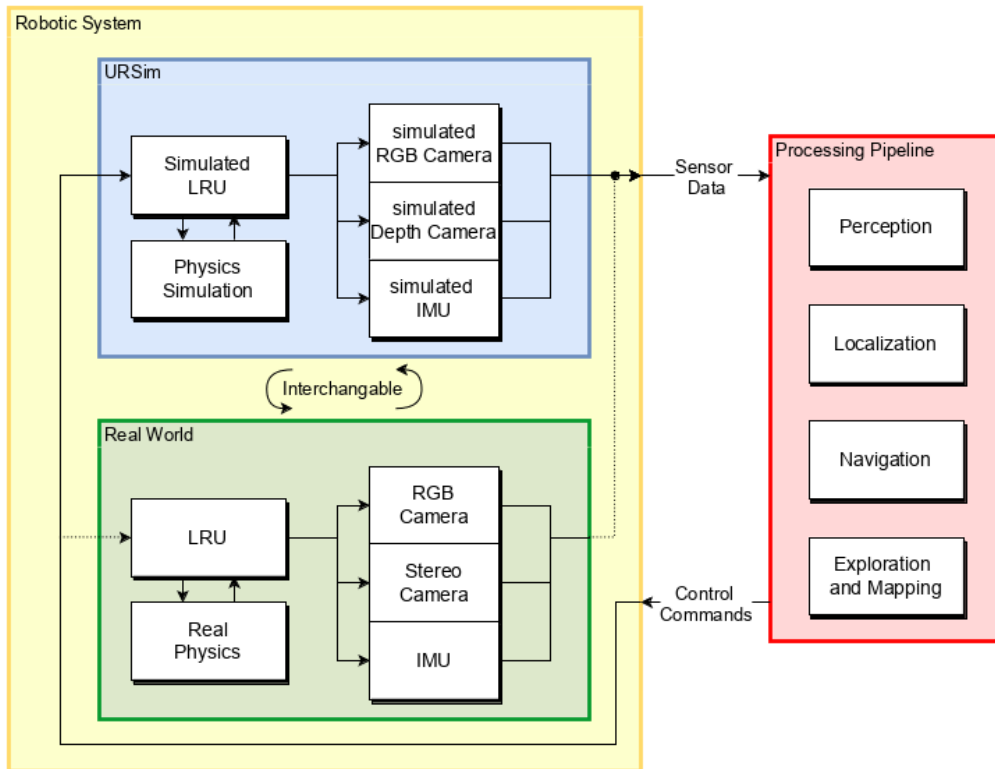
**Figure 5.3:** Example of a 2D input map of our simple 2D simulator for testing the coordination of multiple robots. Black cells illustrate obstacles, white cells known space and the green-grey cells represent unknown space. On the left, we show the input map, which is consecutively revealed during exploration, as shown on the right.

### Software in the loop Simulator

To test our exploration methods together with the complete navigation and mapping pipeline in conditions close to a real mission we apply the two high-fidelity SiL simulators Rover Simulation Toolkit [103] and URSim [9]. With both simulators, we simulate the complete sensor setup of LRU and provide similar interfaces to the real robotic system, as highlighted in fig. 5.4. This allows us to run our complete mapping, navigation, and exploration stack as on the real system.

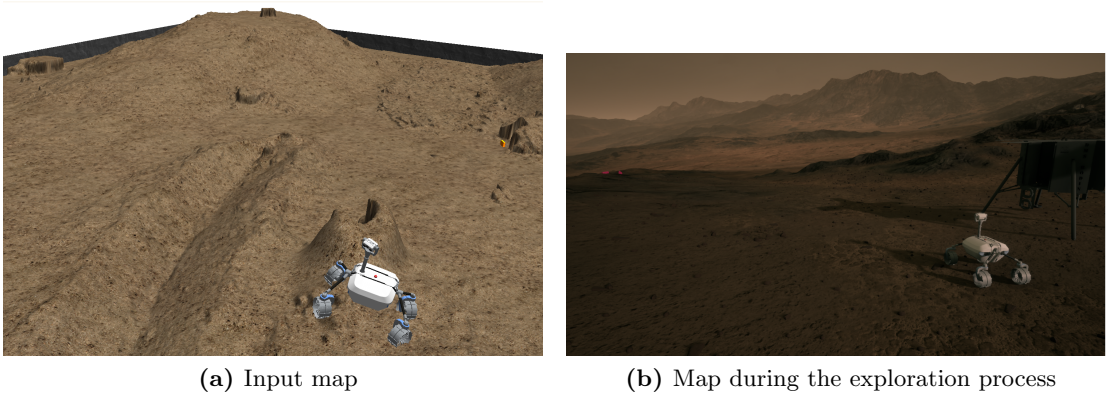
The Rover Simulation Toolkit introduced by Hellerer et al. [103], is especially developed for the simulation of planetary rovers and written in Modelica. To simulate a rover, an interactive 3D visualization applying the DLR visualization library [104] is created. To test the navigation, mapping and exploration pipeline of the LRU, Schuster et al. [105] extend the simulation by integrating the vision sensors and providing the renderings of these with a predefined frame rate. The main focus of the simulation is on the simulation of the locomotion subsystems of a rover, in order to support the design of new planetary rovers. However, the simulator does not provide a photo-realistic rendering of the environment as shown in fig. 5.5a, which is of importance for testing vision based navigation pipelines.

In our publication, Sewtz et al. [9], we present URSim, a complete Software-in-the-Loop (SiL) simulator. It is based on the Unreal Engine 4 (UE4), which offers physics support and state-of-the-art photo-realistic rendering. Different robotic systems can be integrated through our generic robot class. We provide several visual and physical sensors, which can be easily attached to a robot as invisible robot components. URSim is customizable through the modern and adaptable system architecture. This allows the integration of different interprocess communication frameworks. Currently, the Robot Operating System (ROS) and links and nodes are provided. In fig. 5.5b, we show our LRU in a simulated Martian environment, next to a lander.



**Figure 5.4:** The real LRU and the two high-fidelity simulators URSim and the Rover Simulation Toolkit are interchangeable. The SiL simulators provide the same interfaces and inter process communication as applied on the real system. This allows us to run the complete perception and navigation and mapping pipeline as on the system with the simulators. Figure adapted from Sewtz et al. [9]

## 5 Implementation & Integration

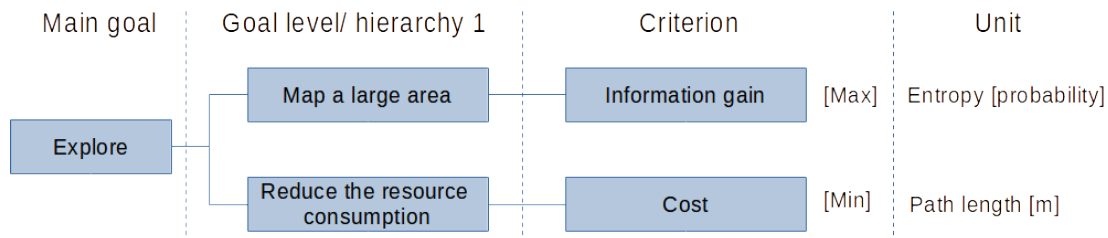


**Figure 5.5:** (a) Simulated outdoor environment employing the simulation based on the Rover Simulation Toolkit [103]. (b) Photo-realistic rendering of the LRU in a Martian landscape in URSim.



## 6 Experimental Evaluation

In this chapter, we present an evaluation of our contributions to methods for planetary robotic exploration. In a series of experiments, conducted on the robotic system and applying the exploration framework described in section 5.1, we demonstrate the four use cases presented in this thesis, in either simulation or real-world experiments. With our evaluations, we aim to cover the mission, environmental and system challenges for planetary robotic exploration we identified in section 1.3. First, we present the evaluations for use case 1, which applies an integrated exploration (section 6.1). Second, we demonstrate use case 2, by setting up a drive-by science mission applying our directed exploration approach (section 6.2). Third, we apply our informed exploration to find a feature of interest as described in use case 3 and compare the time until the feature of interest is found with a greedy exploration method (section 6.3). Fourth, we extend our experiments to our multi-robot setup with two heterogeneous robots and analyze the coordination behavior induced by our leader-follower method (section 6.4). In addition, we analyze the processing and runtime of our general MCDM based exploration concept (section 6.6). We conclude our experimental evaluation with a summary and a discussion of our findings regarding the challenges of robotic exploration (section 6.7).



**Figure 6.1:** Criteria hierarchy applied for the FE method, which we use for comparison. To measure the achievement of the two exploration sub-goals, the information gain criterion  $c_{IG}$  and the cost criterion  $c_{cost}$  are applied.

In all our experiments, the environment is completely unknown to the robot and the robot acts completely autonomously after the mission started. All computations are done online on the system.

For convenience, we abbreviate the exploration behaviors applied for the single robot use cases 1-3 in this chapter. We abbreviate the integrated exploration applied for use case 1 with integrated exploration (IE), the directed exploration applied for use case 2 with directed exploration (DE), and the informed exploration applied for use case 3 with informed exploration (IFE). For comparison, we performed several experiments with the nearest frontier exploration approach, abbreviated with FE, evaluating the cost and

## 6 Experimental Evaluation

information gain criterion, as depicted in the criteria hierarchy in fig. 6.1, and applying the utility function described by Stachniss [1], stated in section 2.1.2 for decision making.

The preference functions introduced with PROMETHEE II mostly depend on criterion characteristics and are to the greatest extent independent of the experimental setup. As the environment for all experiments, applying PROMETHEE II, represents a large unstructured outdoor environment, the preference function, stated in table 6.1, are similar for all single robot experiments. Only the preference function applied for the multi-robot experiments differ between simulation and real-world experiments and are stated in the corresponding section 6.4. The values are defined empirically and by analyzing the minimum and maximum values of the criteria.

Criterion	Preference function	
	Type	Parameter
cost $c_{cost}$	linear	$q = 5, r = 30$
information gain $c_{IG}$	linear	$q = 0.1, r = 0.8$
loop closure likelihood $c_{ll}$	linear	$q = 0.2, r = 0.8$
loop closure impact $c_{li}$	linear	$q = 0.2, r = 0.8$
direction of interest $c_{doi}$	Gaussian	$\sigma = 0.6$
feature of interest $c_{foi}$	linear	$q = 0.2, r = 0.8$

**Table 6.1:** Preference functions used in our simulation and real-world experiments

### 6.1 Use Case 1 - Integrated Exploration

For use case 1, we evaluate the integrated exploration (IE) with active loop closing, which we introduced in section 4.2. We perform experiments with IE applying our novel multi attribute utility function CALE and SALE to assess their exploration, mapping, and localization performance compared with FE. We published the content and the results of this section first in our conference paper [10].

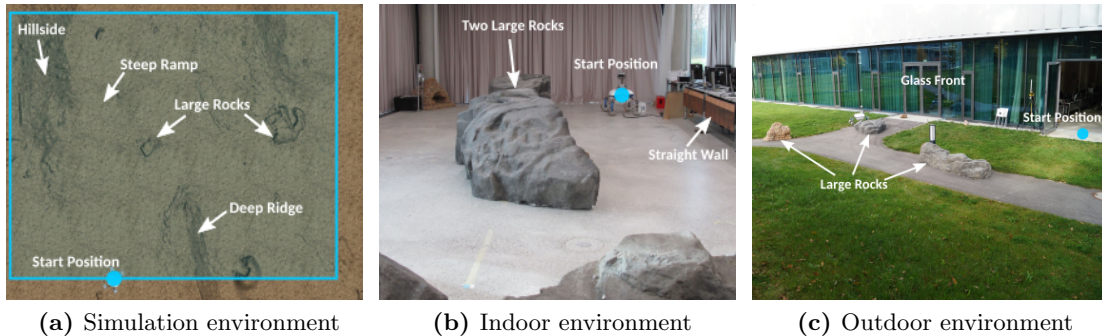
#### Experimental Setup

We conducted experiments in simulation with the Rover Simulation Toolkit[103] ( section 5.3) and in two different real-world scenarios:

- *Simulation:* unstructured rough outdoor environment [106] containing a steep ramp, a deep ridge, and several rocks. The ROI, visualized in the top down view displayed in fig. 6.2a, has a size of 10 m  $\times$  15 m.

## 6.1 Use Case 1 - Integrated Exploration

- *Indoor*: laboratory environment with two large rocks of approximately  $11\text{ m} \times 6\text{ m}$  size displayed in fig. 6.2b. A ceiling-mounted *Vicon* tracking system <sup>1</sup> with 14 *Bonita* cameras provides ground truth pose data.
- *Outdoor*: test area outside the laboratory, containing several large boulders and other human structured obstacles with a challenging reflective glass front presented in fig. 6.2c. The exploration area has a size of  $15\text{ m} \times 20\text{ m}$ .



**Figure 6.2:** Experiment environments for use case 1: The start position is marked with a blue circle. (a) Simulation environment with a ROI (blue rectangle) containing a steep ramp, hillside, large rocks and a deep ridge. (b) Indoor environment with two large rocks and a straight wall. (c) Outdoor environment with large rocks and the large glass front of our mobile robotics laboratory.

The task of the robot is for all experiment scenarios, to explore at least 95% of the defined ROI, before returning to its start position. For this, we did not impose a time limit. However, we aborted the exploration thrice manually, as the robot was not able to navigate autonomously through the environment due to a poor localization accuracy. This happened twice when applying the FE method and once while performing the outdoor experiments. In simulation, we performed ten, indoor five and outdoor one experiment, in each case, with all three methods.

Table 6.2 lists the parameters we used for the experiments. The high distance weight  $w_{cost}$  inhibits the robot from driving long distances during the exploration. The threshold  $\iota$  is higher for CALE, than for SALE, as CALE is always attracted to try for loop closures, therefore usually prevents a poor localization accuracy prior to SALE. The indoor environment is limited by our tracking system and much smaller than the simulation and outdoor environment. Only small local pose estimation error are accumulated during the exploration and hence, we used a lower covariance threshold  $\iota$  for active loop closing.

To prevent the robot from getting stuck in trying to improve its localization, we allow only  $n$  consecutive loop closing actions, i.e. moving to a revisiting goal, using CALE as well as SALE. Although we have a high success rate when actively creating loop closures,

<sup>1</sup><https://www.vicon.com/software/tracker/>

## 6 Experimental Evaluation

	Simulation			Real-world Experiments					
	Experiments			Indoor			Outdoor		
	$\iota$	$\beta$	$w_{cost}$	$\iota$	$\beta$	$w_{cost}$	$\iota$	$\beta$	$w_{cost}$
<b>FE</b>	-	-	0.7	-	-	0.7	-	-	0.7
<b>CALE</b>	0.5	0.5	0.7	0.25	0.3	0.7	0.6	0.5	0.7
<b>SALE</b>	0.3	-	0.7	0.15	-	0.7	0.4	-	0.7

**Table 6.2:** Parameters used in our simulation and real-world experiments to demonstrate use case 1. With  $\iota$  being the threshold, which balances the exploration and re-localization process for CALE and SALE,  $\beta$  being a weight for the loop closure likelihood  $c_{li}$  and loop closure impact criterion  $c_{li}$  and  $w_{cost}$  being the weight for the cost criterion  $c_{cost}$  (for details see section 3.3.1).

we set  $n = 2$ . As the matching is running online but as a slower background process, the effect of loop closures does not necessarily occur immediately.

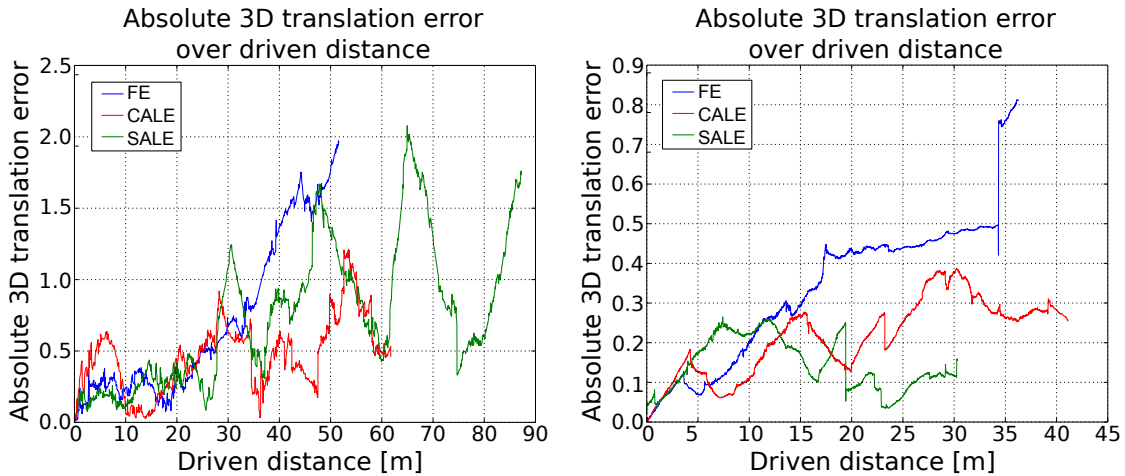
### Results

For each method, we present and discuss the results for their exploration, localization and mapping performance. We report the 3D localization error, the mean driven distance and the number of loop closures, for our simulation and real-world indoor experiments in table 6.3. In the simulation experiments, a mean absolute error of 0.68 m and 0.69 m with a maximum error of 1.28 m and 1.32 m occurred applying CALE and SALE. Applying FE the mean absolute error is with 1.02 m significantly higher compared to CALE and SALE. Further, the plot presented in fig. 6.3 shows a continuously increasing error using FE. In contrast the plot displays several error drops using CALE and SALE, which results from a global optimization of the SLAM graph after a loop closure was detected. The poor localization accuracy with FE even lead twice to an abortion of the exploration, as the robot was not able to navigate autonomously through the environment. To improve the localization accuracy a loop closure is required. Using CALE and SALE twice as many loop closures as with FE are recorded (table 6.3) and thus a better localization performance could be achieved.

However, the exploration efficiency using CALE and SALE is worse than applying FE. To explore the ROI of the simulation environment, the robot had to drive approximately 73.39m with CALE, 81.80m with SALE and 52.01m with FE (table 6.3). In the smaller indoor setup SALE and FE achieve a similar driven distance, while the mean driven distance applying CALE is significantly higher. SALE only rarely directs the robot towards already visited areas in the indoor environment, as the local filter estimates already achieve a high localization accuracy. In fig. 6.4 the higher number of actions conducted with CALE and SALE confirm the increased effort indicated by the driven distance. About 6-8 explore actions are required with each method to explore the environment. CALE and SALE additionally conduct 2-5 re-localization actions.

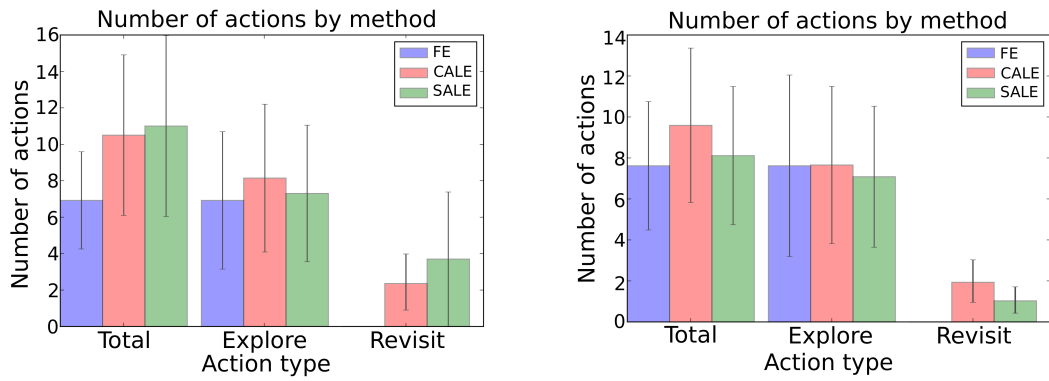
<b>Simulation Experiments</b>	<b>FE</b>	<b>CALE</b>	<b>SALE</b>
mean absolute error [m]	1.02	0.68	0.69
mean standard deviation [m]	0.68	0,41	0,51
mean max error [m]	2.43	1.28	1.32
mean driven distance [m]	52.01	73.39	81.80
mean numbers of loop closures	5.86	13.86	10.43
<b>Real-world Indoor Experiments</b>	<b>FE</b>	<b>CALE</b>	<b>SALE</b>
mean absolute error [m]	0.27	0.23	0.22
mean standard deviation [m]	0.12	0.1	0.2
mean maximal error [m]	0.53	0.44	0.42
mean driven distance [m]	39.4	55.65	42.45
mean numbers of loop closures	4.8	22.8	11.2

**Table 6.3:** Comparison of the localization performance in our simulation (average values from 10 test runs per method) and real-world indoor experiments (average values from 5 test runs per method).

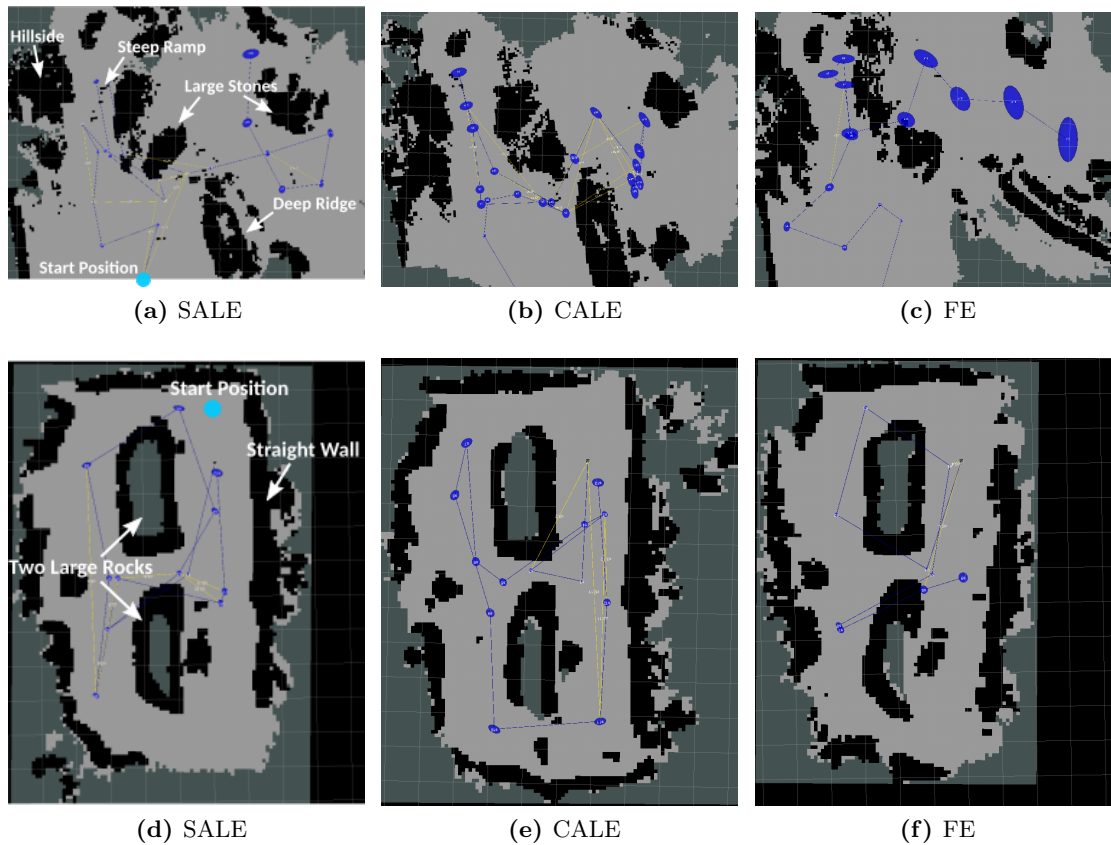


**Figure 6.3:** 3D translational error over the driven distance for two experiments showing the characteristic behavior of the three methods: FE (blue), CALE (red), SALE (green).

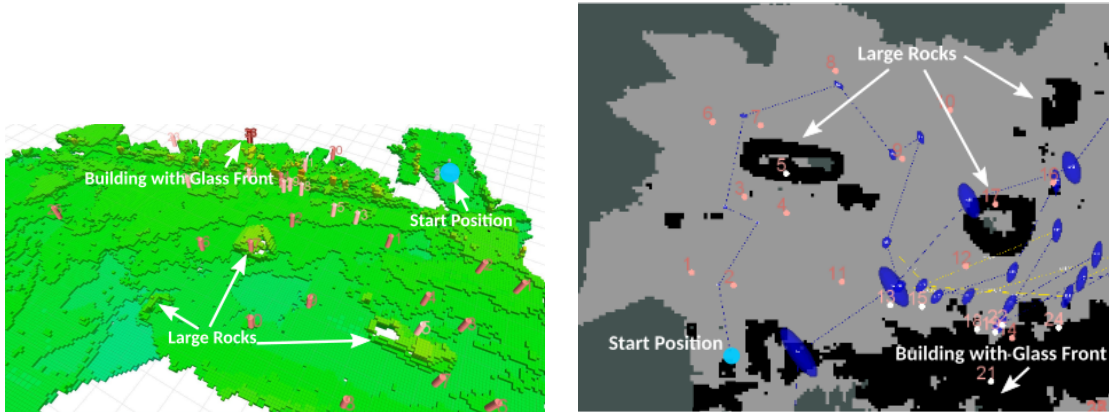
## 6 Experimental Evaluation



**Figure 6.4:** Mean number of actions (with standard deviation) required by each method to fulfill the exploration task.



**Figure 6.5:** Examples of resulting 2D occupancy grid maps and SLAM graph characteristics for each exploration method for the simulation environment (a)-(c) and the indoor environment (d)-(f). The yellow edges of the graph represent loop closure connections. The blue ellipsoids represent the localization uncertainty at each submap.



**Figure 6.6:** Resulting 3D (right) and 2D (left) occupancy grid map and SLAM graph of the outdoor experiment for use case 1.

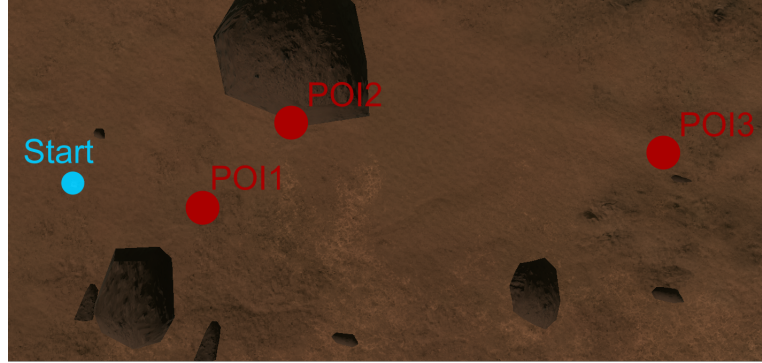
We evaluate the mapping quality by visual inspection of the resulting 2D occupancy grid maps. A statistical quality assessment is difficult, as already small errors in the yaw angle lead to large differences between the ground truth map and the created map. To check if the map is intrinsically correct, we inspect the characteristic elements in the environment such as straight walls and large rocks. In fig. 6.5, we present exemplary maps from our 30 simulated experiments fig. 6.5a-fig. 6.5c and our 15 indoor experiments fig. 6.5d-fig. 6.5f. The 2D occupancy grid maps resulting from the experiments applying CALE and SALE are intrinsically correct for the simulation and the indoor environment. In contrast, the 2D occupancy grid maps created during experiments applying FE, as shown in fig. 6.5c and fig. 6.5f, show a distorted deep ridge in the simulation environment as well as a distorted straight wall for the indoor environment, which indicates a drift of the yaw angle. The error ellipsoids of the SLAM graph are significantly smaller for CALE and SALE and a larger number of loop closure is displayed. To demonstrate our method in a large outdoor environment, we conducted one experiment in the outdoor environment shown in fig. 6.2. In fig. 6.6 we show the resulting 3D voxel grid-map and the 2D occupancy grid-map with the SLAM graph (blue lines and ellipses) and the visited goal locations (rose pins). The highly reflective glass front of the building caused errors in the stereo matching and visual odometry, leading to visible noise in our localization and map estimates.

## 6.2 Use Case 2 - Drive-by Science

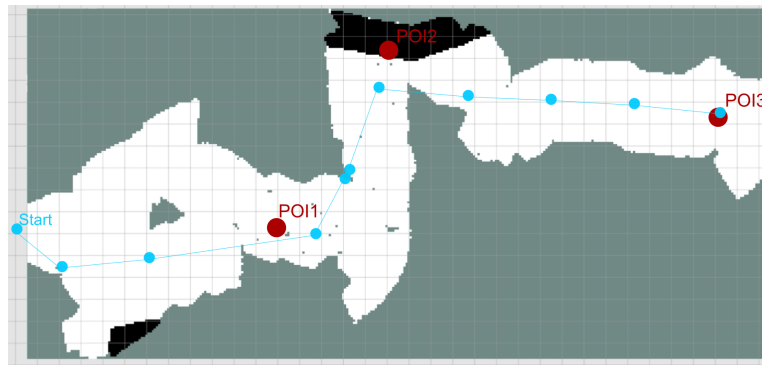
In this section, we demonstrate use case 2 by applying our directed exploration (DE) to set up a drive-by Science mission as described in section 4.3. We show how robotic exploration can be used to approach a predefined global goal location while achieving a high amount of new information during the travel.

## Experimental Setup

We performed one experiment with our simulation URSim (section 5.3) in a Martian Landscape. Fig. 6.7 shows the Martian Landscape in URSim from a top down view with three POIs and the start position of the robot. The task of the robot is to visit



(a) scenario in simulation

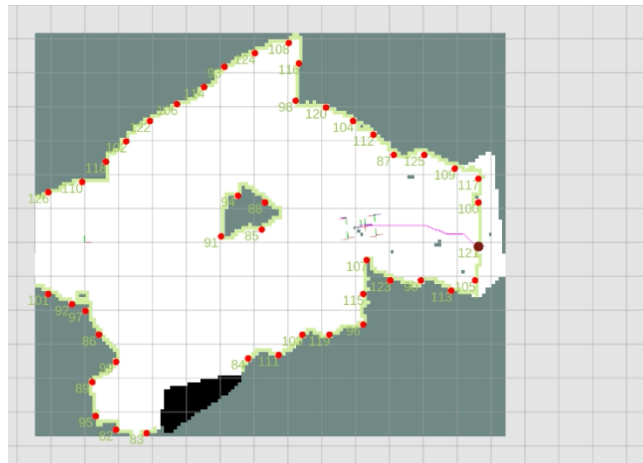


(b) final 2d occupancy gridmap

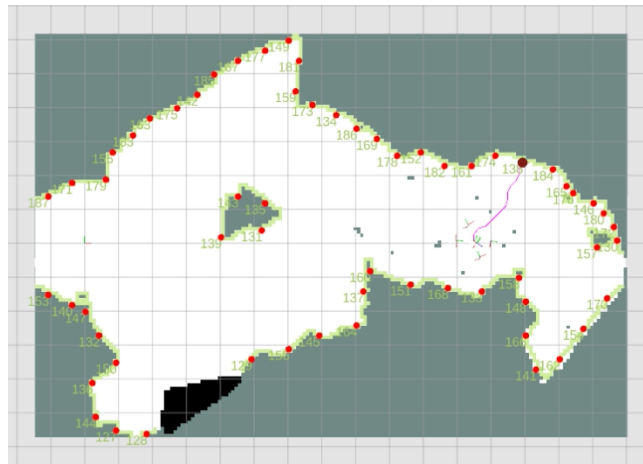
**Figure 6.7:** (a) Topdown view of the martian landscape level with marked POIs the robot has to visit in the given order. (b) Topdown view on the final 2D occupancy grid-map with the trajectory the robot followed (blue).

all POIs in the given order. A POI is reached if the robot is within the vicinity of 2m of the POI. In order to force the robot to move in the direction of the next POI, we apply the direction of interest criterion. The direction of interest always points to the current POI and is switched to point to the next POI when reaching the current POI. The weight of the direction of interest criterion automatically increases, while the weights for the cost and information gain criterion decrease, with progressing mission time. This ensures, that the robot is able to visit all POI with its limited resources in a real mission. We start each experiment with the following criteria weights:  $w_{doi} = 0.1$ ,  $w_{cost} = 0.5$ ,  $w_{IG} = 0.4$ .

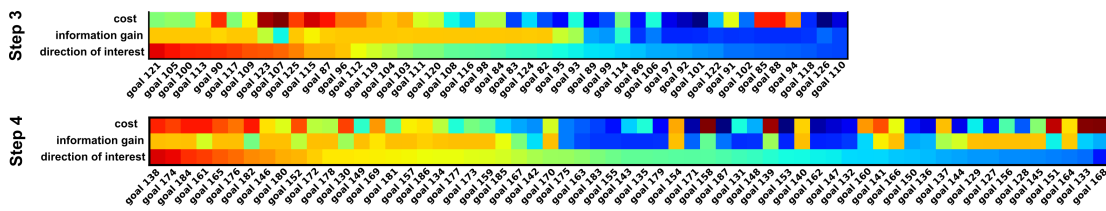




(a) 2D occupancy grid map after exploration action 3



(b) 2D occupancy grid after exploration action 4



(c) Colormaps

**Figure 6.8:** Occupancy grid maps at exploration action three (a) and four(b) with candidate exploration goals (red) and the chosen goal location (dark red). (c) colormaps for the exploration action three and four showing the unicriterion values for the evaluated criteria for each goal location.

### Results

In fig. 6.7b we show the final 2D occupancy grid map with the exploration actions conducted until the last POI is reached. Although the weight on the direction of interest criterion is small, the robot moved directly from the beginning straight towards the first POI. From other experiments with the FE method, we experienced that the robot moves straight towards the unknown space, by only applying the cost and information gain criterion. The direction of interest criterion, regardless of the weight, forces the robot only to move towards the predefined direction. To highlight the effect of the direction of interest criterion, we show in fig. 6.8 the resulting criteria values for all candidate exploration goals for the third and fourth exploration action in a colormap plot, as well as the corresponding 2D occupancy grid maps. The goals plotted in the colormap are sorted by the final multi criteria preference degree. For both actions it is clearly visible that the direction of interest criterion is determining. The goal with the highest multi criteria preference degree has the highest score for the direction of interest criterion.

### 6.3 Use Case 3 - Informed Exploration

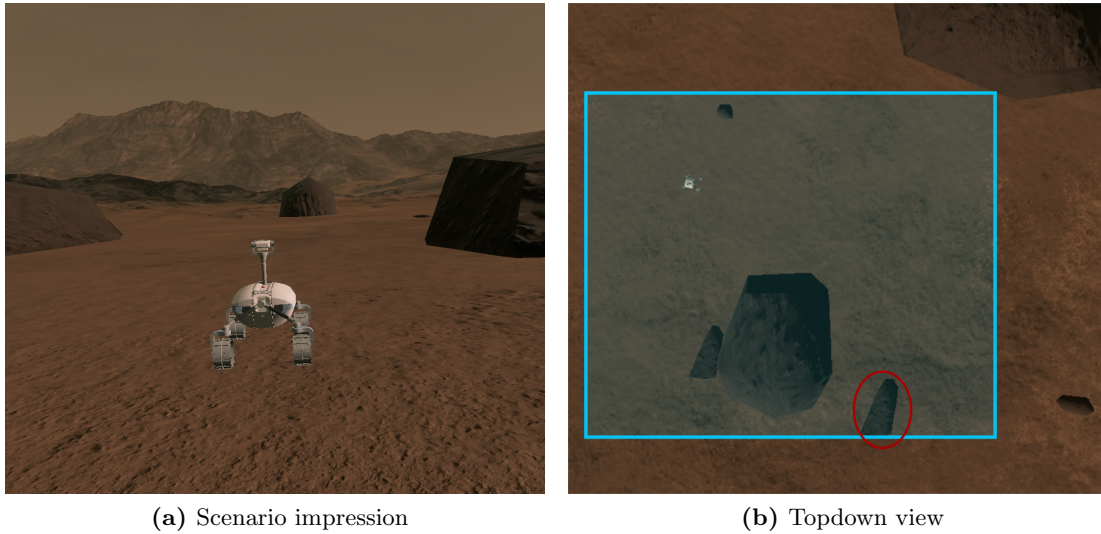
In this section, we evaluate our informed exploration (IFE) introduced in section 4.4, which we apply to demonstrate use case 3. Instead of solely exploring a ROI, as in use case 1, the goal is to find a predefined feature of interest (FOI) in a ROI. With our experiments we aim to show that we can increase the scientific return of a mission by considering search knowledge when deciding where to move next.

#### Experimental setup

We conducted our experiments in the Martian Landscape in URSim. In fig. 6.9 we show an impression of the Martian landscape level of URSim. In the top down view (fig. 6.9b), the ROI with a size of  $304m^2$  ( $19m \times 16m$ ) is highlighted with a blue rectangle. The feature of interest is located at the bottom right corner of the ROI and marked with a red circle.

We assume a scientist defined a certain type of rock as geologic FOI. To be able to identify geologic features of interest, the robot is equipped with an additional semantic camera sensor provided by URSim. In fig. 6.10 we show an example output of the semantic camera sensor and the corresponding color and depth sensor image. The rock of interest is colored blue in the semantically annotated image. The semantic camera sensor simulates the detection output of a real feature detection algorithm, as described in section 4.4.2.

In our experiments, the task of the robot is to find an instance of the geologic FOI in the ROI as fast as possible. The FOI is considered as detected, if the FOI is located within 4m range of the stereo sensor of the robotic system. As described in section 4.4 and depicted in the criteria hierarchy, see fig. 4.9, the subgoals of use case 3, are to find a FOI as fast as possible, as well as mapping a large area while keeping the resource



**Figure 6.9:** URSim Mars Landscape Experiment setup: (a) Impression of the Martian landscape in URSim. (b) Top down view of the landscape used for the directed exploration experiments. The blue rectangle highlights the ROI ( $19\text{ m} \times 16\text{ m}$ ), which contains one feature of interest, marked with a red circle.

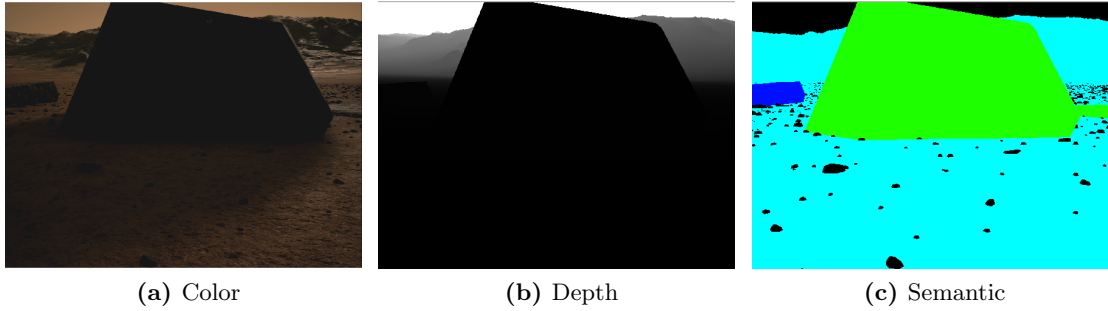
consumption low and maintaining an accurate localization. For our experiments, only ground truth data for the robot’s self-localization could be provided by URSim. With that, we had to skip the subgoal of reducing the localization uncertainty and removed the two criteria, loop closure likelihood, and loop closure impact, from the set of criteria. We conducted three experiments applying IFE with the criteria hierarchy shown in fig. 6.11. For comparison, we further conducted three experiments applying FE, which criteria hierarchy is stated in fig. 6.1.

Method	Criteria weights		
	$c_{cost}$	$c_{IG}$	$c_{interest}$
IFE	0.3	0.1	0.6
FE	0.4	0.6	-

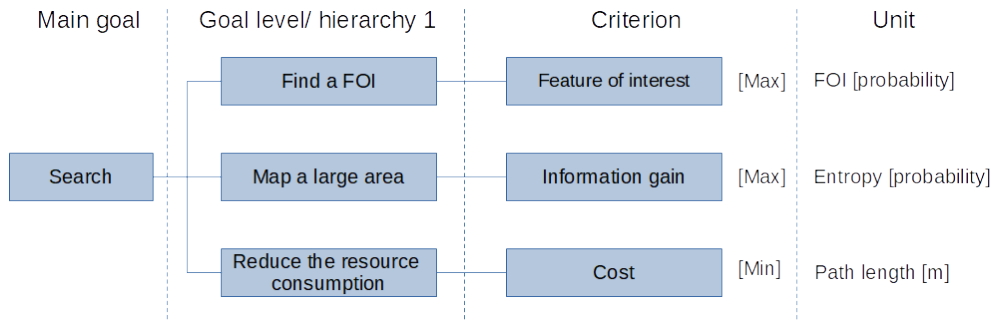
**Table 6.4:** Experiments use case 3: Weights for the criteria applied with IFE and FE in the experiments.

Table 6.4 states the parameters for both IFE and FE, which we applied for the experiments. The parameters were found empirically and defined by a human expert. For both IFE and FE one exploration action comprises, deciding where to move next, moving to the next goal location and performing a 360 sensor swipe at the goal location. However, IFE incorporates the knowledge gained by the sensor swipe with the procedure described in section 4.4.2.

## 6 Experimental Evaluation



**Figure 6.10:** Example output of the virtual visual sensors of URSim.



**Figure 6.11:** Criteria hierarchy of IFE used for the experiments.

## Results

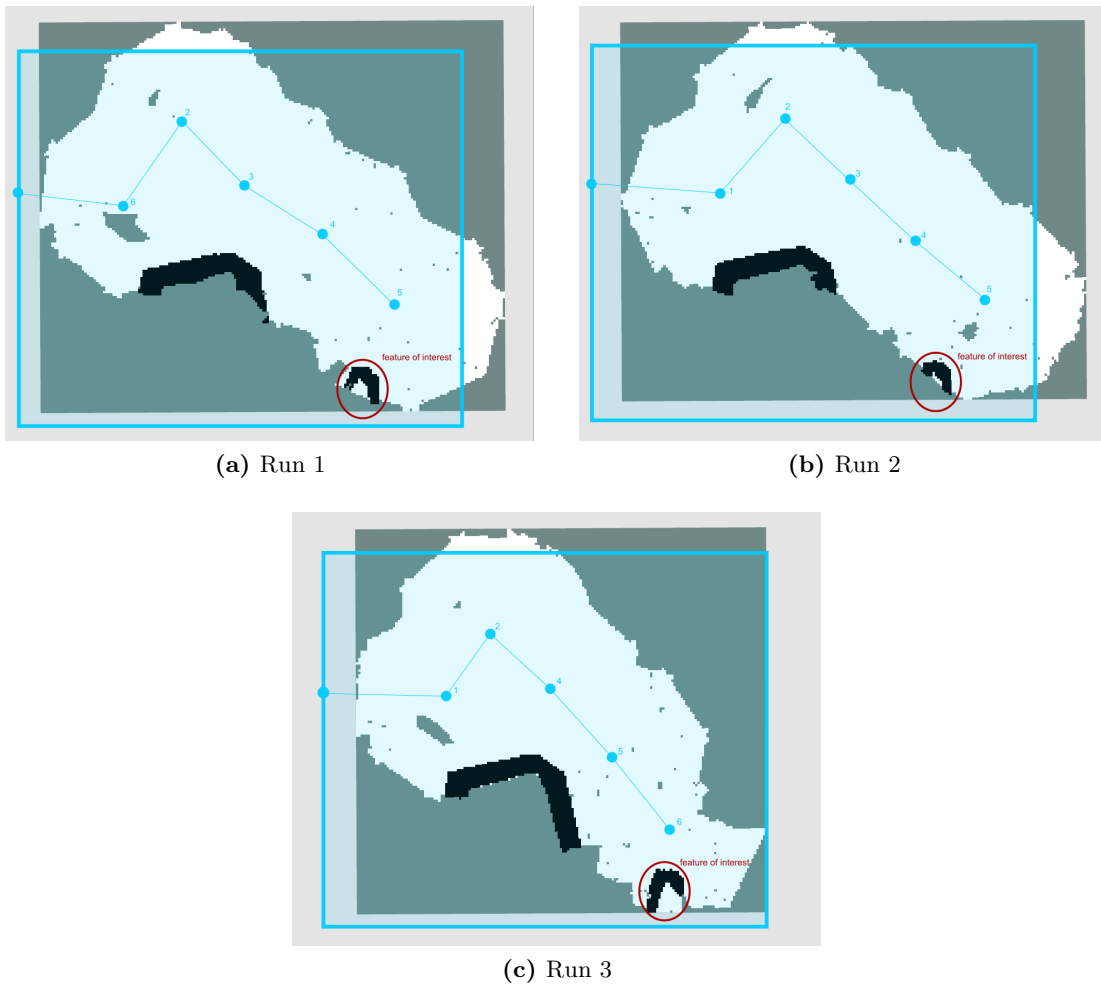
For each method, we present and discuss the results for their search efficiency. In table 6.5 we summarize the results of the search efficiency. Applying IFE the FOI is found after five exploration actions, which took at most 638 sec. Using FE it took up between eight and nine exploration actions and a maximum of 1036 sec to find the FOI. The mean area explored with IFE is  $165m^2$ , whereas the mean explored area applying FE is  $247.7m^2$ . This indicates that the IFE method has a great benefit over a classical exploration when the goal is to find a FOI. The time to find the FOI is reduced by a third using IFE compared with FE. In fig. 6.12 we show the trajectory the robot followed together

	IFE			FE		
	run 1	run 2	run3	run 1	run 2	run 3
Actions	5	5	5	9	8	8
Time	638	529	467	1036	837	703
Area	173	172	150	298	215	230

**Table 6.5:** Experiments use case 3: Number of conducted exploration actions and time required to find the FOI and area mapped for each experiment run applying IFE and FE.

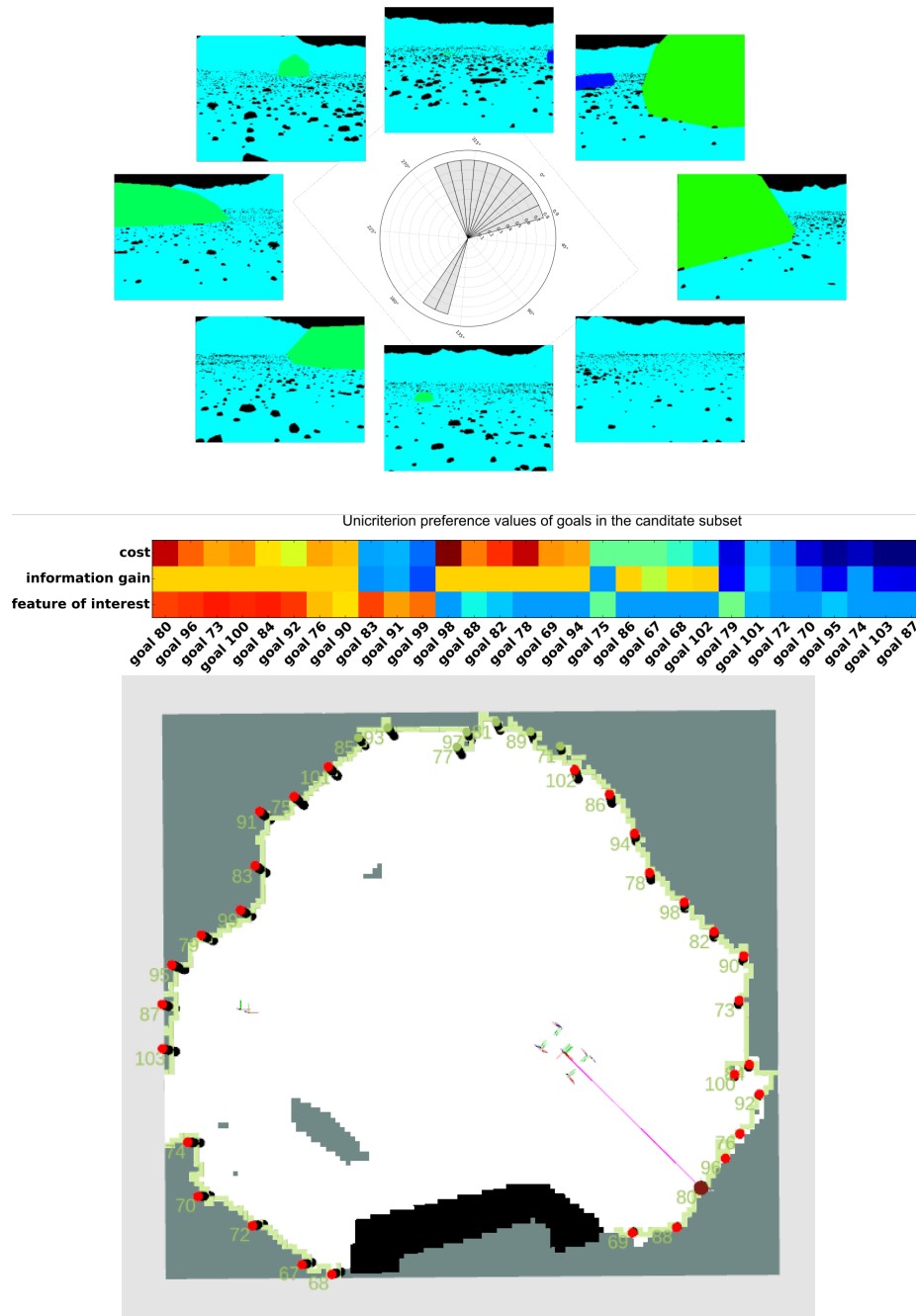
### 6.3 Use Case 3 - Informed Exploration

with the goal locations the robot visited until the FOI is found. At each experiment run applying IFE, the robot detects the FOI with the semantic camera sensor, while scanning the area after the second exploration action is conducted. From then on, the robot moves straight towards the FOI until it is found after finishing the fifth exploration action. To highlight the behavior of the robot using IFE we show the output of the third exploration action of our second experiment in fig. 6.13. We aligned the polar histogram to visibly match the sensor swipe for illustration. The second detection of a FOI depicted in the polar histogram, between 135 – 180, results from a small FOI outside of the ROI. The colormap plot of the unicriterion values, shows that the determining criterion for the choice of the next goal is the feature of interest criterion.



**Figure 6.12:** Resulting 2D occupancy grid maps and robot trajectory with marked goal locations from the three experiment runs applying IFE. The rock of interest is highlighted with a red circle.

## 6 Experimental Evaluation



**Figure 6.13:** Illustration of one decision, where to move next, applying the directed exploration. The occupancy grid map shown on the right, depicts the exploration goals. Goals marked red are candidate exploration goals within the ROI, green goals are outside of the ROI and not considered by the robot. The pink line is the path to the chosen goal 80. The colormap (bottom left) shows the unicriterion preference values for each goal for the three evaluated criteria. The goals are sorted by the final multicriteria preference degree descending from the left to the right.

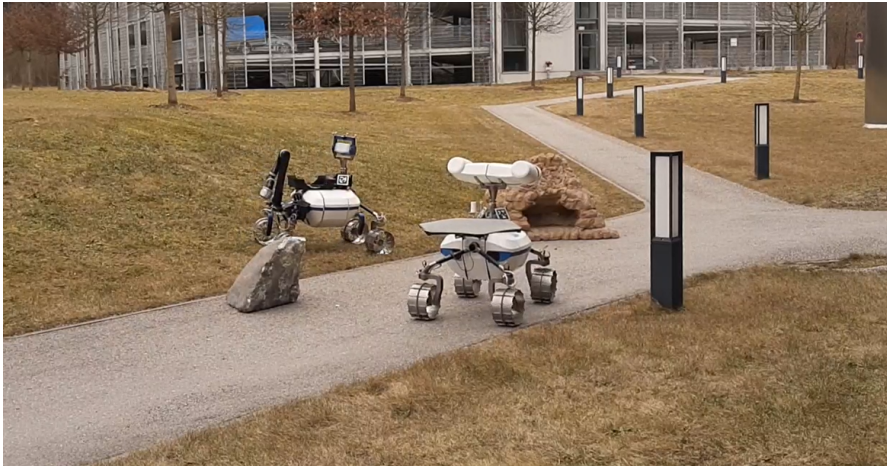
## 6.4 Use Case 4 - Multi-robot Exploration

In this section, we demonstrate use case 4 introduced in section 4.5, which outlines the coordination of two heterogeneous robots. We want to show that it is possible to apply our general exploration concept, even to coordinate a team of robots, and want to highlight the advantages of the leader-follower coordination for space exploration.

### Experimental Setup

We conducted a series of experiments with the basic simulation described in section 5.3 and demonstrate the approach once on our real LRU and LRU2, described in section 5.1, in an outdoor scenario.

We performed experiments in three different maps in simulation, which we show in fig. 6.15a-fig. 6.15c. All maps have a size of  $100m^2$  ( $10m \times 10m$ ) and are bounded by obstacles. Map 1 (fig. 6.15a) contains no obstacles, map 2 (fig. 6.15b) contains three obstacles and map 3 (fig. 6.15c) contains several obstacles dividing the area. We compare the leader-follower method, abbreviated with L-F in the following, introduced in section 4.5 with a simple coordination approach, which only considers goals that have at least 5m distance to the goal of the other robot. In detail, using the simple approach further denoted with sC, similar to the L-F method the robots exchange their current exploration goals. First, each robot evaluates if a goal is too close to the goal towards the other robot is currently moving. All goals fulfilling this condition are evaluated by the information gain and cost criterion. The distance condition to goals of other robots prevents the robots from moving to the same location, which improves the newly gained information and forestalls the robots from colliding.



**Figure 6.14:** Impression of the real world outdoor scenario with LRU (right) and LRU2 (left).

In fig. 6.14 we give an impression of the experimental setup of our real world experiment. In the experiment LRU, with its ScienceCam, is the leader robot and LRU2 the

## 6 Experimental Evaluation

follower robot. In table 6.6 we state the parameters for both robots for the simulation as well as the real world experiments.

	Leader-follower coordination		Simple coordination	
	Leader (LRU)	Follower (LRU2)	Leader	Follower
Simulation				
Cost	0.6	0.3	0.6	0.6
Information gain	0.4	0.1	0.4	0.4
Multirobot distance	-	0.2	-	-
Multirobot alignment	-	0.4	-	-
Real World				
Cost	0.6	0.2	-	-
Information gain	0.4	0.3	-	-
Multirobot distance	-	0.2	-	-
Multirobot alignment	-	0.3	-	-

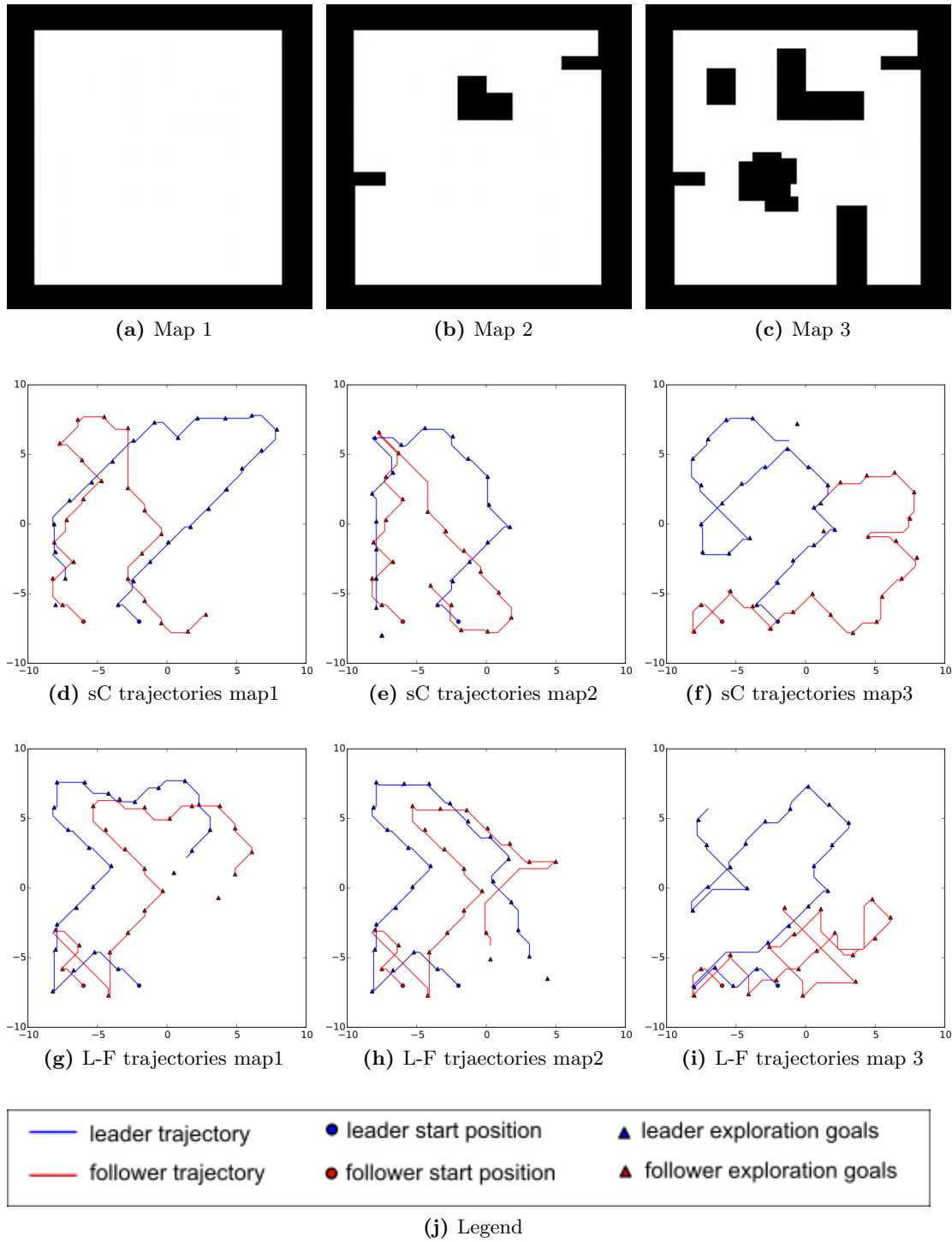
**Table 6.6:** Table stating the criteria parameters used for the experiments.

## Results

We present and discuss the results regarding the L-F coordination strategy and give an impression of the real world experiment. The basic simulation allows several robots to explore a map simultaneously. However, only the exploration goals are exchanged between the robots and not the map as described in section 2.1.3. By this limitation, only the high-level coordination behavior can be analyzed with the conducted experiments.

Fig. 6.15 shows the trajectories of the two robots for each map for the L-F, as well as the sC approach. The trajectory of the leader robot is colored blue, the trajectory of the follower robot is colored red. For simplification, we use the same notations and colors for sC, although the terms leader and follower have no meaning using sC. In fig. 6.15g and fig. 6.15h the expected behavior when applying L-F is visible. The robots explore the map in a compact way and stay always close to each other. The follower is mostly able to keep the optimal distance to the leader robot and always moves in the same direction. In contrast, in map 3 the follower was not able to keep the optimal distance. Both robots head in the same direction, however, due to obstacles blocking the path, the follower can't stay close to the leader. In fig. 6.15d-fig. 6.15f we present the resulting trajectories for each map applying sC. As the map is not shared between the robots, the robots often cross their paths and explore the same area using sC. Looking at the results of map 3, the robots are distributed over the map due to the obstacles in map3.



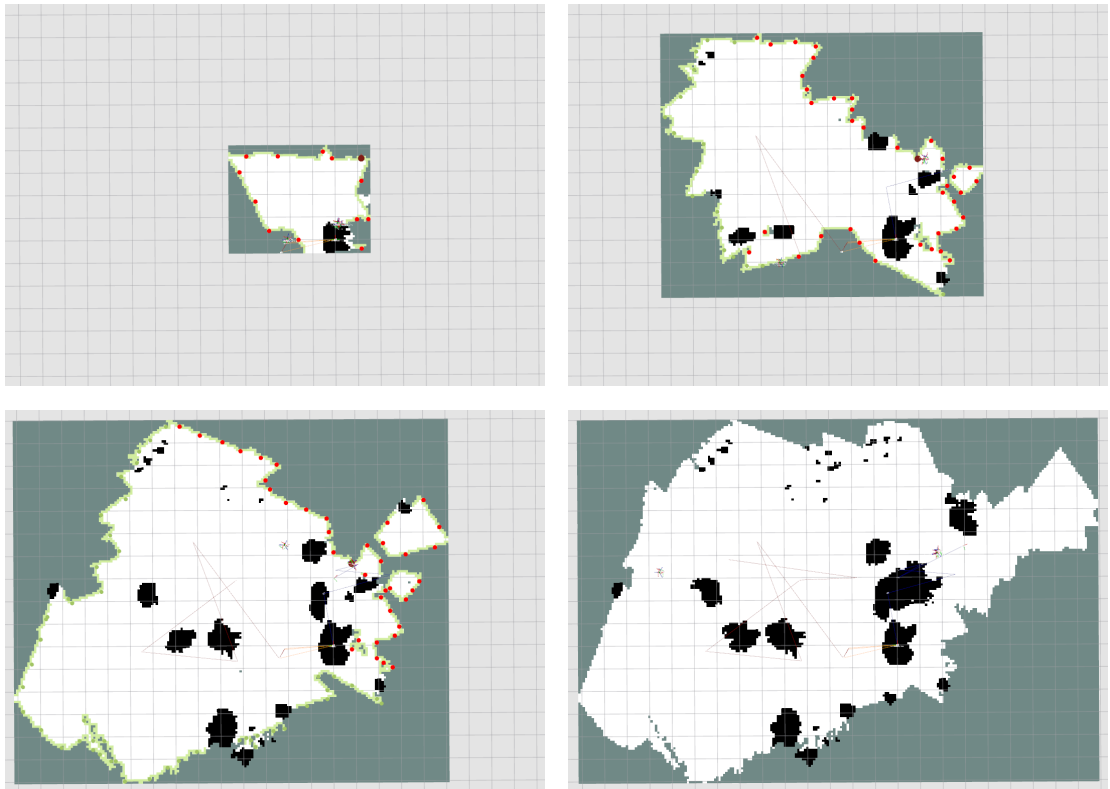


**Figure 6.15:** Top row: maps used for the experiments with the basic simulator. Middle rows: robot trajectories of the exploration experiments using sC and L-F. Bottom row: Legend for the trajectory plots.

## 6 Experimental Evaluation

The results indicate, that L-F is working properly in large areas, with few obstacles. The robots explore the map together in a compact way and are able to react fast to a request of the other robot.

On the real robotic systems, we were able to demonstrate the complete multi-robot approach, with a shared map, shared transformations, and shared exploration goals. The four occupancy grid maps shown by fig. 6.17 depict four stages of the exploration of an ROI, with fig. 6.16d showing the final map. Right at the beginning of the experiment, we manually moved the pantilt of LRU2 to look in the direction of LRU in order to establish a connection between both robots. The yellow lines on the map represent the transformation between the two map frames. During the experiments, unexpected issues with the stereo processing pipeline occurred, and the obstacle detection failed on LRU. Although we could demonstrate the algorithm runs on real prototype space rovers, the conducted experiment data is not sufficient to study the F-L coordination approach.



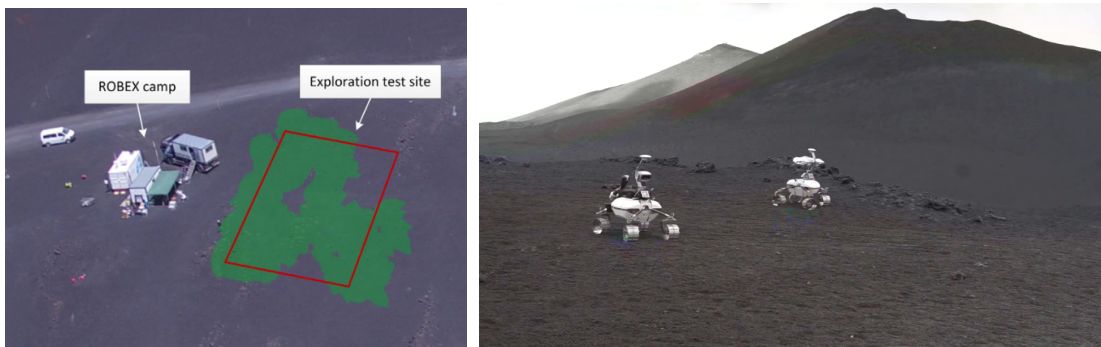
**Figure 6.16:** Common 2D occupancy grid map of LRU and LRU2 created during the real-world outdoor experiment. The exploration goals shown are the goals of LRU. Yellow lines indicate a connection between the robots SLAM graphs.

## 6.5 Multi-robot Exploration Field Tests at Mt. Etna

We had the chance to test our early work on exploration on the Moon-analogue test site on Mt. Etna, Sicily, Italy, has part of the Helmholtz Alliance Robotic Exploration of Extreme Environments (ROBEX) [18].

In our publication Lehner et al. [12] we showed how a lander with a robot on board can precisely land at a predefined spot on planetary bodies that show hard shadows, like the Earth Moon. Our algorithm is called Binary Shadow Matching, and matches pre-rendered shadows with the real shadows acquired during landing with a camera. From the correspondences we are able to triangulate accurately the position of the the lander.

On the moon-analogue test site on Mt. Etna, Schuster et al. [8] conducted a collaborative multi-robot exploration experiment applying parts of our exploration methods first described in Lehner et al. [10]. In the experiment our two lightweight rovers collaboratively explored a ROI of 25m x 20m on the test site on Mt. Etna, as illustrated in fig. 6.18.

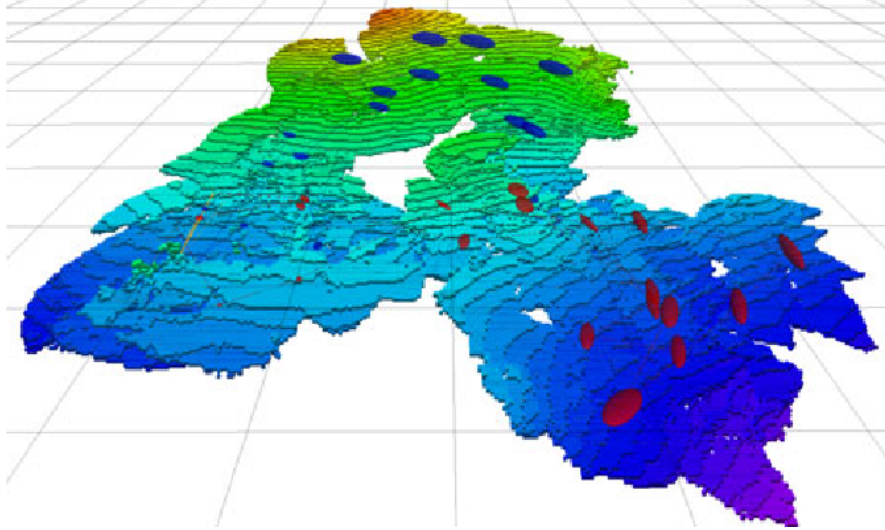


**Figure 6.17:** (a) Aerial overview of the moon-analogue test site with the ROBEX camp. The ROI is manually drawn and overlaid by the explored map (green). Image Credit: Schuster et al. [8]. (b) Illustration of the two light weight rovers LRU and LRU2 exploring collaboratively at the moon-analogue test site on Mt. Etna, Sicily, Italy.

To coordinate the two rovers, the simple coordination strategy denoted as sC in the experimental evaluation of the fourth use case (section 6.4) was used. Each robot evaluated the exploration goal locations by applying our cost  $c_{cost}$  and information gain criterion  $c_{IG}$  as detailed in section 4.2. Similar to our multi-robot experiment setup both rovers sent each other the next best exploration. The exploration goals of the respectively other rover are then included in the goal selection process to constrain the distance between both rovers. For this, Schuster et al. [8] used a minimum distance between the robots of at least 7m. Each rover computes its own path in its own local cost map and applies a local obstacle avoidance. The rovers collaboratively explored an area of  $650m^2$  while driving combined 394m [8]. In fig. 6.18 we show the resulting 3D

## 6 Experimental Evaluation

voxel map generated by Schuster et al. [8]. Due to empty batteries the rovers were not able to explore the complete ROI and had to be moved to the starting point by manually selecting waypoints.



**Figure 6.18:** 3D probabilistic voxel map build during our preliminary multi-robot exploration tests on Mt. Etna. Image credit Schuster et al. [8]

### 6.6 Runtime

In this section, we present the computation time of our MCDM exploration, especially for the usage of PROMETHEE II and analyze the effect of our extension to PROMETHEE II to reduce the runtime. Further, we show how the criteria classification can be used in general to reduce the computation time. We published the content and the results of this section first in our conference paper [11].

#### Experimental Setup

We conduct five experiments with the Rover Simulation Toolkit [103] (section 5.3) in the similar rough outdoor environment, which we already applied for the experiments for use case 1. However, we don't restrain the environment by setting an ROI, but explore the complete  $40\text{ m} \times 40\text{ m}$  area of the environment. For details please refer to section 4.2 and fig. 6.2a.

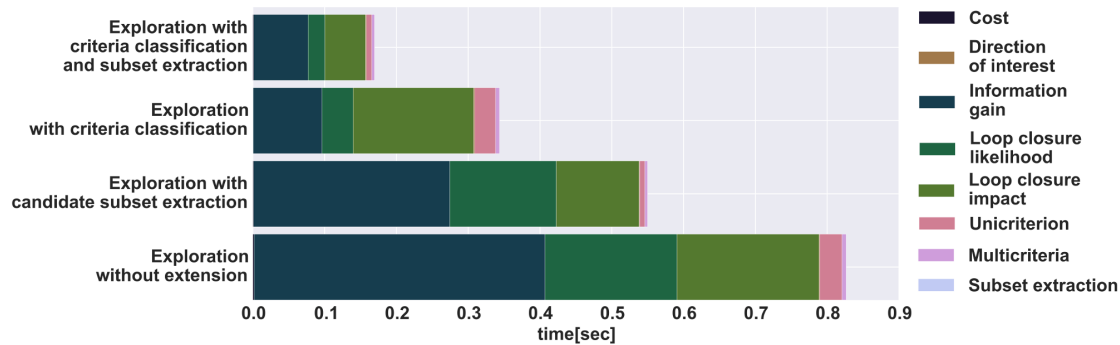
The task of the robot is to explore the area within 10 min. In table 6.7 we state the criteria and their weights used for the experiments. We compare the runtime of the basic PROMETHEE II with our extended PROMETHEE II process, by running the decision making in turn with and without our extension at each exploration action. Using our extension, a subset of candidate goals is extracted based on the cost criterion  $c_{cost}$

$c_{cost}$	$c_{doi}$	$c_{li}$	$c_{ll}$	$c_{IG}$
0.55	0.2	0.05	0.05	0.15
0.2	0.6	-	-	0.2
0.4	0.4	-	-	0.2
0.3	0.4	0.05	0.05	0.2
0.3	0.1	0.2	0.3	0.1

**Table 6.7:** Table stating the criteria parameters used for the experiments to evaluate the runtime of the decision process.

## Results

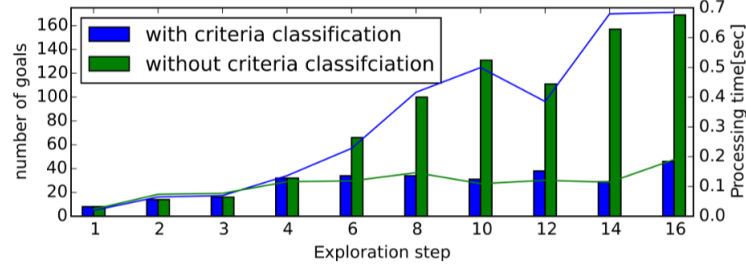
In fig. 6.19 we show a comparison of the mean computation time for the three processing steps: criteria calculation, unicriterion net flow calculation, and multi-criteria net flow calculation for an exploration experiment with and without our extensions. The stated computation time is the mean over one exploration experiment stopped after 10 min. The subset extraction can speed up the decision process by up to 30 %. The extraction of the candidate subset mainly affects the unicriterion calculation, but also the computation time of the criteria is reduced. In total, the plot shows, that the subset extraction can speed up the decision process by up to 30 %.



**Figure 6.19:** 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.

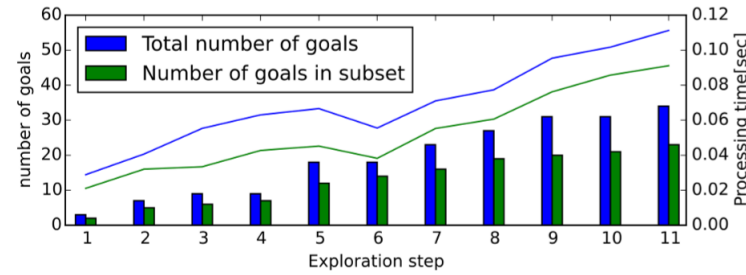
The found time reduction matches with the number of goals in the subset, which we present in fig. 6.20. The subset contains about 70%(+4,3%) of all goals, independent of the total number of goals. Most computationally expensive is the criteria evaluation. As shown in fig. 6.19 the computation of the robot-dependent criteria (cost and DOI) is much faster than the evaluation of the map-depended criteria (information gain and loop closure likelihood). That is, why the criteria classification can speed up the whole

Comparison of the processing time applying criteria classification



(a) Color

Comparison of the processing time applying the subset extraction



(b) Depth

**Figure 6.20:** 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.

decision process by up to 60 %. In fig. 6.20 we can show that other than the subset extraction, the effect of the criteria classification on the runtime increases with progressing exploration, as more goals have to be evaluated. Applying the criteria classification only the robot-dependent criteria have to be recomputed at each exploration action, the map and environment-dependent criteria only have to be recomputed for new goals or goals where the map changed. As the criteria classification mainly decreases the criteria evaluation time and the subset extraction reduces the time for the net flow calculation, both can be combined to reach a reduction of the mean processing time of decisions by approx. 70%.

## 6.7 Summary and Discussion

In this section, we summarize and discuss the results of our experimental evaluation, as well as the application of our general exploration concept based on MCDM. Further, we map our findings to the challenges identified in section 1.3 to emphasize our contributions to robotic exploration of planetary surfaces.

- *Use case 1 - autonomous exploration (section 6.1)*

We evaluated the exploration, localization, and mapping performance of our IE in simulation and a real-world indoor and outdoor scenario. By comparing IE with FE we could show the advantages of IE over FE. Applying IE twice as many loop closures occurred compared to FE. This led to a significantly lower mean absolute error using CALE and SALE and a decrease of 1m of the maximum detected error compared to FE. Applying FE the localization performance was in two cases so poor, that the exploration process couldn't be finished. We compared the mapping performance visually and could show that the final 2D occupancy grids maps generated by CALE and SALE are all intrinsically correct, whereas the 2D occupancy grid maps created by FE have a significant error in the yaw angle. We evaluated the exploration performance by analyzing the number of exploration actions and driven distance of the robot until the task is finished. It requires 2-5 exploration actions more using CALE and SALE compared to FE, as additionally to the exploration actions re-localization actions are performed. However, the loss of exploration efficiency is exceeded by the increased localization and mapping performance.

- *Use case 2 - drive-by science (section 6.2)*

We demonstrated a drive-by science mission by applying DE. With our experiments, we could show that it is possible to direct a robot towards predefined POI's, while maximizing the information gain. The benefit of using an exploration approach is, that the robot can optimize the global plan at each exploration action, to properly react on the detected environment. In our experiment, the robot was able to visit three POI's in given order.

- *Use case 3 - informed exploration (section 6.3)*

We presented and discussed the results for the search efficiency of applying IFE to find a FOI. With IFE only 5 exploration actions and at most 638 sec to find the FOI in an unknown area of approximately  $300m^2$  were required. Applying IFE the robot had to explore only half of the ROI, whereas using FE more than 80 % of the area had to be explored until the FOI was found.

- *Use case 4 - multi-robot exploration (section 6.4)*

We demonstrated, how two heterogeneous robots can be coordinated by applying the same general exploration concept we applied for the three single robot use cases. The in section 4.5 introduced L-F concept was successfully applied in two environments with a small number of obstacles. The follower robot was able to move in the same direction as the leader while keeping an optimal distance. Avoiding large distances between the robots is beneficial, to consistently share the common map and to swiftly react to requests, in order to support each other. In our real-world experiment, we were able to prove that the algorithm is running on two real prototype space rovers.

## 6 Experimental Evaluation

- *Runtime (section 6.6)*

An advanced decision making, which incorporates several criteria and, the comparison of a large number of goals, requires high computational resources. In our experiments we could show that our extension to PROMETHEE II and our criteria classification can reduce the processing time by approximately 70 %. In the large outdoor environment planetary surfaces represent, hundreds of exploration goals have to be compared. By reducing the computation time of the decision process, we enabled the application of advanced decision making methods such as PROMETHEE II for robotic planetary exploration missions.

With our demonstrations, experiments, and evaluation we directly or indirectly cover the challenges of robotic exploration of planetary surfaces identified in section 1.3. Our general exploration concept based on MCDM is able to cover the induced mission challenges. As we impressively show with the demonstration of our four different use cases, the concept can be used to cope with the changing mission requirements. With the possibility to model the different use cases solely by adding and removing criteria and, adapting their weights, an operator is able to set up a mission despite the limited communication between the operator and the robot. Our concept further allows an operator to supervise the exploration process. The decisions of the robot are reasonable and special MCDM tools<sup>2</sup> are available to visualize the decisions of the robot. We directly tackle the environmental challenges with the integrated exploration and the informed exploration. The integrated exploration with the active loop closing is the qualification to explore large unknown environments. Without keeping an accurate localization and a good map quality the mission success is not guaranteed in large environments. By incorporating search knowledge in our informed exploration, we could show how to find a FOI, i.e. prioritizing areas which are more interesting. This is important as it is not possible to explore the whole area, due to its size and limited resources. With our general exploration concept and our implemented exploration behaviors we indirectly tackle the system challenge for uncertainties. The underlying Mapping and Navigation pipeline copes the sensor uncertainties. Our information gain calculation uses a probabilistic procedure and by applying our novel loop closure likelihood and loop closure impact criterion, we consider the localization uncertainty of the robot. Partly, we directly cover the challenge of limited resources, by reducing the runtime of our decision making process significantly. Indirectly, our exploration tries to save energy by considering the cost criterion and favoring goals close by. Further, the informed exploration is able to detect aFOI faster, which also saves resources. Our whole concept of applying MCDM for exploration, also presents a solution for the challenge of conflicting objectives. The clear mapping from mission objectives/goals to exploration criteria, as presented in our criteria hierarchies, allows an operator to formulate his/her preferences. A large number of criteria of various types and with different units can be evaluated to find the goal which best fits all objectives. With our integrated exploration, we especially tackled the

---

<sup>2</sup>e.g. <http://en.promethee-gaia.net/visual-promethee.html>



## 6.7 *Summary and Discussion*

trade-off between the conflicting objectives of exploration efficiency, map quality, and localization accuracy.



# 7 Conclusion

In this chapter, we conclude our thesis by summarizing the main aspects and identifying future work.

## 7.1 Summary and Conclusion

In this thesis, we proposed a generalized concept for robotic exploration based on Multi-Criteria Decision Making, which enables an operator to model various exploration mission scenarios. In space exploration the robot is confronted with challenges induced by the hazardous and large environment of planetary surfaces, challenges arising from a limited resource budget and challenges originating from the mission itself. Due to the extreme distance between the Earth and target planets the communication between robot and operator is limited to short time slots. To maximize the scientific return of space exploration missions it is on the one hand important to increase the level of autonomy of a robot and on the other hand to never lose the possibility to supervise the robot by human operators and scientists.

Our generalized concept based on MCDM, gives the operator the possibility to model and adapt an exploration mission solely by adding and removing criteria. The exploration process can be observed in the short communication time slots and on need be adapted. To decide ‘where to move next?’, we apply amongst others the outranking method PROMETHEE II, which is originally designed to help humans structuring decision problems, thus gives results that are reasonable for a human operator.

In this thesis we present four use cases, deduced from the Exploration Roadmap of the ISECG. We model all use cases with our generalized exploration concept, by specifying the mission objectives and mapping the corresponding criteria to them, in order to evaluate the potential goal locations. In our experimental evaluations we demonstrate each use case on real space rover prototype hardware or in a high fidelity Software in the Loop Simulator.

In our first use case the task is to explore an ROI efficiently, while keeping an accurate localization and good map quality. Without an accurate localization and a correct map, mission success is not guaranteed. We present an integrated exploration, which tackles the trade-off between exploration efficiency and map quality, which is based on active loop closing. To evaluate if a loop closure can be triggered by moving to a goal, we describe the novel criteria loop closure likelihood and loop closure impact. Our evaluations show, that the map quality and localization accuracy could be significantly improved with our integrated exploration compared to a solely information gain based greedy exploration approach.

## 7 Conclusion

In our second use, we model an exploration in a drive-by science fashion. The robot has to autonomously move towards several POIs, while maximizing the information about the unknown environment. We present a directed exploration and introduce the novel criterion direction of interest. The criterion can be used to direct a robot towards a certain direction within an autonomous exploration. The operator sets the direction of interest and the robot evaluates, if a goal is lying within the given direction. We successfully demonstrate a drive-by science mission, where the robot visited three POIs in a given order, while exploring, i.e maximizing the information about the environment.

In our third use case, we directly maximize the scientific return of a mission. For this we introduce an informed exploration approach, which extends the main concept of exploration as formulated by Yaumauchi et al. [4] to:

*‘Given what you know about the world and **what you would like to know**, where should you move to gain **valuable** information?’*

We equip the robot with the capabilities to detect geologic features of interest and incorporate the probability of detecting a feature of interest in a certain direction into the exploration process. Our novel feature of interest criterion measures, if a feature of interest is present in the direction where the goal is located. In our experimental evaluations we achieved a speed up to a factor of approximately 1.6 compared to an information gain based greedy exploration to find a feature of interest in an ROI.

In our fourth use case, we present a simple multi-robot coordination approach applying the same general exploration concept and purely by adding new criteria to the exploration process. Although, our coordination method can’t keep up with some advanced methods of recent publications, e.g. [92, 86], we believe our method benefits from using our general exploration framework, which allows to install coordination between robots in the short time slots available for an operator on Earth. We implement a leader-follower coordination, by which one robot acts as leader and is followed by another robot. For space exploration mission in large environments it is important that a team of robots stays close to each other in order to quickly react on a request for support. We demonstrate our multi-robot exploration on real-space rover prototype hardware and analyze our coordination approach in simulation. In large environments with few obstacles we could show the intended leader-follower behavior. To explore a large unstructured environment, a robot has to compare hundreds of potential goal locations, which is even more challenging on the limited CPU resources of space rover hardware. To keep the processing resources in limited bounds we suggest to classify the criteria in the following three classes: robot-dependent, map-dependent and environment-dependent. Categorizing the criteria has the advantage, that it is clear which criteria have to be updated at each exploration step and which have to be updated only form time to time. Robot-dependent criteria change when the robot moves and have to be updated at each exploration step. Map-dependent criteria only change when the map where the goal is located is swiped again by the sensor. Whereas, we assume that environment-dependent criteria never change in our static space scenario. As the criteria, which computations are most time consuming can be classified as map-dependent, e.g. information and loop

closure likelihood, we could reduce the run time of the criteria evaluation about 30 %. To further speed up the decision process, we extended the PROMETHEE II algorithm. By first subtracting a subset of potential goal candidates, based on the comparison of only one criterion for each goal, we are able to reduce the runtime further. In total, we are able to reduce the runtime about approximately 70%. With that we can process hundreds of goals in a reasonable time.

In this thesis, we present a general concept for planetary exploration and develop three different single robot exploration methods and one multi-robot exploration coordination method and also demonstrate four relevant planetary exploration use cases with them. We believe, that considering robotic exploration as a general MCDM problem is an opening for more autonomy in robotic space exploration. It offers a simple interface for a human operator to model and adapt the exploration behavior to full fill challenging tasks. We have already shown four different tasks, however other applications can be easily modeled with the provided set of criteria. Section 3 is meant to be a guidance on how to model different robotic exploration tasks as MCDM problems.

## 7.2 Future Work

This thesis is both an approach towards enhanced autonomy for space exploration missions and an improved interface to human operators. Although, we indirectly consider sensor uncertainties within our exploration and the underlying SLAM system, an interesting investigation is the direct consideration of uncertainties in the criteria values. The sensor uncertainties propagate to the criteria values, which could lead to a wrong decision on the next best goal location, which in worst case could endanger the robot.

Space exploration scenarios are challenging in many ways. To be applicable in such critical scenarios it is important to investigate guarantees and policies for exploration. Additionally, to providing an interface to monitor and understand the exploration process as we have suggested in this thesis, a human operator should be able to set and rely on performance bounds, which for example consider the limited resource budget. In this thesis, we have used an exploration approach, which only plans for the next exploration action. However, to further increase the exploration efficiency, it would be worth to investigate algorithms, which plan several steps ahead. However, planning ahead also requires, an fast and accurate prediction of future robot states, which is currently a lack in robotic exploration research. A further interesting topic for future investigations is the coordination of multiple robots. In this thesis, we have presented a simple coordination as an advanced coordination approach was out of scope of this thesis. However, a team of heterogeneous robots could achieve much more complex tasks by combining their capabilities, than a single robot. In general, the exploration of large areas would be much faster with several robots. To increase the scientific return significantly, for example a flying system could be integrated as scout in the team. To fully benefit from the various capabilities of several robots in a team, a highly advanced coordination and communication strategy is required.



# Bibliography

- [1] C. Stachniss and W. Burgard. Exploring Unknown Environments with Mobile Robots using Coverage Maps. In *IJCAI*, pages 1127–1134, 2003.
- [2] H. H. Gonzalez-Banos and J.-C. Latombe. Navigation Strategies for Exploring Indoor Environments. *IJRR*, 21(10-11):829–848, 2002.
- [3] I. S. E. C. Group. The global exploration roadmap (3rd edition). Technical report, International Space Exploration Coordination Group (ISECG), 2018.
- [4] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, pages 146–151, 1997.
- [5] R.A. Yingst et al. Mars system science: Why mars remains a compelling target for solar system science. Technical report, Mars Exploration Program Analysis Group (MEPAG), 2020.
- [6] Banfield D. et al. Mars scientific goals, objectives, investigations, and priorities: 2020. Technical report, Mars Exploration Program Analysis Group (MEPAG), 2020.
- [7] J.-P. Brans and P. Vincke. Note—A Preference Ranking Organisation Method: (The PROMETHEE Method for Multiple Criteria Decision-Making). *Management Science*, 31(6):647–656, 1985.
- [8] 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.
- [9] M. Sewtz, H. Lehner, Y. Fanger, J. Eberle, M. Wudenka, M. G. Müller, T. Bodenmüller, and M. Schuster. URSim - A Versatile Robot Simulator for Extra-Terrestrial Exploration, March 2022.
- [10] 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*, pages 6191–6198, 2017.
- [11] H. Lehner, M. J. Schuster, T. Bodenmüller, and R. Triebel. Exploration of Large Outdoor Environments Using Multi-Criteria Decision Making. In *ICRA*, 2021.

## BIBLIOGRAPHY

- [12] H. Kaufmann. Shadow-based matching for robust absolute localization during lunar landings. In *IEEE Aerospace Conference*, March 2014.
- [13] M. J. Schuster, M. G. Müller, S. G. Brunner, H. Lehner, P. Lehner, R. Sakagami, A. Dömel, L. Meyer, B. Vodermayr, R. Giubilato, M. Vayugundla, J. Reill, F. Steidle, I. von Bargaen, K. Bussmann, R. Belder, P. Lutz, W. Stürzl, M. Smí\vsek, M. Moritz, S. Stoneman, A. F. Prince, B. Rebele, M. Durner, E. Staudinger, S. Zhang, R. Pöhlmann, E. Bischoff, C. Braun, S. Schröder, E. Dietz, S. Frohmann, A. Börner, H.-W. Hübers, B. Foing, R. Triebel, A. O. Albu-Schäffer, and A. Wedler. The arches space-analogue demonstration mission: Towards heterogeneous teams of autonomous robots for collaborative scientific sampling in planetary exploration. *IEEE Robotics and Automation Letters*, 5(4):5315–5322, October 2020.
- [14] M. J. Schuster, B. Rebele, M. G. Müller, S. G. Brunner, A. Dömel, B. Vodermayr, R. Giubilato, M. Vayugundla, H. Lehner, P. Lehner, F. Steidle, L. Meyer, K. Bussmann, J. Reill, W. Stürzl, I. von Bargaen, R. Sakagami, M. Smisek, M. Durner, E. Staudinger, R. Pöhlmann, S. Zhang, C. Braun, E. Dietz, S. Frohmann, S. Schröder, A. Börner, H.-W. Hübers, R. Triebel, B. Foing, A. O. Albu-Schäffer, and A. Wedler. The arches moon-analogue demonstration mission: Towards teams of autonomous robots for collaborative scientific sampling in lunar environments. In *European Lunar Symposium (ELS)*, 2020.
- [15] M. J. Schuster, M. G. Müller, S. G. Brunner, H. Lehner, P. Lehner, A. Dömel, M. Vayugundla, F. Steidle, P. Lutz, R. Sakagami, et al. Towards heterogeneous robotic teams for collaborative scientific sampling in lunar and planetary environments. 2019.
- [16] A. Wedler, M. Wilde, A. Dömel, M. G. Müller, J. Reill, M. Schuster, W. Stürzl, R. Triebel, H. Gmeiner, B. Vodermayr, K. Bussmann, M. Vayugundla, S. Brunner, H. Lehner, P. Lehner, A. Börner, R. Krenn, A. Dammann, U.-C. Fiebig, E. Staudinger, F. Wenzhöfer, S. Flögel, S. Sommer, T. Asfour, M. Flad, S. Hohmann, M. Brandauer, and A. O. Albu-Schäffer. From single autonomous robots toward cooperative robotic interactions for future planetary exploration missions. In *69th International Astronautical Congress (IAC)*, Proceedings of the 69th International Astronautical Congress (IAC). International Astronautical Federation (IAF), October 2018.
- [17] 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. Vodermayr, T. Bodenmüller, C. Brand, W. Friedl, I. Grixia, 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.
- [18] A. Wedler, M. Vayugundla, H. Lehner, P. Lehner, M. J. Schuster, S. G. Brunner, W. Stürzl, A. Dömel, H. Gmeiner, B. Vodermayr, B. Rebele, I. L. Grixia,



- K. Bussmann, J. Reill, B. Willberg, A. Maier, P. Meusel, F. Steidle, M. Smisek, M. Hellerer, M. Knapmeyer, F. Sohl, A. Heffels, L. Witte, C. Lange, R. Rosta, N. Toth, S. Völk, A. Kimpe, P. Kyr, and M. Wilde. First results of the robex analogue mission campaign: Robotic deployment of seismic networks for future lunar missions. In *68th International Astronautical Congress (IAC)*, volume 68 of *68th International Astronautical Congress (IAC)*. International Astronautical Federation (IAF), September 2017.
- [19] C. Stachniss, G. Grisetti, and W. Burgard. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. In *RSS*, pages 65–72, 2005.
- [20] C. Brand, M. J. Schuster, H. Hirschmüller, and M. Suppa. Submap Matching for Stereo-Vision Based Indoor/Outdoor SLAM. In *IROS*, 2015.
- [21] H. Hirschmüller. Stereo Processing by Semiglobal Matching and Mutual Information. *TPAMI*, 30(2):328–341, 2008.
- [22] K. Schmid, F. Ruess, and D. Burschka. Local Reference Filter for Life-Long Vision Aided Inertial Navigation. In *FUSION*, 2014.
- [23] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *AURO*, 34(3):189 – 206, 2013.
- [24] C. Hwang and K. Yoon. *Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey*. Lecture Notes in Economics and Mathematical Systems. Springer Berlin Heidelberg, 2012.
- [25] V. Belton and T. Stewart. *Multiple Criteria Decision Analysis: An Integrated Approach*. Springer Nature Book Archives Millennium. Springer US, 2002.
- [26] J. S. Dyer, P. C. Fishburn, R. E. Steuer, J. Wallenius, and S. Zionts. Multiple criteria decision making, multiattribute utility theory: the next ten years. *Management science*, 38(5):645–654, 1992.
- [27] L. G. Vargas. An overview of the analytic hierarchy process and its applications. *European journal of operational research*, 48(1):2–8, 1990.
- [28] B. Roy. Classement et choix en présence de points de vue multiples. *Revue française d’informatique et de recherche opérationnelle*, 2(8):57–75, 1968.
- [29] D. Holz, N. Basilico, F. Amigoni, and S. Behnke. Evaluating the Efficiency of Frontier-based Exploration Strategies. *ISR Robotik*, 2010.
- [30] S. Wirth and J. Pellenz. Exploration Transform: A stable exploring algorithm for robots in rescue environments. In *SSRR*, pages 1–5, 2007.
- [31] C. Stachniss, D. Hahnel, and W. Burgard. Exploration with Active Loop-Closing for FastSLAM. In *IROS*, pages 1505–1510, 2004.

## BIBLIOGRAPHY

- [32] N. Basilico and F. Amigoni. Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. *Autonomous Robots*, 31(4):401, 2011.
- [33] P. Taillandier and S. Stinckwich. Using the PROMETHEE multi-criteria decision making method to define new exploration strategies for rescue robots. In *SSRR*, pages 321–326, 2011.
- [34] C. Wang, W. Chi, Y. Sun, and M. Q.-H. Meng. Autonomous robotic exploration by incremental road map construction. *IEEE Transactions on Automation Science and Engineering*, 16(4):1720–1731, 2019.
- [35] H. Carrillo, P. Dames, V. Kumar, and J. A. Castellanos. Autonomous robotic exploration using a utility function based on rényi’s general theory of entropy. *Autonomous Robots*, 42(2):235–256, 2018.
- [36] J. A. Placed and J. A. Castellanos. Fast autonomous robotic exploration using the underlying graph structure. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6672–6679, 2021.
- [37] C. Papachristos, F. Mascarich, S. Khattak, T. Dang, and K. Alexis. Localization uncertainty-aware autonomous exploration and mapping with aerial robots using receding horizon path-planning. *Autonomous Robots*, 43(8):2131–2161, 2019.
- [38] B. Fang, J. Ding, and Z. Wang. Autonomous robotic exploration based on frontier point optimization and multistep path planning. *IEEE Access*, 7:46104–46113, 2019.
- [39] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte. An Experiment in Integrated Exploration. In *IROS*, pages 534–539, 2002.
- [40] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information Based Adaptive Robotic Exploration. In *IROS*, pages 540–545, 2002.
- [41] R. Sim and N. Roy. Global A-Optimal Robot Exploration in SLAM. In *ICRA*, pages 661–666, 2005.
- [42] R. Valencia, J. V. Miró, G. Dissanayake, and J. Andrade-Cetto. Active Pose SLAM. In *IROS*, pages 1885–1891, 2012.
- [43] L. C. Garaffa, M. Basso, A. A. Konzen, and E. P. de Freitas. Reinforcement learning for mobile robotics exploration: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [44] L. Tai and M. Liu. Towards cognitive exploration through deep reinforcement learning for mobile robots. *arXiv preprint arXiv:1610.01733*, 2016.

- [45] L. Tai and M. Liu. Mobile robots exploration through cnn-based reinforcement learning. *Robotics and biomimetics*, 3(1):1–8, 2016.
- [46] G. A. Cardona, C. Bravo, W. Quesada, D. Ruiz, M. Obeng, X. Wu, and J. M. Calderon. Autonomous navigation for exploration of unknown environments and collision avoidance in mobile robots using reinforcement learning. In *2019 South-eastCon*, pages 1–7, 2019.
- [47] R. B. Issa, M. Saferi Rahman, M. Das, M. Barua, and M. G. Rabiul Alam. Reinforcement learning based autonomous vehicle for exploration and exploitation of undiscovered track. In *2020 International Conference on Information Networking (ICOIN)*, pages 276–281, 2020.
- [48] H. Shi, L. Shi, M. Xu, and K.-S. Hwang. End-to-end navigation strategy with deep reinforcement learning for mobile robots. *IEEE Transactions on Industrial Informatics*, 16(4):2393–2402, 2020.
- [49] D. Zhu, T. Li, D. Ho, C. Wang, and M. Q.-H. Meng. Deep reinforcement learning supervised autonomous exploration in office environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7548–7555, 2018.
- [50] H. Li, Q. Zhang, and D. Zhao. Deep reinforcement learning-based automatic exploration for navigation in unknown environment. *IEEE transactions on neural networks and learning systems*, 31(6):2064–2076, 2019.
- [51] C. Craye, D. Filliat, and J.-F. Goudou. RL-IAC: An Exploration Policy for Online Saliency Learning on an Autonomous Mobile Robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, October 2016.
- [52] M. Woods, A. Shaw, D. Barnes, D. Price, D. Long, and D. Pullan. Autonomous science for an exomars rover-like mission. *Journal of Field Robotics*, 26(4):358–390, 2009.
- [53] M. Cashmore, M. Fox, D. Long, D. Magazzeni, and B. Ridder. Opportunistic planning in autonomous underwater missions. *IEEE Transactions on Automation Science and Engineering*, 15(2):519–530, 2017.
- [54] G. Hedrick, N. Ohi, and Y. Gu. Terrain-aware path planning and map update for mars sample return mission. *IEEE Robotics and Automation Letters*, 5(4):5181–5188, 2020.
- [55] A. Arora, P. M. Furlong, R. Fitch, T. Fong, S. Sukkarieh, and R. Elphic. Online multi-modal learning and adaptive informative trajectory planning for autonomous exploration. In *Field and Service Robotics*, pages 239–254. Springer, 2018.

## BIBLIOGRAPHY

- [56] A. Candela, D. Thompson, E. N. Dobrea, and D. Wettergreen. Planetary robotic exploration driven by science hypotheses for geologic mapping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3811–3818, 2017.
- [57] L. Kunze, K. K. Doreswamy, and N. Hawes. Using qualitative spatial relations for indirect object search. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 163–168. IEEE, 2014.
- [58] L. Kunze and N. Hawes. Indirect object search based on qualitative spatial relations. In *Proc. IROS, Workshop on AI-based Robotics*, 2013.
- [59] C. Dornhege, A. Kleiner, A. Hertle, and A. Kolling. Multirobot coverage search in three dimensions. *Journal of Field Robotics*, 33(4):537–558, 2016.
- [60] A. Aydemir, A. Pronobis, K. Sjöo, M. Göbelbecker, and P. Jensfelt. Object search guided by semantic spatial knowledge. *The RSS*, 11, 2011.
- [61] A. Aydemir, K. Sjo, J. Folkesson, A. Pronobis, and P. Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2818–2824. IEEE, 2011.
- [62] K. Sjöo, A. Aydemir, and P. Jensfelt. Topological spatial relations for active visual search. *Robotics and Autonomous Systems*, 60(9):1093–1107, 2012.
- [63] T. Kollar and N. Roy. Utilizing object-object and object-scene context when planning to find things. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2168–2173. IEEE, 2009.
- [64] T. H. Chung and J. W. Burdick. A decision-making framework for control strategies. In *in Probabilistic Search, in Intl. Conference on Robotics and Automation. ICRA*, 2007.
- [65] M. Lorbach, S. Hofer, and O. Brock. Prior-assisted propagation of spatial information for object search. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2904–2909. IEEE, 2014.
- [66] M. Göbelbecker, A. Aydemir, A. Pronobis, K. Sjöo, and P. Jensfelt. A planning approach to active visual search in large environments. In *Automated Action Planning for Autonomous Mobile Robots*, 2011.
- [67] L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez. Manipulation-based active search for occluded objects. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2814–2819. IEEE, 2013.
- [68] L. L. Wong, L. P. Kaelbling, and T. Perez. Using knowledge of related objects to drive attention for active object search and large-scale estimation. *IEEE ICRA, Hong Kong, China*, 2014.

- [69] T. Gedicke, M. Günther, and J. Hertzberg. Flap for caos: Forward-looking active perception for clutter-aware object search. *IFAC-PapersOnLine*, 49(15):114–119, 2016.
- [70] D. Joho and W. Burgard. Searching for objects: Combining multiple cues to object locations using a maximum entropy model. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 723–728. IEEE, 2010.
- [71] D. Joho, M. Senk, and W. Burgard. Learning search heuristics for finding objects in structured environments. *Robotics and Autonomous Systems*, 59(5):319–328, 2011.
- [72] A. Aydemir, A. Pronobis, M. Gobelbecker, and P. Jensfelt. Active visual object search in unknown environments using uncertain semantics. *Robotics, IEEE Transactions on*, 29(4):986–1002, 2013.
- [73] M. Kulich, T. Juchelka, and L. Přeučil. Comparison of exploration strategies for multirobot search. 2015.
- [74] R. Cipolleschi, M. Giusto, A. Q. Li, and F. Amigoni. Semantically-informed coordinated multirobot exploration of relevant areas in search and rescue settings. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 216–221. IEEE, 2013.
- [75] A. Rasouli and J. Tsotsos. Visual saliency improves autonomous visual search. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 111–118, 2014.
- [76] K. Shubina and J. K. Tsotsos. Visual search for an object in a 3d environment using a mobile robot. *Computer Vision and Image Understanding*, 114(5):535–547, 2010.
- [77] M. Kulich, L. Preucil, and J. J. M. Bront. Single robot search for a stationary object in an unknown environment. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5830–5835. IEEE, 2014.
- [78] M. Kulich, J. J. Miranda-Bront, and L. Přeučil. A meta-heuristic based goal-selection strategy for mobile robot search in an unknown environment. *Computers & Operations Research*, 84:178–187, 2017.
- [79] A. Rasouli and J. Tsotsos. Attention in autonomous robotic visual search. 2014.
- [80] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 476–481. IEEE, 2000.

## BIBLIOGRAPHY

- [81] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3):376–386, 2005.
- [82] A. Bautin, O. Simonin, and F. Charpillet. Minpos : A novel frontier allocation algorithm for multi-robot exploration. pages 496–508, 10 2012.
- [83] C. Stachniss, O. M. Mozos, and W. Burgard. Speeding-up multi-robot exploration by considering semantic place information. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1692–1697. IEEE, 2006.
- [84] A. Solanas and M. A. Garcia. Coordinated multi-robot exploration through unsupervised clustering of unknown space. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 1, pages 717–721. IEEE, 2004.
- [85] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme. Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2124–2130. IEEE, 2017.
- [86] C. Nieto-Granda, J. G. Rogers III, N. Fung, S. Kemna, H. I. Christensen, and G. Sukhatme. On-line coordination tasks for multi-robot systems using adaptive informative sampling. In *International symposium on experimental robotics*, pages 318–327. Springer, 2018.
- [87] M. Juliá, A. Gil, and O. Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots*, 33(4):427–444, 2012.
- [88] H. Lau. Behavioural approach for multi-robot exploration. In *Australasian Conference on Robotics and Automation*. Australian Robotics and Automation Association Inc, 2003.
- [89] R. C. Arkin and J. Diaz. Line-of-sight constrained exploration for reactive multiagent robotic teams. In *7th International Workshop on Advanced Motion Control. Proceedings (Cat. No. 02TH8623)*, pages 455–461. IEEE, 2002.
- [90] M. N. Rooker and A. Birk. Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(4):435–445, 2007.
- [91] M. Juliá, A. Gil, L. Payá, and O. Reinoso. Local minima detection in potential field based cooperative multi-robot exploration. *International Journal of Factory Automation, Robotics and Soft Computing*, 3, 2008.
- [92] S. Manjanna, A. Q. Li, R. N. Smith, I. Rekleitis, and G. Dudek. Heterogeneous multi-robot system for exploration and strategic water sampling. In *2018 IEEE*

- International Conference on Robotics and Automation (ICRA)*, pages 4873–4880, 2018.
- [93] T. Andre and C. Bettstetter. Collaboration in multi-robot exploration: To meet or not to meet? *Journal of Intelligent and Robotic Systems*, 82, 05 2016.
- [94] P. Quin, A. Alempijevic, G. Paul, and D. Liu. Expanding wavefront frontier detection: An approach for efficiently detecting frontier cells. In *Australasian Conference on Robotics and Automation, ACRA*, 2014.
- [95] J.-C. Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.
- [96] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. 2005.
- [97] C. Brand, M. J. Schuster, H. Hirschmüller, and M. Suppa. Stereo-Vision Based Obstacle Mapping for Indoor/Outdoor SLAM. In *IROS*, 2014.
- [98] E. J. Allender, R. B. Stabbin, M. D. Gunn, C. R. Cousins, and A. J. Coates. The exomars spectral tool (exospec): an image analysis tool for exomars 2020 pancam imagery. 2018.
- [99] H. Hirschmüller. *Stereo Vision Based Mapping and Immediate Virtual Walk throughs*. De Montfort University, 2003.
- [100] H. Lau, S. Huang, and G. Dissanayake. Optimal search for multiple targets in a built environment. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3740–3745. IEEE, 2005.
- [101] 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.
- [102] S. G. Brunner, F. Steinmetz, R. Belder, and A. Dömel. Rafcon: A graphical tool for engineering complex, robotic tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3283–3290, 2016.
- [103] M. Hellerer, M. J. Schuster, and R. Lichtenheldt. Software-in-the-Loop Simulation of a Planetary Rover. In *i-SAIRAS*, 2016.
- [104] M. Hellerer, T. Bellmann, and F. Schlegel. The dlr visualization library - recent development and applications. In *The 10th International Modelica Conference 2014*, Linköping Electronic Conference Proceedings, pages 899–911. LiU Electronic Press, March 2014.
- [105] M. J. Schuster, C. Brand, S. G. Brunner, P. Lehner, J. Reill, S. Riedel, T. Bodenmüller, K. Bussmann, S. Büttner, A. Dömel, W. Friedl, I. Grix, M. Hellerer, H. Hirschmüller, M. Kassecker, Z.-C. Márton, C. Nissler, F. Ruess, M. Suppa, and

## BIBLIOGRAPHY

- A. Wedler. The LRU Rover for Autonomous Planetary Exploration and its Success in the SpaceBotCamp Challenge. In *ICARSC*, 2016.
- [106] M. Schadler, J. Stückler, and S. Behnke. Data set Spacebot Arena. [http://www.ais.uni-bonn.de/mav\\_registration/](http://www.ais.uni-bonn.de/mav_registration/), 2014.