

Konzept und Simulation für die autonome Navigation von planetaren Rovern

Stefan Kral

DIPLOMARBEIT

eingereicht an der

TECHNISCHEN UNIVERSITÄT BERLIN
INSTITUT FÜR LUFT- UND RAUMFAHRTTECHNIK
FACHBEREICH RAUMFAHRTTECHNIK

im Mai 2011

Betreut durch: Prof. Dr.-Ing. Klaus Briß (TU Berlin)
Dipl.-Ing. Walter Frese (TU Berlin)
Dipl.-Inf. Frank Trauthan (DLR)

Zusammenfassung

Im Rahmen dieser Diplomarbeit wurde ein grundlegendes Verfahren zur autonomen Navigation von planetaren Rovern entwickelt. Dazu konnte der Ansatz der Sensor-based Random Tree (SRT) Exploration adaptiert und für die Anwendung in dreidimensionalen Umgebungen erweitert werden. In einer ersten Umsetzung, basierend auf einer simulierten Planetenoberfläche, wurde die Funktion und Leistungsfähigkeit des vorgeschlagenen Verfahrens demonstriert.

Das Konzept für eine Explorationsstrategie wurde aus den Zielen einer Beispielmision abgeleitet. Für die beschriebene ExoPlanet-Mission ist vorgesehen, mehrere Mikro-Rover in einer ausgewählten Marsregion zu landen, die daraufhin autonom das Zielgebiet in Vorbereitung auf eine Geologiemitmission flächendeckend kartieren und klassifizieren. Zur Verifikation des Navigationsverfahrens und zur Diskussion hinsichtlich der Erfüllbarkeit des Missionskonzepts und der sich daraus ergebenden Anforderungen wurde zudem eine Software für die Modellierung planetarer Landschaften entwickelt. Mit dem Ziel, Sensordaten für die Explorationsrover zu generieren, wurden bei der Geländesimulation verschiedene Parameter berücksichtigt. Der unterschiedliche Geländeanstieg ergab sich direkt aus den verwendeten Höhenprofilen der Mars Reconnaissance Orbiter Mission. Darüber hinaus wurden weitere Navigationskriterien, wie Bodenbeschaffenheit und Unebenheiten sowie Hindernisse modelliert.

Mit der Durchführung einer Closed-Loop-Simulation konnte als Ergebnis dieser Arbeit das entwickelte Gesamtkonzept für die autonome Navigation evaluiert werden. Dazu wurde ein einzelner Explorationsrover betrachtet sowie eine Flotte aus mehreren Rovern. In der Diskussion der Resultate wurden abschließend auch Möglichkeiten zur Verbesserung und Erweiterung des Verfahrens aufgezeigt.

Vielen Dank Frank Trauthan
Martin Lingenauber
meiner Familie

Inhalt

1	Motivation und Problemstellung	1
1.1	Einleitung	1
1.2	Aufgabenstellung	3
1.3	Aufbau und Inhalt der Arbeit	4
2	ExoPlanet – eine Beispielmission	6
2.1	Marsforschung als internationale Strategie	6
2.2	Missionsbeschreibung	7
2.2.1	Mission Statement	7
2.2.2	Missionsziele	7
2.3	Operationsgebiet	8
2.4	Konzept	8
3	Autonome Navigation	11
3.1	Begriffseinordnung und -definition	11
3.1.1	Autonomie	12
3.1.2	Autonome Rover	12
3.1.3	Navigationsaufgaben für autonome mobile Systeme	14
3.2	Verfahren als Grundlage für die autonome Navigation	16
3.2.1	Kartenerstellung	16
3.2.2	Lokalisierung	18
3.2.3	Pfadplanung	21
3.2.4	Exploration	23
3.3	Methoden zur Pfadsuche	25
3.3.1	Der Algorithmus von Dijkstra	26
3.3.2	Der A*-Algorithmus	29
3.3.3	Der D*-Algorithmus	29
3.3.4	Weitere Verfahren zur Pfadsuche	29
4	Landschaftsmodellierung	31
4.1	Darstellung von 3D-Geländemodellen	31
4.1.1	Das ROAM-Verfahren	32
4.1.2	Geometrical Mipmaps	33
4.1.3	Paging	33

4.2	Anwendungen für die Simulation von Planetenoberflächen	34
4.2.1	NASA: <i>Maestro</i> , <i>SimScape</i> , <i>ROAMS</i> , <i>SOOPS</i>	35
4.2.2	ESA: <i>PANGU</i> , <i>3DROV</i>	36
4.3	Entwurf einer Simulationsumgebung für Geländemodelle	37
4.3.1	Anforderungen	37
4.3.2	Konzept und Funktionen der OGRE 3D Grafikkbibliothek	39
4.3.3	Geländedarstellung basierend auf MRO-Höhenprofilen	41
4.3.4	Modellierung der Geländeparameter	43
4.3.5	Ergebnis der Geländesimulation	47
5	Navigationskonzept für planetare Rover	50
5.1	Entwurf einer Explorationsstrategie	50
5.1.1	Anforderungen und Problemstellung	50
5.1.2	Das SRT-Verfahren	52
5.1.3	Beschreibung der ExoPlanet-Explorationsstrategie	55
5.1.4	Integration von Geländesimulation und Navigationsanwendung	59
5.2	Durchführung der Simulation	62
5.2.1	Exploration mit einem Rover	62
5.2.2	Exploration mit mehreren Rovern	65
6	Bewertung und Diskussion	67
6.1	Bewertung der Ergebnisse	67
6.2	Schlussfolgerungen und Fazit	72
6.3	Ausblick	74
	Quellenverzeichnis	76
	Literatur	76
	Online-Quellen	83

Abkürzungen

DLR	Deutsches Zentrum für Luft- und Raumfahrt
DTM	Digital Terrain Model
ESA	European Space Agency
GIS	Geoinformationssystem
GPU	Graphics Processing Unit
HiRISE	High Resolution Imaging Science Experiment
JPL	Jet Propulsion Laboratory
LOD	Level of Detail
MER	Mars Exploration Rover
MOC-NA	Mars Observer Camera – Near Angle
MRO	Mars Reconnaissance Orbiter
MSL	Mars Science Laboratory
NASA	National Aeronautics and Space Administration
OGRE	Object-Oriented Graphics Rendering Engine
SLAM	Simultaneous Localization And Mapping
Sol	Marstag (solar day)
SRT	Sensor-based Random Tree

Abbildungen

1	Mars-Rover	1
2	Operationsgebiet der ExoPlanet-Mission	8
3	Konzepte für planetare Mikro-Rover	10
4	Zusammenhang zwischen den unterschiedlichen Navigationsaufgaben	15
5	Stereobild der vorderen MER-Hazcam	16
6	Bildprojektion basierend auf Tiefeninformationen	17
7	Lokalisierung mit Hilfe von Landmarken	19
8	Lokales Geländemodell der Roverumgebung	20
9	Globale und lokale Pfadplanung	22
10	Einfluss der Kostenfunktion auf die Pfadplanung	23
11	Innenraum-Exploration	24
12	Repräsentation einer metrischen Karte als Graph	27
13	Algorithmus von Dijkstra	28
14	Einfaches Geländemodell eines Kraters	31
15	Schematischer Renderingprozess	32
16	Verfahren zur Darstellung von Geländemodellen	34
17	Anwendungen der NASA für die 3D-Visualisierung und Simulation	35
18	Anwendungen der ESA für die 3D-Visualisierung und Simulation	37
19	Beispiele für die Visualisierung mit OGRE 3D	39
20	Schematisches UML-Diagramm der OGRE 3D Bibliothek	40
21	Geländemodellierung auf Grundlage von MRO HiRISE Daten	42
22	Simulation unterschiedlicher Geländeparameter	44
23	Ergebnis der Geländesimulation	48
24	McMurdo-Panorama als Tiefenkarte	49
25	Prinzip der SRT-Exploration	53
26	Repräsentation der metrischen Umgebung	54
27	Prinzip der ExoPlanet-Explorationsstrategie	56
28	Weitere Beispiele für Navigationskarten.	57
29	Komponenten der integrierten Simulationsplattform	59
30	Sequenzdiagramm für den Simulationsprozess	61
31	Die ExoPath-Benutzeroberfläche	62
32	Ergebnis der simulierten Exploration	63
33	Ergebnis der simulierten Exploration mit mehreren Rovern	66

1 Motivation und Problemstellung

1.1 Einleitung

Im Jahr 2004 landeten auf dem Mars zwei mobile Roboter mit dem Ziel, die Planetenoberfläche zu erkunden. Die Mars Exploration Rover (MER) Spirit und Opportunity (Abbildung 1a) sind in der Lage, mit Hilfe einer Vielzahl wissenschaftlicher Instrumente die Bodenbeschaffenheit und die Zusammensetzung von Gesteinen zu untersuchen, um Rückschlüsse auf geologische Prozesse zu ermöglichen, die den Planeten geformt haben. Als Hauptziel der MER-Mission, die bis heute, sieben Jahre nach der Landung der Rover, nicht beendet ist, sollen Hinweise darauf gefunden werden, ob in früheren Zeiten auf dem Mars Wasser vorhanden war.

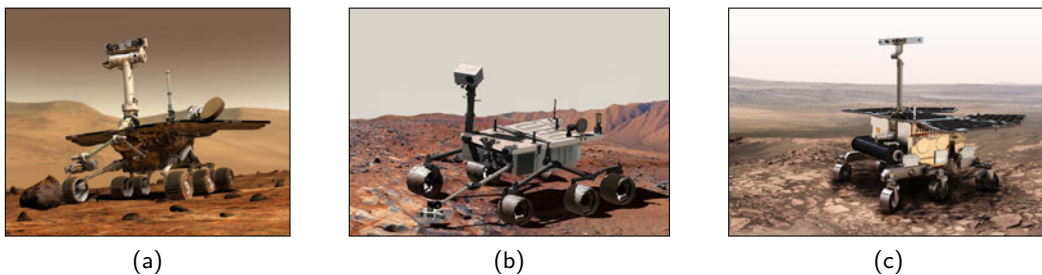


Abbildung 1. Verschiedene Modelle von Mars-Rovern. (a) Abbildung der baugleichen Rover Spirit und Opportunity der MER-Mission. (b) Der MSL-Rover Curiosity. (c) Ein Entwicklungsmodell des ExoMars-Rovers. Quellen a: [35], b: [31], c: [16].

Der Einsatz der Mars Exploration Rover stellt einen enormen technischen und wissenschaftlichen Erfolg dar. Damit zeigt sich, dass mobile Roversysteme als wichtiges Instrument für die planetare Exploration gelten. So sind auch zukünftige Missionen in Planung, um noch mehr Erkenntnisse über die Entstehung unseres Sonnensystems anhand von Untersuchungen auf der Marsoberfläche zu sammeln. Als MER-Folgemission entwickelt die NASA das Mars Science Laboratory (MSL). Der Start dieses Rovers mit dem Namen Curiosity (siehe Abbildung 1b), ist für Ende 2011 geplant. Auch die ESA möchte im Rahmen der ExoMars-Mission 2018 ein Explorationsfahrzeug (Abbildung 1c) auf der Marsoberfläche landen.

Für den Forschungsbereich der Planetengeologie ist die Analyse von Boden- und Gesteinsproben eine wichtige Quelle für den Erkenntnisgewinn. Auch wenn sich viele Ergebnisse aus Daten orbitaler Fernerkundungsmissionen ableiten lassen, ist dennoch eine direkte Untersuchung der Materialien notwendig, um Detailfragen tief greifender beantworten zu können. Dazu wurden

bereits verschiedene Explorationssonden auf Planetenoberflächen gelandet, wobei die mobilen Systeme als besonders leistungsfähige Instrumente gelten – an die aber dementsprechend auch hohe wissenschaftliche und technische Ansprüche gestellt werden. Allgemein lassen sich für die Planung und Durchführung von robotischen Missionen folgende charakteristische Randbedingungen ableiten:

- lange Signallaufzeiten, Kommunikation über Relais-Systeme
→ keine Echtzeitsteuerung möglich,
- komplexe kinematische Systeme, kontinuierliche offene Umgebung
→ viele Freiheitsgrade, schwierige Modellierung,
- begrenzte Strukturmaße
→ Oberflächen der Solarpanele/Energieversorgung nicht beliebig erweiterbar.

Aus diesen designbestimmenden Randbedingungen entstehen bei mobilen Sonden insbesondere an die Steuerung hohe Anforderungen. Damit verbunden ist ein großer operationeller Aufwand und Personalbedarf (bei der täglichen Planung für die MER-Mission waren am Anfang 150 Ingenieure und Wissenschaftler beteiligt), was insbesondere bei mehreren gleichzeitig operierenden Rovern ein Problem darstellt.

Als problematisch gilt auch die zeitliche Verzögerung, die sich aus der Teilung der Prozesse Planung, Ausführung und Erfolgsbewertung bei sequenziellen Manövern ergibt (z. B. Navigation zu einem Stein, Positionierung von Rover und Greifarm, Durchführung der Bohrung). Vom Zeitpunkt einer wissenschaftlichen Vorgabe bis zum Beginn ihrer Durchführung vergehen deshalb oft mehrere Missionstage. Für eine effiziente Nutzung der Missionsdauer ist es daher notwendig, dass die Roversysteme möglichst viele operationelle Aufgaben, besonders im Bereich der Navigation, selbständig ausführen. So ist der Erfolg der MER-Mission auch darauf zurückzuführen, dass dafür Rover entwickelt wurden, die auf Grundlage einer Zielvorgabe automatisch ihre Antriebssysteme ansteuern, eine Wegstrecke planen und Hindernissen ausweichen können. Zudem sind sie bereits in der Lage, die Bewegung während der Ausführung der Manöver zu überwachen und wenn notwendig zu korrigieren.

Durch die zunehmende Komplexität der Steuerung von planetaren Rovern sind leistungsfähigere Softwaresysteme zur Unterstützung der Entwicklung und im Betrieb notwendig. Visualisierung und Simulation sind zwei wesentliche Einsatzbereiche in denen Computeranwendungen helfen können, zuverlässige Systeme zu entwerfen und zu betreiben. Die Nutzung reicht vom Test einzelnen Algorithmen bis zur Verifikation der kompletten Onboard-Datenverarbeitung. Gleichzeitig ist es für die operationellen Entscheidungen unter technischen, aber auch wissenschaftlichen, Gesichtspunkten notwendig, den Istzustand des Systems genau zu erfassen. Die Visualisierung der Systemparameter sowie der aktuellen Mess- und Umgebungsdaten stellt hierfür ein essenzielles Hilfsmittel dar. Aber auch Simulationsläufe können während des Betriebs beitragen, Fehlentscheidungen zu vermeiden oder geplante Manöver zu optimieren. Zudem kann der Personalaufwand reduziert werden und die Wissenschaftler und Ingenieure können sich auf die Problemstellungen der primären Missionsziele konzentrieren.

1.2 Aufgabenstellung

In der bisherigen Entwicklung von planetaren Explorationsrovern sind diese meist als autark agierende Sonden konzipiert. Umgebungserkundung, wissenschaftliche Analyse, Kommunikation – alle Hauptaspekte der Mission werden von einem einzigen System durchgeführt. Für zukünftige Explorationsziele können sich aber sehr unterschiedliche Anforderungsprofile ergeben, die bisherigen Konzepte sind dann nicht ohne Weiteres übertragbar. Die Erkundung von weit entfernten Planeten und Monden oder die großflächige geologische Untersuchung einer Planetenoberfläche erfordern z. B. als technische Randbedingungen:

- einen noch höheren Grad an Autonomie sowie
- eine zunehmende Anzahl agierender Sonden, diese müssen miteinander kooperieren.

Unter Berücksichtigung dieser Vorgaben soll als Ergebnis der vorliegenden Diplomarbeit ein Navigationskonzept für planetare Rover beschrieben und demonstriert werden. Dazu sind verschiedene Teilaufgaben zu bearbeiten.

a) Missionsbeschreibung und Konzeptvorschlag

Als praktischer Bezug für die weitere Konzeptentwicklung soll zunächst eine Beispielmision zur planetaren Exploration definiert werden. Aus den formulierten Zielen und einem daraus abgeleiteten Missionskonzept ergeben sich die Anforderungen an die zu entwickelnde Explorationsstrategie.

b) 3D-Modellierung einer planetaren Landschaft

Das für eine Explorationsaufgabe notwendige Navigationsverfahren setzt sich aus mehreren Verarbeitungsschritten zusammen. Bereits der Entwurf der einzelnen Komponenten kann anhand von simulierten Sensordaten verifiziert werden. Dafür ist eine Software zu entwickeln, mit der Geländemodelle und deren Ansichten aus der jeweiligen Roverperspektive berechnet und dargestellt werden können. Die Anforderungen für die Modellierung einer planetaren Landschaft leiten sich aus der definierten Beispielmision sowie aus der Zielstellung der Simulation eines Navigationsverfahrens ab.

c) Entwicklung und Simulation des Navigationsverfahrens

Aufbauend auf den bisherigen Ergebnissen soll ein Navigationskonzept für planetare Rover erarbeitet und durch eine Simulation demonstriert werden. Das vorgestellte Verfahren ist dazu ebenfalls als Softwareanwendung zu implementieren und in die entwickelte Geländesimulation zu integrieren. In der Simulation soll der gleichzeitige Einsatz mehrerer Rover möglich sein.

Anhaltspunkte beim Entwurf der Simulationsumgebung können die bereits eingesetzten Anwendungen der verschiedenen Raumfahrtagenturen sein. Die ESA und die NASA verfügen über Software, die im Bereich der Simulation und Visualisierung bereits eine Vielzahl an Funktionen

bieten. Ein Beispiel für diese Entwicklungen ist das Projekt *Science Operations on Planetary Surfaces*, in der die Möglichkeiten einer umfassenden Simulationsplattform demonstriert wurden. In einem Video über die Projektziele [5] wurde die Bedeutung derartiger Anwendungen nochmals zusammengefasst und soll auch für diese Arbeit als Motivation dienen.

Die im Auftrag der Raumfahrtagenturen entwickelte Software ist meist nicht frei verfügbar, nicht quellenoffen und im Fall der NASA sogar teilweise von den International Traffic in Arms Regulations (ITAR) mit einem Exportverbot belegt. Dies stellt mitunter ein Problem dar, da die Entwicklungen entweder nicht genutzt oder nicht für allgemeine Zwecke angepasst werden können. Daher soll als Vorgabe für die Diplomarbeit die zu erstellende Software möglichst auf bereits existierenden Open Source Projekten aufbauen. Auch weil das Grundprinzip der Open Source Entwicklung, die Offenlegung der Lösungen für die gemeinschaftliche Verifikation, Verbesserung oder Erweiterung, auch für Raumfahrtmissionen gelten sollte. Insbesondere für die Exploration unseres Sonnensystems, da dieses Ziel auch mit einer gesellschaftlichen und kulturellen Bedeutung verbunden ist.

1.3 Aufbau und Inhalt der Arbeit

Im Folgenden wird mit einer kurzen Inhaltsangabe zu den einzelnen Kapiteln ein Überblick über die vorliegende Diplomarbeit gegeben. Gleichzeitig werden die wesentlichen Schwerpunkte erläutert und dargestellt, in welchen Zusammenhang die verschiedenen Abschnitte zur Problemstellung stehen. Dem einleitenden Teil, in dem bereits die Aufgaben und Ziele für diese Arbeit formuliert und thematisch eingeordnet wurden, folgen fünf weitere Kapitel.

Als Arbeitsgrundlage wird in Kapitel 2 eine Mission für die planetare Exploration definiert. Damit wird ein praktischer Bezug für die weitere Konzeptentwicklung hergestellt. Die Beispielmision mit dem Namen *ExoPlanet* dient als Zielvorgabe, um sowohl für die Entwicklung einer Explorationsstrategie als auch für die Umsetzung in Form einer Softwaresimulation Anforderungen abzuleiten. Für die ExoPlanet-Mission, die einen Beitrag zur Erforschung des Mars leisten soll, wird ein Mission Statement formuliert und damit verbunden die wesentlichen Zielstellungen festgelegt. Des Weiteren wird ein Missionsgebiet abgegrenzt und ein grundlegendes Gesamtkonzept für die Zielerreichung beschrieben.

Im 3. Kapitel werden die Grundlagen für die autonome Navigation von planetaren Rovern erarbeitet, um darauf die spätere Konzeptentwicklung für ein eigenes Navigationsverfahren aufzubauen. Für diese Aufgabe wird zunächst eine Begriffsdefinition vorgenommen. Anschließend werden für die allgemeine Problematik der autonomen Navigation die grundlegenden Problemstellungen abgegrenzt. Für die Bereiche Kartenerstellung, Lokalisierung, Pfadplanung und Exploration werden zudem verschiedene Methoden als Lösungsansätze für diese Teilprobleme vorgestellt. Die Ausführungen zu diesen Themen sind bewusst etwas weiter gefasst, denn die Auseinandersetzung mit den unterschiedlichen Aspekten der Rovernavigation ist notwendig, um eine realistische Explorationsstrategie zu erarbeiten und zu beurteilen.

Das darauf folgende 4. Kapitel hat die Modellierung und Visualisierung einer planetaren Landschaft zum Thema. Im Ergebnis dieses Kapitels soll eine Software zur Geländesimulation entwickelt werden. Dazu werden zunächst wichtige grundlegende Verfahren beschrieben, die eine Darstellung von 3D-Objekten, insbesondere mit Hinblick auf detaillierte Landschaftsmodelle, ermöglichen. Anschließend werden Simulationsanwendungen vorgestellt, die im Bereich der Raumfahrt entwickelt und eingesetzt werden. Darauf folgt die Formulierung der Anforderungen für eine eigene Simulationsumgebung. Im Zusammenhang mit der ExoPlanet-Beispielmission werden als Modellierungsgrundlage Höhenprofile des Mars verwendet, die um verschiedene für die Navigation relevante Geländeeigenschaften erweitert werden. Als Grundlage für die Umsetzung der 3D-Darstellung wird die Grafikkbibliothek OGRE 3D verwendet, deren Aufbau und Kernfunktionen ebenfalls kurz beschrieben werden. Abschließend werden die Ergebnisse präsentiert und mit Bezug auf die ursprünglichen Anforderungen diskutiert.

Der Beschreibung der Grundlagen der autonomen Rovernavigation und der Entwicklung einer Geländesimulation folgt das 5. Kapitel, in dem das ExoPlanet-Explorationskonzept erarbeitet und vorgestellt wird. Für den Entwurf eines geeigneten Verfahrens werden zunächst einige Anforderungen formuliert. Anschließend wird ein eigener Ansatz für die autonome Navigation erläutert, wobei das Prinzip für die Exploration dem SRT-Verfahren folgt, das an dieser Stelle ebenfalls beschrieben wird. Außerdem wird dargestellt, wie die Navigationskomponenten in die Geländesimulation integriert werden können, da für die Darstellung der Funktionsweise der Algorithmen bereits erste Simulationsdaten verwendet werden.

Ebenfalls in diesem Kapitel werden die Ergebnisse einer Closed-Loop-Simulation präsentiert. Dazu wurden zwei verschiedene Simulationsläufe durchgeführt. Zunächst ein Versuch mit einem einzelnen Rover und darüber hinaus mit einer Flotte bestehend aus vier Rovern.

Im abschließenden Kapitel werden die erzielten Ergebnisse der Simulation bewertet. Dazu werden die verschiedenen Anforderungen nochmals aufgegriffen und mit Hinblick auf die Aufgabenstellung wird untersucht, inwieweit die eigenen Ansätze die Vorgaben erfüllen können. Gleichzeitig wird an dieser Stelle diskutiert, welche Vorteile, aber auch kritische Aspekte, mit den erarbeiteten Lösungen verbunden sind und welche Erweiterungsmöglichkeiten bestehen. Mit einem thematischen Ausblick wird die Arbeit insgesamt abgeschlossen.

2 ExoPlanet – eine Beispielmision

Im Übereinkommen zur Gründung einer europäischen Weltraumorganisation aus dem Jahr 1980 heißt es: „Die Vorhaben der ESA sind darauf ausgerichtet, [...] mehr über die Erde, ihre unmittelbare Umgebung im Weltraum, das Sonnensystem und das Universum in Erfahrung zu bringen. [...] Die ESA arbeitet außerdem eng mit Raumfahrtinrichtungen außerhalb Europas zusammen, um die Vorteile der Raumfahrt der gesamten Menschheit zugutekommen zu lassen. Die Raumfahrt regt zu Forschung, Innovationen und technologischen Entwicklungen an.“ [ESA05]

Um dieser Bestimmung gerecht zu werden, sind langfristige Strategien und Herausforderungen notwendig, an denen die einzelnen Missionen ausgerichtet werden können. Dabei ist eines dieser Ziele die Erforschung des Planeten Mars. In Zusammenarbeit mit der NASA werden im Rahmen der ExoMars-Mission und dem Mars Robotic Exploration Preparatory (MREP) Programm konkrete Konzepte erarbeitet, die mit neuen technischen Möglichkeiten und wissenschaftlichen Fragestellungen einhergehen.

2.1 Marsforschung als internationale Strategie

Besonders für den Bereich der Astrobiologie kann die Erkundung der Marsoberfläche wertvolle Erkenntnisse liefern. Aber auch die Planetengeologie bietet interessante Forschungsfelder, die zu einem besseren Verständnis der Einflüsse von Wasser, Atmosphäre oder Plattentektonik auf die frühzeitliche Entwicklung des Himmelskörpers führen können. Die Fragen aus den Gebieten Mineralogie, Vulkanismus, Kraterbildung und womöglich nach einem globalen Klimawandel können auch für die Erde wichtige Zusammenhänge aufzeigen. Mit diesem Hintergrund hat die NASA für die Erforschung des Mars verschiedene strategische Ziele vereinbart [30]. Die zukünftigen Missionen sollen:

1. die Frage beantworten, ob es auf dem Mars Leben gibt oder gab,
2. das Klima des Planeten erforschen,
3. die Geologie und geologische Prozesse des Planeten charakterisieren sowie
4. eine bemannte Mission zum Mars vorbereiten.

Im Wesentlichen gleichen diese Ziele denen der europäischen Programme [17], auch in Verbindung mit dem Ziel der Entwicklung und Erprobung neuer Technologien. Das betrifft wissenschaftliche Instrumente, Raumfahrtsysteme, robotische Anwendungen sowie neuartige Verfahren und Methoden für ihren Einsatz.

2.2 Missionsbeschreibung

Als Hauptvorgabe für die Ausrichtung aller Entwicklungsaktivitäten einer Raumfahrtmission wird ein Mission Statement formuliert. In dieser Form werden nachfolgend auch die Absichten und Ziele der ExoPlanet-Mission charakterisiert.

2.2.1 Mission Statement

Die Strategien und Programme der europäischen Raumfahrt für die nächsten 20 Jahre hat die ESA unter das Motto *Cosmic Vision* gestellt [15]. Eines der großen Ziele für diese Zeit ist die Erforschung des Mars, womit eine technologische, wissenschaftliche aber auch gesellschaftliche Weiterentwicklung erreicht werden soll.

Als ein Meilenstein für die Mars-Exploration unter diesem Leitbild stellt sich auch die Mission ExoPlanet dar. Ziel der Mission ist es, sowohl neue wissenschaftliche Erkenntnisse zu ermöglichen als auch nachfolgende Missionen vorzubereiten. Dazu soll ein zusammenhängendes Gebiet der Planetenoberfläche präzise kartiert und klassifiziert werden. Der Einsatz von mobilen autonomen Sonden soll gleichzeitig neue technische Möglichkeiten aufzeigen, denn die planetare Exploration wird in Zukunft auf einen hohen Grad an Autonomie angewiesen sein, um Kosten zu verringern, die Zuverlässigkeit zu erhöhen und gleichzeitig die Missionsziele zu erweitern.

Die Mission ist Teil einer Gesamtstrategie für den Einsatz von planetaren Sonden. Dabei soll eine funktionale Teilung in spezifische Systeme angestrebt werden. ExoPlanet soll zeigen, wie mit Hilfe von Scout-Rovern umfassendes Kartenmaterial erstellt werden kann. Auf Grundlage dieser Daten soll in einer nachfolgenden Geologiemission der Einsatz der wissenschaftlichen Instrumente optimiert werden, wobei z. B. die Routenplanung basierend auf den bereits vorselektierten geologisch relevanten Gebieten erfolgen soll.

2.2.2 Missionsziele

Primäre Ziele

1. Durch die flächendeckende Kartierung und Klassifizierung des Zielgebiets sollen Referenzdaten für die zuverlässige Interpretation von Orbitermessungen gewonnen werden.
2. Mit der Klassifizierung der Boden- und Gesteinsstrukturen soll eine Geologiemission zur großflächigen Analyse der Planetenoberfläche in das Missionsgebiet vorbereitet werden.

Sekundäre Ziele

1. Es soll der Einsatz autonomer Rover zur planetaren Exploration demonstriert werden.
2. Es soll der Einsatz einer Roverflotte als verteiltes Gesamtsystem demonstriert werden.

2.3 Operationsgebiet

Das Lande- und Operationsgebiet für die ExoPlanet-Mission soll die Marsregion der Columbia Hills sein. Für dieses Gebiet existiert umfangreiches Datenmaterial vom Mars Exploration Rover Spirit. Diese Daten können zur Validierung der Missionsergebnisse herangezogen werden, da der Schwerpunkt insbesondere auf der Technologiedemonstration liegt.

Wie Abbildung 2 zeigt, liegt das Gebiet der Columbia Hills innerhalb des Gusev Krater, der einen Durchmesser von 166 km besitzt. Das ExoPlanet-Operationsgebiet wird von einer 2 km×2 km großen Fläche definiert, welche sowohl flaches Gelände mit Kratersenken als auch bergige Zonen aufweist. Der maximale Höhenunterschied innerhalb des ausgewählten Gebietes beträgt ca. 100 m.

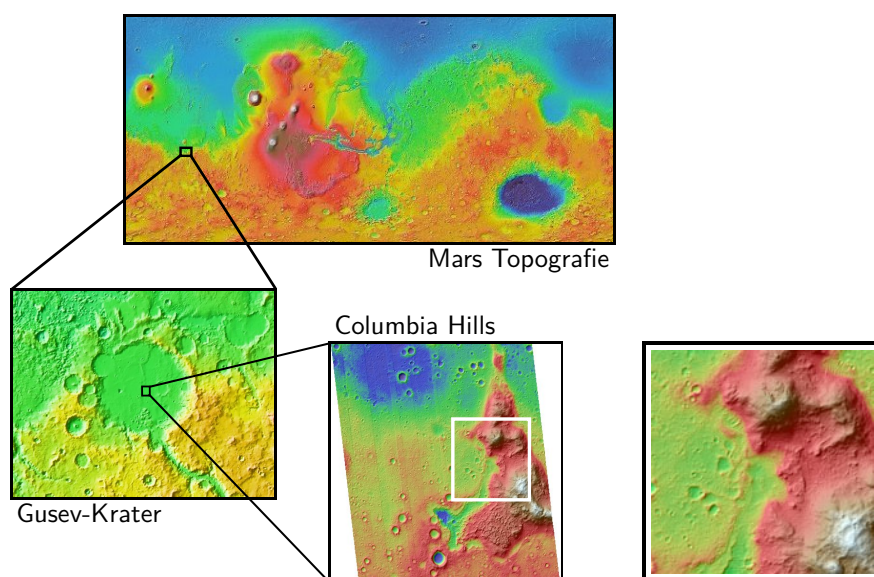


Abbildung 2. Operationsgebiet der ExoPlanet-Mission. Die 2 km×2 km große Region (siehe weiße Markierung) liegt im Zielgebiet des Mars Exploration Rovers Spirit im Gusev Krater. Quelle: eigene Darstellung basierend auf [7] und [38].

2.4 Konzept

Wie bereits im Mission Statement formuliert, ist die Bedeutung der ExoPlanet-Mission in einer größeren Gesamtstrategie zu sehen. So wurden bei den bisher durchgeführten und in naher Zukunft geplanten Rovermissionen nur punktuelle Messungen durchgeführt. Ein Grund ist unter anderem die begrenzte Autonomie und Navigationsleistung. Mit der fortschreitenden technischen Entwicklung werden aber neue Konzepte möglich und gefragt, die eine Alternative zu den bisherigen Ein-Rover-Systemen darstellen.

Aus dem Umstand der limitierten Energieversorgung, zusammen mit dem wissenschaftlichen Ziel einer großen Flächenabdeckung bei gleichzeitiger Risikominimierung, lässt sich die Über-

legung ableiten, Explorationssonden als verteilte Systeme zu konzipieren. Zudem ist es denkbar, auf unterschiedliche Aufgaben (Kartierung, Probenanalyse, Kommunikation) spezialisierte Fahrzeuge entwickeln. Somit könnte z. B. die Erkundung und Kartierung der Umgebung losgelöst von den geologischen Untersuchungen durchgeführt werden. Durch die funktionale Teilung kann zudem gemäß dem Prinzip *Teile und herrsche* die Komplexität für die operationelle Planung reduziert werden. Darüber hinaus gewährleistet der Datenaustausch den zielgerichteten Einsatz der Rover.

ExoPlanet ist diesen Überlegungen folgend der erste Teil einer Mehrfachmission in das gleiche Zielgebiet. Es soll die Marsoberfläche mit Hilfe von Explorationsrovern erfasst werden, wobei neben der Umgebungskartierung die Klassifizierung der Bodenstrukturen im Vordergrund steht. So sollen bereits geologisch relevante Gesteinsstrukturen verortet und Gebiete gefunden werden, die sich für Analysen besonders eignen (kleinere Abhänge oder Risse im Boden, die auf Orbiterbildern nicht zu erkennen sind). In einer Nachfolgemission kommen dann Wissenschaftsrover zum Einsatz, die auf Grundlage der gewonnenen Daten gezielt im Missionsgebiet eingesetzt werden können. Für dieses Scout-Rover-Szenario lassen sich zudem noch weitere grundlegende Konzeptideen ableiten.

Die funktionale Teilung bei mehreren eingesetzten Rover lässt eine Verkleinerung der Fahrzeuge zu. Im Bereich der Orbitalmissionen wurden in den vergangenen Jahren verschiedene Systeme entwickelt, die sich insbesondere durch ein kleines Strukturvolumen und ein geringes Gewicht auszeichnen. Diese Kleinsatelliten konnten im Bereich der Erdbeobachtung (z. B. der Satellit BIRD, entwickelt vom DLR [10] oder die RapidEye Konstellation [11]) erfolgreich demonstrieren, dass durch kompakte Bauweise die Startmasse und damit auch die Projektkosten reduziert werden können.

Die ExoPlanet-Rover sollen sich daher an den Designprinzipien aus der Mikrosatelliten-Entwicklung orientieren. So wird z. B. der MSL-Rover (siehe Abbildung 1b) ein Gewicht von 900 kg aufweisen [27], durch die Verteilung der Funktionen könnte die Masse der Einzelsysteme demgegenüber um den Faktor 10 reduziert werden. Für derartige Mikro-Rover in der Gewichtsklasse 10–100 kg existieren bereits verschiedene Konzepte und Designstudien (dargestellt in Abbildung 3).

Ein besonderer Vorteil der Mikro-Rover ist die Möglichkeit zum gleichzeitigen Start mehrerer Systeme. Durch den gemeinsamen Einsatz der Fahrzeuge im Landegebiet kann ein derartiges Konzept zudem von einer der Haupteigenschaften eines verteilten Systems profitieren – der Risikominimierung bezüglich eines Totalausfalls. Darüber hinaus zeichnet sich eine aus mehreren Sonden bestehende Konfiguration durch eine sehr hohe Agilität aus. Die Instrumente können entweder auf eine bestimmte Region konzentriert oder über ein großes Gebiet verteilt eingesetzt werden. Weitere Vorteile für den Betrieb ergeben sich aus der Möglichkeit der Kooperation. So kann z. B. ein Objekt von verschiedenen Standorten gleichzeitig erfasst werden, was wiederum die Messgenauigkeit verbessern kann oder eine zusätzliche Information für die weitere Operationsplanung darstellt.

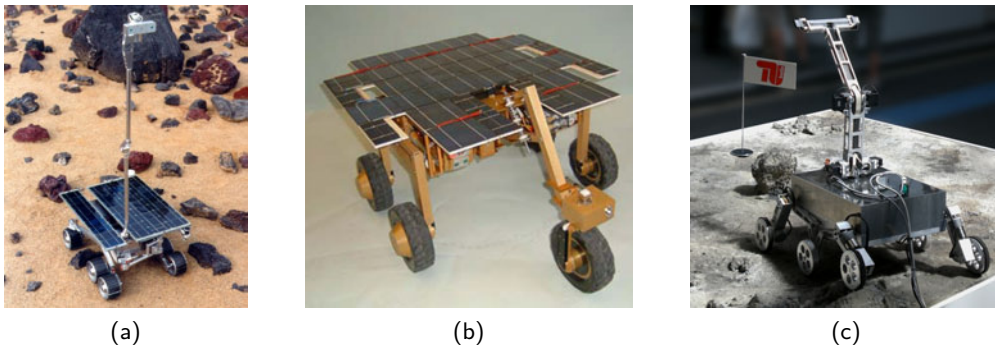


Abbildung 3. Konzepte für planetare Mikro-Rover. (a) Die Testplattform Rocky 7 der NASA. (b) Der ESA Mikro-Rover Solero. (c) Ein Roverkonzept der TU Berlin. Quellen a: [23], b: [Lam08], c: [44]

Das ExoPlanet-Missionskonzept sieht vor, dass mehrere Explorationsrover von einem gemeinsamen Startpunkt ausgehend durch ein zusammenhängendes Gebiet navigieren, um Daten für ein möglichst flächendeckendes Geländemodell zu sammeln. Die Erfassung und Vermessung der lokalen Umgebung erfolgt durch Stereokamera-Systeme. Neben den daraus erstellten Höhenprofilen soll eine Klassifizierung von Hindernissen und Bodenstrukturen die Grundlage für ein autonomes Navigationsverfahren darstellen.

Der Schwerpunkt für die Navigation liegt dabei auf der möglichst überschneidungsfreien Generierung von lokalen Geländemodellen. Das entspricht einer Aufklärungsaufgabe, wobei das erfasste Gebiet eines Rovers nicht die bereits vermessenen Bereiche der anderen Fahrzeuge überdecken darf.

3 Autonome Navigation

Das Forschungs- und Anwendungsfeld Robotik bietet ein sehr breites Spektrum an Problemfeldern und zugehörigen Lösungsansätzen. Insbesondere im industriellen Bereich sind Systeme für die Automatisierung oder Unterstützung unterschiedlicher Tätigkeiten notwendig. Hierbei handelt es sich vor allem um stationäre Prozesse in definierten Umgebungen, wie beispielsweise die Positionierungsaufgaben eines Greifroboters. Darüber hinaus gibt es einen Bedarf an mobilen Plattformen. Sie werden für Operationen in gefährlichen oder unzugänglichen Umgebungen benötigt, aber auch für alltägliche Dienstleistungen, wie ein automatischer Büchertransport in Bibliotheken [48] bis hin zu Staubsaugrobotern. Für die Raumfahrttechnik ist die Entwicklung von robotischen Systemen ebenfalls eine wichtige Komponente für den Erfolg von Missionen. So hat sich für den Bereich der Planetengeologie und -fernerkundung in der Vergangenheit bereits gezeigt, dass sich mit Roversystemen wertvolle Erkenntnisse erzielen lassen.

Betrachtet man die verschiedenen Anwendungen mobiler Systeme, so zeigt sich zugleich ein weiterer Aspekt für den Robotereinsatz: die Autonomie. Der Entwicklungsschritt von einem Automaten mit einer fest programmierten Steuerung bis zu einem autonomen System, welches sich selbst seine Umgebung erschließen und auf Ereignisse reagieren kann, stellt eine große technische Herausforderung dar. Das gilt insbesondere für mobile Systeme, denn die Anforderungen werden um so höher, je unvorhersagbarer die Umgebung wird.

Für die Navigation von mobilen autonomen Systemen soll im Folgenden gezeigt werden, welche allgemeinen Problemstellungen sich abgrenzen lassen. Dazu sollen verschiedene Ansätze und Verfahren vorgestellt werden, die für eine integrierte Lösung notwendig sind. Diese Teilstrategien dienen später als Grundlage für das zu entwickelnde Navigationskonzept für die ExoPlanet-Rover, mit ihrer Aufgabe, möglichst effizient ein unbekanntes Terrain eigenständig zu erkunden. Zunächst soll aber der Begriff der Autonomie genauer eingegrenzt werden, da sich daraus wiederum Anforderungen an das Roversystem bzw. das zu Grunde liegende Konzept ableiten lassen.

3.1 Begriffseinordnung und -definition

Eine präzise Definition des Begriffs Autonomie ist auf Grund seiner weitläufigen Verwendung nicht möglich. Nach einer kurzen Beschreibung der Wortbedeutung wird daher versucht, durch die Charakterisierung des Problemfeldes, das sich für autonome Systeme ergibt, der Bezeichnung eine für diese Arbeit geltende Bedeutung zu geben.

3.1.1 Autonomie

In seiner allgemeinen Bedeutung leitet sich der Begriff Autonomie vom griechischen Wort $\alpha\upsilon\tau\omicron\nu\omicron\mu\iota\alpha$ (autonomia) ab, welches sich zusammensetzt aus den Teilen $\alpha\upsilon\tau\omicron$ (auto: selbst, eigenständig) und $\nu\omicron\mu\omicron\varsigma$ (nomos: Regel, Gesetz). Es kann also im ursprünglichen Sinn übersetzt werden mit *sich selbst Gesetze gebend* und beschreibt damit das Recht griechischer Stadtstaaten, ihre eigenen Gesetze zu erlassen.

Autonom wird in der Robotik im Wesentlichen als Synonym für *unabhängig* verwendet. Es gibt z. B. die autonome Energieversorgung, wenn ein Gerät über Batterien verfügt, mit denen es (kurzzeitig) eigenständig operieren kann, oder die autonome Steuerung, mit der ein System selbständig auf sich ändernde Umgebungsparameter reagieren kann. [Ste95] Im Unterschied zur manuellen Steuerung/Fernsteuerung ist hier im Wesentlichen die Unabhängigkeit des Systems gegenüber externer Überwachung und somit auch gegenüber dem Eingreifen in unbestimmten Situationen gemeint. Ein autonom steuernder Roboter muss also die Fähigkeit besitzen, seine Umgebung zu erfassen, um dann entsprechend seiner Zielvorgaben eigenständig zu agieren.

Neben dem allgemeinen Konsens in der Bedeutung von Autonomie als unabhängiges Agieren eines Systems innerhalb einer sich ändernden Umgebung wird der Begriff in sehr unterschiedlicher Weise interpretiert. Man findet die Bezeichnung für sehr einfache Systeme mit festen Aktionsregeln bis hin zu kooperierenden Agentensystemen mit dynamischen Konfigurationen. Um diesem Problem zu begegnen, wird unter anderem versucht, einen variablen Autonomiebegriff zu finden [Glo04]. Betrachtet wird hierbei der Grad der Autonomie (bestimmt durch den Anteil der externen Anweisungen für eine zu erfüllende Aufgabe), der sich zudem während einer Operation ändern kann. Es wird also im Wesentlichen ein Betriebsmodus beschrieben, der stark von der Zielvorgabe des Systems abhängt. Bei einem Autonomiegrad von 0 % handelt es sich demnach um ein ferngesteuertes (Teleoperation) System.

3.1.2 Autonome Rover

Für einen planetaren Rover sind für die Steuerung hauptsächlich das Navigations- und das Antriebssystem zuständig. Für die Anforderung der autonomen Erkundung des Missionsgebietes ergeben sich nach [Cha96] für diese Subsysteme folgende Aufgaben:

1. Erfassung und Modellierung der Umgebung
Jede Art von Bewegung benötigt mindestens eine Repräsentation der lokalen Umgebung, meistens auch globale Informationen. Der Navigationsprozess muss entscheiden, wo, wann und welche Daten dafür erfasst werden sollen.
2. Lokalisierung
Der Rover muss wissen, wo er sich in seinem Umgebungsmodell befindet und wo sein Ziel verortet ist.

3. Routenplanung

Es muss entschieden werden, wohin und auf welchem Weg sich bewegt werden soll. Einerseits lokal, aber auch langfristig. Wenn möglich, soll dazu eine Bahn bestimmt werden, auf der Hindernissen ausgewichen und unwegsames Gelände gemieden wird.

4. Steuerung

Hinsichtlich der getroffenen Navigationsentscheidung müssen Befehle an das Antriebssystem übergeben werden. Diese sind auszuführen, möglichst mit einer sensorbasierten Kurskontrolle auf Grundlage der eigenen Umgebungsinformationen.

Die angesprochene Variabilität im Grad der Autonomie wird auch im Design von planetaren Rovern deutlich. Sie können einerseits als Befehl ausführen, eine vorberechnete Wegstrecke mit festgelegten Systemparametern (z. B. Geschwindigkeit) zu verfolgen, aber auch nur die Zielvorgabe erhalten, eine entfernte Position zu erreichen. Die Option der Veränderung des Autonomielevels mit unterschiedlichen manuellen Vorgaben ist in der Raumfahrt oft eine wichtige Anforderung. Die Projekte haben eine lange Vorbereitungszeit, sind sehr teuer und die Missionsdauern sind stark begrenzt. Deshalb muss ein maximaler wissenschaftlicher Informationsgewinn angestrebt werden. So wird auch bei bemannten Missionen möglichst jede Handlung im Vorfeld geplant und auf Alternativen geprüft, um die notwendigen Entscheidungen während der Operation auf ein Minimum zu reduzieren. Die Autonomieanforderungen an ein planetares Roversystem sind daher immer ein Kompromiss aus operationellem Aufwand und der Sicherheit und Planbarkeit. Dazu kommt, dass sich bei der Erkundung von unbekannt Gebieten die konkreten wissenschaftlichen Aufgaben meist erst mit der fortschreitenden Datengewinnung ergeben.

Oft gelten technisch aufwendige Systeme nicht gleichzeitig als hochgradig autonome Systeme [Glo04], was sich auch in der angestrebten Semi-Autonomie der Rover widerspiegelt. Auf Grund der Komplexität der freien Geländennavigation wird es schwer sein, an diesem Design in naher Zukunft etwas zu ändern. Dem gegenüber stehen andere Explorationskonzepte, wie z. B. Ballonsysteme. Diese können vollautonom agieren, indem sie mit einer einfachen verhaltensbasierten Steuerung ausgestattet werden [Hon09]. Damit können diese Systeme aber nicht mehr beliebige Randbedingungen für die Navigationsaufgaben berücksichtigen und eine präzise zeitliche Vorhersagbarkeit der Bewegung ist ebenfalls nicht gegeben.

Zusammenfassend können die folgenden Designkriterien für die Konzeption von planetaren Rovern genannt werden (siehe [Was99]). Als Anforderungen an das Gesamtsystem sollten sie entsprechend auf die verschiedenen Subsysteme, von der Entwicklung der Navigationssoftware bis zur Mechanikauslegung, angewendet werden.

Robuste und flexible Funktionsweise

Die Funktion der Systeme darf nicht gefährdet sein, sollten die Umgebungsbedingungen von ihren Idealmodellen abweichen (z. B. durch Lichtspiegelungen, Staubablagerungen oder bei extremen Manövern). Wenn möglich, sollten mehrere alternative Verfahren implementiert werden, die je nach Zielvorgabe und Rahmenbedingungen zu Einsatz kommen.

Ressourcenoptimierter Betrieb

Um die Missionsdauer optimal auszunutzen, sollte das System immer die gesamten zur Verfügung stehenden Ressourcen ausschöpfen. Engpässe müssen erkannt und die Operationsprofile angepasst werden.

Fähigkeit zur Gefahrenerkennung und Notfallplanung

Das Roversystem muss seine Aktivitäten überwachen und beim Abweichen vom Sollzustand oder bei unvorhergesehenen Veränderungen der Umgebungsparameter (z. B. bei Erkennung zusätzlicher Hindernisse) muss eine entsprechende Neukonfiguration stattfinden. In Gefahrensituationen sollte solange in einem Sicherheitsmodus operiert werden, bis die volle Einsatzfähigkeit wiedererlangt ist. Bei Missionsgefährdung sind externe Anweisungen abzuwarten.

3.1.3 Navigationsaufgaben für autonome mobile Systeme

Wie bereits dargestellt, ist es für die autonome Navigation mobiler Systeme notwendig, ein Modell ihrer Umgebung zu generieren, um daraus abgeleitet weitere Aktionsplanungen durchzuführen. Betrachtet man eine allgemeine Navigationsaufgabe, können für diesen Prozess drei wesentliche Teilaufgaben identifiziert werden:

1. Kartenerstellung
2. Lokalisierung
3. Pfad- und Bewegungsplanung

Bei der Kartenerstellung wird aus den Sensorinformationen eine Beschreibung der Umgebung modelliert. Nach [Sta09] lautet die grundsätzliche Frage für diese Aufgabe: *Wie ist die Welt um mich herum beschaffen?* Das Problem der Lokalisierung ist gleichbedeutend mit der Bestimmung der eigenen Position, also: *Wo befinde ich mich?* und die dritte Teilaufgabe beantwortet die Frage: *Wie kann ich ein vorgegebenes Ziel erreichen?* Das letzte Problem ist dabei mit der Aufgabe verbunden, eine Planung für die Wegstrecke durchzuführen, aber auch für die Bewegung, d. h. für die Steuerung des eigenen Antriebssystems.

Ein Hauptmerkmal dieser drei Teilaufgaben ist, dass sie nicht unabhängig voneinander gelöst werden können. Um eine Position bestimmen zu können, benötigt man eine Karte. Für die Erstellung einer Karte muss aber gleichzeitig bekannt sein, an welcher Stelle die Sensordaten erfasst wurden. Gleichmaßen ist auch die Wegplanung abhängig von einem Umgebungsmodell und den Koordinaten des Start- und Zielortes [Sta09].

In Abbildung 4 sind die genannten Teilaufgaben der Navigation dargestellt, die auch in der nachfolgenden Beschreibung der Lösungsansätze teilweise nochmals aufgegriffen werden. Durch diese Grafik wird auch der Zusammenhang zwischen den einzelnen Begriffen deutlich. Es zeigt sich, dass es für die Bereiche zu Überschneidungen kommt und dass sich dadurch noch zusätzliche Problemstellungen ergeben.

- SLAM (simultaneous localization and mapping)

Das SLAM-Problem ergibt sich aus der gegenseitigen Abhängigkeit der Kartenerstellung und der Lokalisierung. Eine genaue Karte setzt die genaue Kenntnis der Position voraus, von der aus die Umgebung erfasst wird. Die Bestimmung des Standorts oder der zurückgelegten Strecke ist aber gleichzeitig nur mit Hilfe eindeutiger Karten möglich. Insbesondere bei auftretenden Messungenauigkeiten werden unkorrigierte Kartendaten durch diese Abhängigkeit immer weiter von der realen Umgebung abweichen.

- Aktive Lokalisierung

Die Routenplanung unterstützt hier die Qualität der Ortung, es überschneiden sich also die Bereiche von Pfadplanung und Lokalisierung. Durch eine günstige Streckenwahl kann z. B. sichergestellt werden, dass immer genug Landmarken im Sichtfeld liegen und daher eine zuverlässige Lokalisierung möglich ist¹.

- Exploration

Hierunter wird die Aufgabe verstanden, Daten über die Umgebung zu sammeln, wozu ist eine systematische Routenplanung notwendig ist. Einen möglichen Ansatz für diese Aufgabenstellung stellen einfache regelbasierte Navigationsverfahren dar (z. B. fahre immer an einer Wand entlang). Für komplexe Umgebungen wird schnell deutlich, dass neben der Routenplanung und der Kartenerfassung auch die Roverposition notwendig ist. Es ergibt sich dann ein integriertes Navigationsproblem aus allen Teilbereichen.

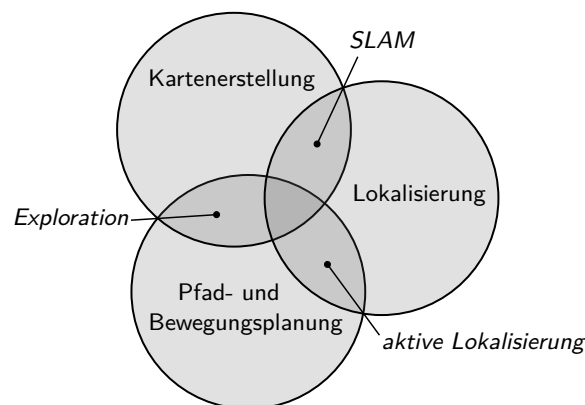


Abbildung 4. Allgemeiner Zusammenhang zwischen den unterschiedlichen Navigationsaufgaben.
Quelle: eigene Darstellung nach [Mak02].

Für die autonome Navigation sind meist integrierte Verfahren notwendig, d. h., es muss ein Lösungsansatz gefunden werden, der alle Teilaufgaben berücksichtigt. Bei der Konzeption für planetare Rover spricht man daher auch von einer *Explorationsstrategie*. Darunter werden alle Schritte zusammengefasst, die für die sichere Navigation unter einer gegebenen Zielvorgabe notwendig sind. Im Folgenden werden dazu einige Methoden und Verfahren vorgestellt, die später die Grundlage für die Entwicklung eines eigenen Navigationskonzepts darstellen. Zudem zeigen sie den Stand der Technik auf diesem Gebiet.

¹Ein Beispiel hierfür ist die historische Seenavigation, bei der versucht wurde, entlang der Breitengrade oder Küsten zu segeln, anstelle der direkten Route zu folgen.

3.2 Verfahren als Grundlage für die autonome Navigation

3.2.1 Kartenerstellung

Die Hauptaufgabe für die Kartenerstellung ist die Erfassung der Umgebung. Dafür existieren aktive und passive Verfahren. Eine in der Innenraumnavigation verbreitete Methode ist die Laserdistanzmessung. Hierbei wird aktiv die Entfernung von Objekten ermittelt und in einem Referenzsystem dargestellt. Als passiver Ansatz werden meist photogrammetrische Verfahren, wie die Stereo-Bildverarbeitung, eingesetzt. Diese Methode ist besonders für planetare Rover geeignet und soll deshalb exemplarisch beschrieben werden.

Das sogenannte maschinelle Stereosehen (Stereo Vision) beruht, wie auch das menschliche Sehen, auf der Abbildung einer Szene durch zwei oder mehr räumlich getrennte Kameras. Aus den verschiedenen Aufnahmen wird ein Tiefenbild generiert (d. h. möglichst jedem Pixel wird der Abstand zur Kamera zugeordnet), indem zunächst einzelne markante Bildpunkte extrahiert werden. Für diese kann dann abgeleitet von den Kamerapositionen und den bekannten Abbildungsparametern eine räumliche Koordinate bestimmt werden. Im Mittelpunkt des Verfahrens steht daher das Zuordnen von korrespondierenden Merkmalen, also den Punkten, die in den unterschiedlichen Projektionen dieselbe Struktur einer Szene repräsentieren. [Jia97]

Die Mars Exploration Rover verfügen zur Navigation und als wissenschaftliche Instrumente über mehrere Stereokamerasysteme. Darunter eine Navigationskamera am Rovermast zur Umgebungserfassung und zwei Hazard-Avoidance Kameras (Hazcams) zur Hinderniserkennung direkt vor den Rädern. Aus den gelieferten Stereobildern, wie sie in den Abbildungen 5a und b für eine Hazcam dargestellt sind, werden wie beschrieben Tiefenbilder generiert und die räumlichen Koordinaten der Umgebung abgeleitet. Basierend darauf kann dann z. B. der Geländeanstieg bestimmt werden (Abbildung 5c), was eine entscheidende Information für die Navigationsverfahren darstellt, da die Befahrbarkeit der Umgebung davon abhängt. Für detaillierte Informationen über die MER-Kameras siehe auch [Bel03; Mak03].

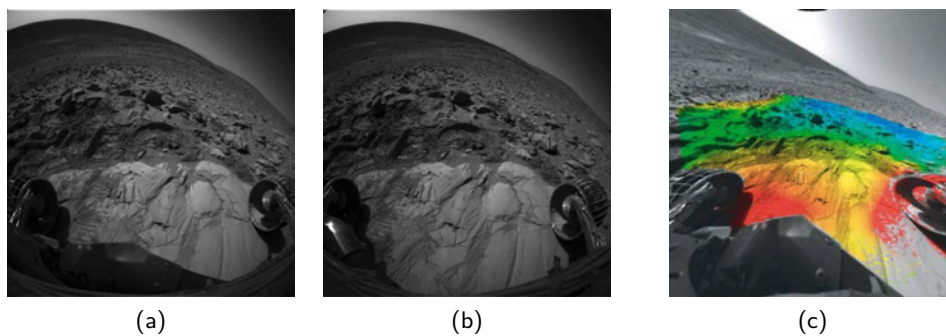


Abbildung 5. Stereobild der vorderen MER-Hazcam. (a, b) Die Rohbilder der linken und rechten Kamera. (c) Aus den Tiefeninformationen wurde das Gefälle des Geländes bestimmt und auf ein linearisiertes Teilbild projiziert (blau bedeutet flach, rot sind steile Bereiche). Quellen a-c: [Ale06].

Die Ergebnisse der Stereobildverarbeitung dienen jedoch nicht nur der Rovernavigation, sie können auch genutzt werden, um digitale Geländemodelle zu erzeugen. Durch eine Kombination dieser Modelle mit den Kamerabildern ist die Visualisierung der dreidimensionalen Roverumgebung, insbesondere zur Unterstützung der technischen und wissenschaftlichen Missionskontrolle, möglich. Abbildung 6a zeigt ein Bild der MER-Panoramakamera. Abgeleitet aus den Tiefeninformationen können beliebige Projektionen, z. B. bezüglich der Roverposition (Abbildungen 6b und c), dargestellt werden.

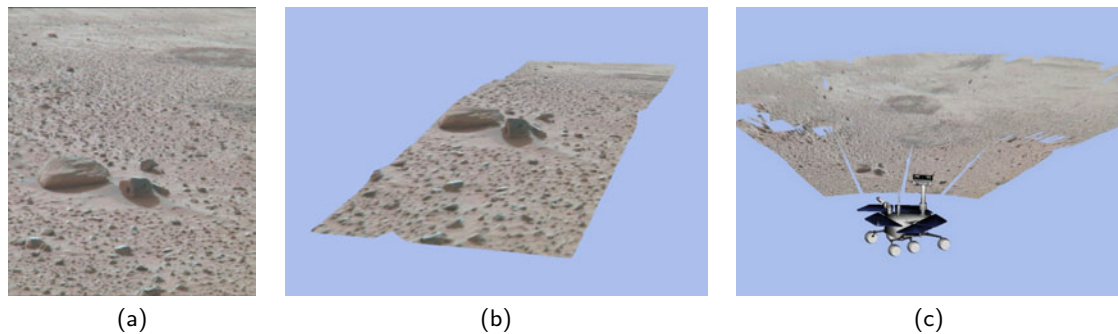


Abbildung 6. Bildprojektion basierend auf Tiefeninformationen. (a) Eine Teilaufnahme der MER-Panoramakamera. (b) Die Projektion des Bildes basierend auf den Tiefeninformationen. (c) Mehrere räumlich zur Roverposition referenzierte Aufnahmen. Quellen a–c: eigene Darstellung mit Hilfe der Software *Maestro* (siehe [6]).

Neben der geometrischen Erfassung der Umgebung auf Grundlage der Sensordaten besteht eine weitere Fragestellung darin, wie die Geländedaten durch ein Kartenmodell repräsentiert werden können. Hierzu existieren zwei wesentliche Ansätze: das metrische und das topologische Modell. Metrische Karten beziehen sich auf ein einheitliches Koordinatensystem, in dem alle Objekte verortet sind. Dies können sowohl Raster- als auch Vektorkarten sein. Ein Vorteil dieser Darstellung ist, dass sie direkt von Algorithmen zur Routenplanung verwendet werden können, da Entfernungen und Abstände unmittelbar bestimmbar sind.

Demgegenüber stellen topologische Modelle die Beziehungen zwischen Objekten dar. Oft werden die zu kartierenden Strukturen bezüglich markanter Standorte (Landmarken) beschrieben und stark abstrahiert (z. B. lediglich durch ihre Eckpunkte charakterisiert) dargestellt. Während metrische Karten in der Regel durch Raster bzw. Matrizen repräsentiert werden, ist es bei topologischen Karten meist ein Graph, der als Datenstruktur verwendet wird. Ein Beispiel dafür sind Netzpläne von U-Bahnen. Diese entsprechen auch der menschlichen kognitiven Abbildung einer Umgebung. Um ein Ziel zu finden, muss nicht der geometrisch genaue Streckenverlauf bekannt sein, sondern nur eine grobe Richtung und die Information, an welcher Stelle abzubiegen bzw. umzusteigen ist. Die Vorteile einer topologischen Karte sind zum einen ihre Kompaktheit und zum anderen die Möglichkeit einer auf den jeweiligen Anwendungszweck angepassten Darstellung.

Je nach Problemstellung eignet sich eine metrische oder topologische Repräsentation der Umgebung besser. Darüber hinaus werden auch kombinierte Modelle verwendet. So ist ein hierarchischer Ansatz möglich, der auf einer topologischen Beschreibung von großräumigen Gebieten und der metrischen Abbildung der direkten Umgebung beruht (siehe z. B. [Thr96] und [Pon02]). Dabei können die topologischen Beziehungen aus den Sensordaten abgeleitet werden, die in der Regel metrische Informationen in Form von lokalen Koordinaten liefern.

3.2.2 Lokalisierung

Das Problem der Lokalisierung besteht darin, die Position und die Ausrichtung des Rovers zu bestimmen. Diese Aufgabe kann unterteilt werden in die absolute und die relative Lokalisierung. Bei der absoluten Positionsbestimmung müssen die Roverkoordinaten bezüglich eines globalen Referenzsystems ermittelt werden. Dagegen wird bei der Positionsverfolgung über Wegmessungen (Odometrie) versucht, die relative Koordinatenänderung zwischen zwei Standorten zu bestimmen. Bei diesem auch für die Koppelnavigation verwendeten Verfahren werden die Positionsänderungen aufsummiert und es ergibt sich daraus eine Information über den aktuellen Standort. Jedoch summieren sich auch die Messfehler, so dass zeitweise durch eine absolute Lokalisierung das Bezugssystem aktualisiert werden muss.

Eine absolute Referenzierung ist z. B. mit Hilfe eines Sternensensors oder durch die Erfassung des Sonnenstandes möglich, wobei zu berücksichtigen ist, dass Sternensensoren nur nachts eingesetzt werden können, wenn keine Solarenergie zur Verfügung steht. Außerdem erhält man auf Grund der erforderlichen Genauigkeit nur die Information über die Ausrichtung des Fahrzeugs, die absolute Roverposition lässt sich in der notwendigen Auflösung nicht aus diesen Daten ableiten. Jedoch ist eine präzise Richtungsbestimmung entscheidend für die Genauigkeit der relativen Bewegungsschätzung [Tre01; Lam08].

Sind für das Missionsgebiet hoch aufgelöste Orbiteraufnahmen verfügbar, so kann auch eine Lokalisierung durch Landmarkenortung erfolgen. Abbildung 7a zeigt ein Kamerabild des Rovers Opportunity. Die sichtbaren Hügel konnten den entsprechenden Erhebungen in einer Orbiteraufnahme (Abbildung 7b) zugeordnet werden. Dadurch ist es möglich, in etwa den Standort und wiederum sehr genau die Blickrichtung des Rovers zu bestimmen. Unter Landmarken versteht man im Sinne der visuellen Lokalisierung markante Punkte oder Strukturen in den zwei- oder dreidimensionalen Abbildungen der Umgebung (Bergkuppen, Kraterränder, Steine, Bodenschattierungen, etc.).

Ein anderes Verfahren zur absoluten Lokalisierung ist der Mustervergleich (Matching) der projizierten Bilder aus der direkten Roverumgebung (vgl. Abbildung 6) mit Orbiterfotos. Meist werden aus den Bildern charakteristische Strukturen wie Steine extrahiert und für den Bildvergleich herangezogen, um das Matchergebnis zu verbessern [Kar10; Wan08]. Insgesamt lassen sich mit dieser Methode die absolute Position und Ausrichtung sehr präzise bestimmen. Ist

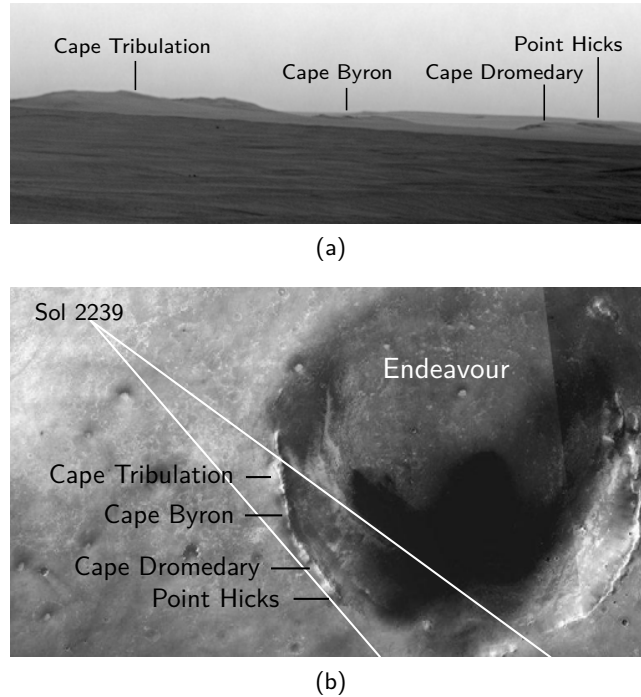


Abbildung 7. Lokalisierung mit Hilfe von Landmarken. (a) Der Rand des Kraters Endeavour aus Sicht des Rovers Opportunity am 12. Mai 2010. (b) Aufnahme des Kraters vom Mars Reconnaissance Orbiter mit eingezeichnetem Blickwinkel des Rovers. Quellen a: [36], b: [37].

zudem ein digitales Höhenmodell vom Missionsgebiet verfügbar, so können die 3D-Geländekoordinaten ebenfalls in den Matchingprozess einfließen. Damit ergibt sich eine Invarianz gegenüber Veränderungen der sichtbaren Strukturen z. B. durch unterschiedliche Schatten und Beleuchtungsverhältnisse. Für die ausführliche Beschreibung eines derartigen Ansatzes siehe [Lin10].

Wie bereits erwähnt, können während der Navigation die zurückgelegte Wegstrecke und Richtungsänderung gemessen und die Positionsdaten darauf basierend aktualisiert werden. Ein möglicher Ansatz dafür ist die Erfassung der Drehwinkel der Räder. Für eine hohe Genauigkeit muss bei diesem Verfahren zudem die Untergrundbeschaffenheit berücksichtigt werden (für eine Schlupfkorrektur). Zusätzlich kann auch die Antriebssteuerung für das jeweilige Gelände optimiert werden, um bessere Odometrieergebnisse zu erhalten [Lam08]. Außerdem hat die Schiefelage des Fahrzeugs einen Einfluss auf die Streckenberechnung. Sogenannte Inertialsensoren (Inertial Measurement Units IMU), die im Wesentlichen aus mehreren Beschleunigungs- und Drehratensensoren bestehen, liefern diese Information jedoch sehr genau.

Eine weitere Methode für die Positionsverfolgung ist die visuelle Odometrie. Dieses Verfahren, welches auch bei den Mars Exploration Rovern zur Navigation eingesetzt wird, basiert auf dem Vergleich der Stereoaufnahmen von verschiedenen Standpunkten aus. Indem einzelne markante Stellen, deren Geometrie durch die Stereoanalyse bekannt ist, in den Bildern einander zugeordnet werden, kann die Positionsänderung des Rovers bestimmt werden. Diese Daten können dann mit den Ergebnissen der Rad-Odometrie kombiniert werden, um eine Positionsdrift von

unter 10 % bei 100 m zurückgelegter Strecke zu erreichen. Insbesondere bei großen Anstiegen, sandigem Boden oder bei der Überwindung von größeren, teilweise losen Steinen ist diese Genauigkeit durch die Rad-Odometrie allein nicht zu gewährleisten (siehe dazu [Mai07]). Außerdem ist durch die visuelle Auswertung der aktuelle Schlupf der Räder bestimmbar, wodurch z. B. ungünstige Bodenbeschaffenheiten und damit Gefahrenzonen erkennbar sind.

Werden die Aufnahmen und Tiefenkarten von verschiedenen Standorten gespeichert, so ist es möglich, diese untereinander zu referenzieren (ein Verfahren, welches diesen Ansatz verfolgt, ist z. B. das Bundle Adjustment). Als Ergebnis erhält man ein lokales Geländemodell, und die jeweils neuen Positionsdaten können in dieses Geländemodell eingepasst werden, was eine zusätzliche Korrekturmöglichkeit für die relative Lokalisierung darstellt. Da das genannte Bundle Adjustment ein sehr rechenintensives Verfahren ist, werden die Bilddaten auf der Erde ausgewertet (im Gegensatz zur visuellen Odometrie, die in nahezu Echtzeit onboard durchgeführt wird) [Kar10]. Die Korrekturen können aber rückwirkend auf die Positionsverfolgung angewendet werden. Abbildung 8 zeigt ein digitales Geländemodell (Digital Terrain Model DTM), welches mit diesem Verfahren erzeugt wurde, zusammen mit der daraus resultierenden topographischen Karte.

Auf Grundlage einer sehr genauen Odometrie existiert ein weiteres Verfahren für die Referenzierung des Pfades: das Wide Baseline Mapping. Die Aufnahmen des Horizontes von zwei verschiedenen Standorten aus, wobei die dazwischen liegende Entfernung und Richtungsänderung genau bekannt sind, können als Stereobildpaar für eine Tiefenanalyse verwendet werden. Bei einer Basislinie von etwa 8 m zwischen den Aufnahmeorten ist eine Abstandsauflösung im Meterbereich bis zu einer Entfernung von 400 m möglich (im Unterschied zur MER-Panoramakamera mit einem Messfehler von über 1 m bereits ab 60 m Entfernung) [Che08]. Mit Hilfe dieser Methode können sowohl die Ergebnisse der Lokalisierung als auch die topographischen Karten verbessert werden.

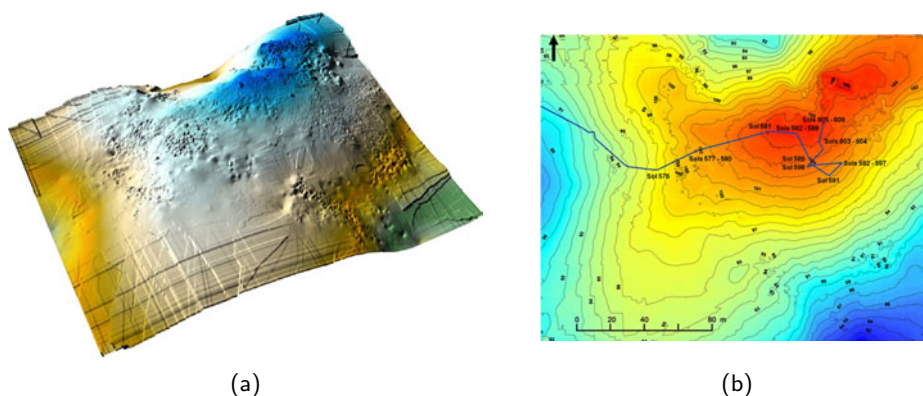


Abbildung 8. Lokales Geländemodell der Roverumgebung. Zu sehen ist ein Gebiet um Husband Hill (ca. 200×200 m) aufgenommen vom Mars Rover Spirit. (a) Das Höhenmodell als Ergebnis der Stereoanalyse von mehreren Standorten aus. (b) Eine topographische Karte, abgeleitet aus dem lokalen DTM. Quellen a, b: [Di08].

3.2.3 Pfadplanung

Die dritte wesentliche Aufgabe für ein autonomes Navigationsverfahren, neben der Kartenerstellung und der Lokalisierung, ist die Pfadplanung. Darunter fallen zum einen die Routenplanung, aber auch die konkrete Planung der Antriebssteuerung, also die jeweilige Bewegungsplanung zur Routenverfolgung. Allgemein lässt sich das Problem der Pfadplanung mit der folgenden Aufgabe charakterisieren: *Finde den für die geltenden Randbedingungen optimalen Pfad zwischen zwei gegebenen Roverkonfigurationen.*

Der Begriff der Konfiguration bezeichnet in der Robotik einen Zustandsvektor des Systems über seine Freiheitsgrade, z. B. jeweils 3 Koordinaten für die Position und die Ausrichtung. Sind zusätzliche kinematische Elemente (wie ein Greifarm) auf dem Roboter zu berücksichtigen, erhöht sich die Dimension des Konfigurationsvektors entsprechend.

Für die Navigation lassen sich zunächst die globale und die lokale Pfadplanung unterscheiden. Ein von der NASA festgelegtes Ziel für die Technologieentwicklung im Rahmen der Mars erkundung ist die Fähigkeit der autonomen Navigation der Rover bis zu 300 m pro Missionstag [Hay05]. Eine Routenplanung auf diese Distanz erfolgt zunächst global, indem geeignete Wegpunkte bestimmt werden. Als Basis für die Wegpunktbestimmung können die vom Rover erstellten Geländemodelle dienen (siehe Abbildung 8) oder verfügbare topographische Orbiterkarten². Die Höhenprofile sind auf Grund der Messmethoden meist stark interpolierte Modelle, kleine Hindernisse oder die verschiedenen lokalen Geländeparameter (wie Unebenheiten, sandige Bereiche oder Geröll) bleiben darin unberücksichtigt.

Abbildung 9a verdeutlicht das Prinzip der Pfadplanung. Für ein vorgegebenes Ziel werden mehrere Wegpunkte einer Route ermittelt, die bereits um unwegsames Gelände und Gefahrenzonen herumführt. Der Rover versucht, im Wesentlichen diesem Pfad zu folgen. Gleichzeitig werden jedoch auch lokale Sensordaten ausgewertet und sollte ein Hindernis auftauchen, so wird eine Pfadkorrektur zur Umfahrung dieser Stellen durchgeführt. Für diese lokale Pfadplanung wurde für die MER-Mission der Algorithmus GESTALT (Grid-based Estimation of Surface Traversability Applied to Local Terrain) entwickelt [Mai06]. Dieses Verfahren wertet die Stereobilder der Hinderniskameras (vgl. Abbildung 5) aus, um die Befahrbarkeit des Untergrundes zu klassifizieren. Wie Abbildung 9b zeigt, wird dazu das Gelände durch ein Raster repräsentiert, in dem jeder Zelle ein Skalenwert, bezüglich des Anstieges, der Unebenheit, etc. zugeordnet wird. Anschließend können verschiedene Ausweichrouten und -manöver innerhalb des Geländerrasters ermittelt werden.

Allgemein geht es bei der Pfadplanung darum, eine sogenannte Kostenfunktion zu minimieren. Als Kostenfaktoren in diesem Sinne gelten die verschiedenen Parameter, die auf die Navigation einen Einfluss haben, also die Geländebeschaffenheit, der Aufwand bezüglich der notwendigen

²Zum Start der MER-Mission im Jahr 2003 waren von den Operationsgebieten Höhenmodelle auf Grundlage von MOC-NA Stereobildern vorhanden. Diese hatten eine Rasterauflösung von 10 m bei einer Höhenauflösung von 2 m [Li06]. Erst seit Ende 2006 stehen mit den MRO HiRISE Daten genauere Geländemodelle mit einem Raster von unter einem Meter zur Verfügung [McE07].

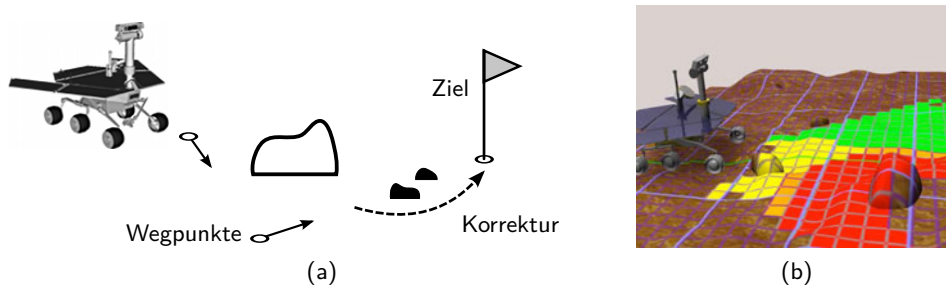


Abbildung 9. Globale und lokale Pfadplanung. (a) Selektion von globalen Wegpunkten für die autonome Zielerreichung. (b) Klassifizierung des Geländes zur lokalen Hindernisvermeidung. Quellen a: eigene Darstellung nach [Mai06], b: [Mai06].

Manöver oder die Länge der Strecke. Aber auch wissenschaftliche Randbedingungen, wie die Erfassung möglichst vieler geologisch interessanter Gesteinsformationen auf der geplanten Route oder technische Aspekte, wie die Ausrichtung zur Sonne und damit die Aufrechterhaltung der Energieversorgung während der Fahrt, können hier eine Rolle spielen. Bezüglich dieser Kriterien wird dann für jeden Routenabschnitt eine Bewertung vorgenommen. Anschließend werden die einzelnen Parameter gewichtet und die resultierenden Werte aufsummiert. Für die weitere Planung wird dann ein Pfad gewählt, bei dem dieser Gesamtkostenwert möglichst gering ist. Das Optimum für eine Kostenfunktion variiert insbesondere mit der vorgenommenen Gewichtung der verschiedenen Parameter. Abbildung 10a zeigt in einem Geländemodell zwei Pfadalternativen, die unter Berücksichtigung unterschiedlicher Bewertungskriterien als Optimum gefunden wurden.

Die Grundidee bezüglich der Minimierung einer Kostenfunktion ist äquivalent für die globale und lokale Routenplanung. Der Unterschied besteht darin, dass verschiedene Parameter und Randbedingungen in die Pfadbewertung einfließen. So wird z. B. der Rover für die globale Planung meist als punktförmig betrachtet, für eine lokale Pfadkorrektur sind die Geometrie des Fahrzeugs und seine Manövrierfähigkeit jedoch sehr wichtig. In Abbildung 10b ist als schattierte Karte ein Testfeld für eine Marsumgebung von ca. $30\text{ m} \times 60\text{ m}$ dargestellt. Die grüne Kurve ist das Ergebnis einer globalen Pfadsuche zwischen dem Startpunkt und einem gegebenen Ziel. Als triangulierte Fläche ist das Geländemodell auf Grundlage der Roversensorik eingezeichnet. Anhand dieser lokalen Topographie wurde ein Alternativpfad gefunden (blau), der eine Korrektur der globalen Route darstellt (der eingezeichnete gelbe Kreis stellt den maximalen lokalen Planungshorizont dar).

Die Schwierigkeit bei der Auswertung der Pfadkosten liegt insbesondere bei der Gewichtung der Randbedingungen, da mitunter auch widersprüchliche Parameter berücksichtigt werden müssen. So ist z. B. ein wichtiges Sicherheitskriterium aus technischer Sicht, unwegsames Gelände zu meiden. Für die geologische Forschung sind jedoch Abhänge und Felsformationen besonders interessant. Eine Möglichkeit ist es daher, die Wegpunkte auch an schwieriges Gelände heranzuführen und wenn notwendig, lokale Korrekturen oder auch eine dynamische Neuplanung durchzuführen.

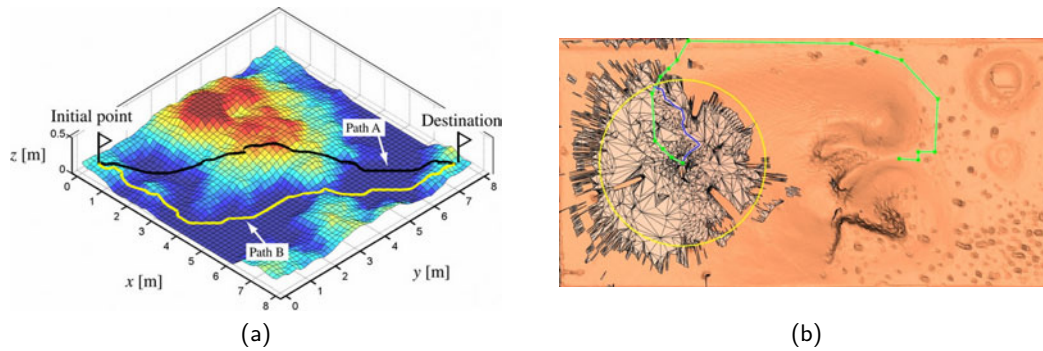


Abbildung 10. Einfluss der Kostenfunktion auf die Pfadplanung. (a) Je nach Gewichtung der Kostenfaktoren ergeben sich verschiedene optimale Wegstrecken. (b) Für globale und lokale Pfadplanung werden unterschiedliche Kriterien berücksichtigt, dadurch ergibt sich lokal ein anderes Optimum (blau) für die Zielführung als global (grün). Die Route kann dementsprechend abschnittsweise angepasst werden. Quellen a: [Ish07], b: [Dup08].

Neben der Unterscheidung in globale und lokale Routenplanung hängt das Problem der Routenplanung und damit der Formulierung einer geeigneten Kostenfunktion von den konkreten Anforderungen der Navigationsaufgabe ab. Eine spezielle Problemstellung ist hierbei z. B. die visuelle Zielverfolgung (Target Tracking). Eine Anwendung dafür ist die Positionierung des Rovers für eine wissenschaftliche Untersuchung. Innerhalb einer Entfernung von ca. 10 m wird auf Grundlage der Kamerabilder ein Objekt vorgegeben und die Navigationsaufgabe für den Rover ist es, sich selbständig diesem Ziel zu nähern und relativ dazu auszurichten [Kim09].

Wird der gesamte Prozess der Pfadplanung von der Auswahl der Wegpunkte über die lokale Korrektur der Route bis zur Steuerung vom Rover übernommen, so spricht man auch von einer Over-The-Horizon Navigation, da die Zielvorgabe auch außerhalb der Sensorreichweite des Fahrzeugs liegen kann [Dup08]. Die Rover zukünftiger Missionen, wie ExoMars oder des Mars Science Laboratory, sind für Tagesstrecken von mehreren 100 m ausgelegt, dementsprechend müssen auch die Navigationsverfahren dahingehend angepasst werden. Für eine Beschreibung von weiteren möglichen Anforderungen und praktischen Methoden für die Pfadplanung für planetare Rover siehe [Tom05]. Eine umfassende Übersicht über Pfadplanung im Allgemeinen gibt [LaV06].

3.2.4 Exploration

Bereits zu Beginn dieses Abschnitts wurde dargestellt, dass innerhalb der verschiedenen Problemstellungen zur Navigation auch Überschneidungen auftreten (siehe Abbildung 4). Als eines dieser integrierten Anwendungen soll an dieser Stelle auf den Bereich Exploration eingegangen werden, da dieser für die spätere Konzeptentwicklung eine Rolle spielt.

Unter Exploration versteht man die Aufgabe, unter verschiedenen Randbedingungen durch ein Gelände zu navigieren, ohne dass ein konkretes Streckenziel vorgegeben ist. Eine Beispielanwendung hierfür ist die Kartierung einer unbekannteren Umgebung. Aber auch alle Aufgaben-

stellungen, bei denen Objekte in einem vorgegebenen Gebiet zu suchen sind, erfordern die strukturierte Erfassung der Gesamtumgebung. Die Explorationsstrategien hängen dabei eng mit den eingesetzten Verfahren zur Kartenerstellung zusammen. So existieren unterschiedliche Ansätze, je nachdem ob ein metrisches oder topologisches Kartenmodell verwendet wird.

Für die Erkundung von Innenräumen bzw. 2D-Umgebungen existieren bereits verschiedene Verfahren. Allgemein handelt es sich um eine Navigationsaufgabe, für die nur Daten über die jeweils lokale Umgebung zur Verfügung stehen. Ein weit verbreiteter Ansatz ist dabei, die Grenzbereiche zu unbekannte Regionen immer weiter aufzulösen, was sich insbesondere für metrische Kartenmodelle eignet. Die Idee dieser als Frontier-Based-Exploration bezeichneten Strategie geht zurück auf [Yam97], wobei bereits eine Vielzahl modifizierter Verfahren entwickelt wurde, z. B. um den Ansatz auch für verteilte Systeme zu verwenden. Abbildung 11a zeigt das Ergebnis einer Arbeit, bei der drei mobile Roboter eingesetzt wurden, um einen Innenraumbereich ohne Vorkenntnisse zu erfassen.

Eine andere grundlegende Explorationsstrategie, die eng mit der topologischen Abbildung der Umgebung verbunden ist, beruht auf dem Prinzip, solange einer einfachen Pfadbeschreibung zu folgen (z. B. im Abstand von 50 cm an einer Wand entlang) bis sich mögliche Verzweigungspunkte ergeben. Diese Verzweigungs- bzw. Kreuzungspunkte und deren Verbindungen untereinander genügen, um eine Beschreibung der gesamten Umgebung abzuleiten. Ziel der Exploration ist es, jede mögliche Richtung mindestens einmal auszuwählen und den Verlauf zu überprüfen. In Abbildung 11b ist die zurückgelegte Strecke für ein derartiges Verfahren abgebildet.

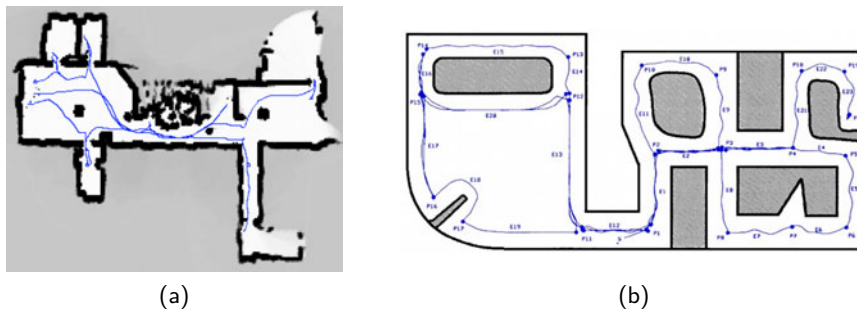


Abbildung 11. Innenraum-Exploration. Mit Hilfe von autonomen mobilen Systemen und ohne Vorkenntnisse über die Umgebung soll der gesamte Raum erfasst werden. Die blauen Kurven zeigen die jeweils zurückgelegte Strecke. (a) Ein Verfahren basierend auf einem metrischen Umgebungsmodell bei drei eingesetzten Erkundungsrobotern. (b) Eine Explorationsstrategie mit einem zu Grunde liegenden topologischen Modell. Quellen a: [Sim00], b: [Kui91].

Darüber hinaus finden sich in der Literatur noch viele weitere Verfahren, wie z. B. die Familie der *Bug*-Algorithmen [LaV06; Cho05] oder Methoden, die sich vom sogenannten Verfolger- bzw. Aufklärungsproblem (Pursuit-evasion) ableiten und teilweise auch auf das freie Gelände übertragbar sind [Kol10]. Insbesondere diese Erweiterung der Methoden für die Anwendung in einer dreidimensionalen Umgebung ist mit einigen Schwierigkeiten verbunden.

Die Verfahren für die Innenraumnavigation lassen sich im Allgemeinen nicht ohne Anpassungen für das freie Gelände verwenden. Dafür gibt es verschiedene Gründe. Einerseits sind die Umgebungen auf Grund ihres Detailreichtums schwer zu modellieren, was aber die Voraussetzung dafür ist, eine Route zu definieren und dieser zu folgen. Dazu kommt, dass der Begriff von Hindernissen in einem freien Gelände nicht klar definiert ist. Was als Hindernis gilt, hängt stark von der Manövrierfähigkeit des Rovers ab, so kann selbst flacher Untergrund ein Problem darstellen, wenn das Fahrzeug darauf ein schlechtes Traktionsverhalten zeigt [Iag04].

Dennoch existieren verschiedene Konzepte für die Exploration im Außenbereich, wobei sich hier auch Parallelen zwischen den Aufgaben von Raumfahrtmissionen und terrestrischen Anwendungen ergeben. Ein Beispiel für eine notwendige Geländeerkundung mit autonomen Systemen auf der Erde ist die Katastrophenhilfe, also die Kartierung des Umgebungszustandes nach Zerstörungen oder die Suche nach Verletzten (siehe z. B. [Kle07]). Eine wissenschaftliche Aufgabenstellung ist die Suche nach Meteoriten in der Antarktis [Ped00]. Insgesamt zeigt sich für die eingesetzten Algorithmen eine zunehmende Komplexität durch den Übergang von einem zweidimensionalen zu einem dreidimensionalen Umgebungsmodell. Insbesondere für die begrenzten Berechnungsressourcen auf den Robotersystemen kann dies ein Problem darstellen. Daher ist es wichtig eine leistungsfähige Simulationsumgebung zu entwickeln, um verschiedene Verfahren zu testen, zu vergleichen und zu optimieren.

3.3 Methoden zur Pfadsuche

Wie bereits beschrieben, ist die Pfadplanung eine wichtige Grundlage für die autonome Navigation. Basierend auf den Verfahren zur Kartenerstellung und Roverlokalisierung gilt es, eine Route auszuwählen, die unter den gegebenen Randbedingungen ein Optimum darstellt. Eine Umsetzung dessen ist die Formulierung einer Kostenfunktion, deren Ergebnis für einen guten Pfad möglichst gering ausfallen sollte.

Da für die Minimierung der Kostenfunktion sehr viele Pfadalternativen ausgewertet und verglichen werden müssen, ist ein naheliegender Ansatz, das Problem als eine Graphensuche darzustellen. Ein Graph wird dabei als eine Struktur aus Knoten und Kanten begriffen (die Kanten repräsentieren die Verbindungen zwischen den Knoten und es werden ihnen als Wert die Streckenkosten zugeordnet). Für die Navigationsaufgaben muss daher zunächst eine Möglichkeit gefunden werden, ein Geländemodell zusammen mit der Kostenfunktion in eine Graphenstruktur abzubilden. Dafür existieren im Wesentlichen zwei Verfahren: die Zelldekomposition (Cell Decomposition) und die Roadmap-Methode [Lat90; Cho05].

Bei der Zelldekomposition wird das Umgebungsmodell im einfachsten Fall auf ein metrisches Raster abgebildet (vgl. Abbildung 10a). Die einzelnen Zellen stellen dabei die Knoten eines Graphen dar. Die Kanten des Graphen ergeben sich aus den Verbindungen zwischen benachbarten Feldern. Die Zellenstruktur muss allerdings kein gleichförmiges Raster sein. Auch unterschiedliche Feldgrößen (sogenannte Quad-Trees) oder triangulierte Flächen, wie die lokale

Umgebung in Abbildung 10b, sind möglich. Darüber hinaus besteht eine weitere Möglichkeit darin, zunächst verschiedene Geländebereiche zu segmentieren. Die Navigation erfolgt dann zwischen den als befahrbar eingestuften Gebieten (die jeweils einen Knoten eines Graphen darstellen).

Wie bereits auf Seite 21 kurz aufgegriffen, geht es bei der Beschreibung mittels Konfigurationen darum, alle möglichen geometrischen Zustände zu erfassen. Das Roadmap-Verfahren ist eine Methode der Pfadplanung, die versucht, eine Verbindung des Start- und Zielzustandes direkt innerhalb des Konfigurationsraums zu finden. Dabei ergibt sich ein sehr komplexes Netzwerk aus den einzelnen möglichen Übergängen zwischen den Konfigurationen. So ändert eine Fahrt geradeaus oder eine Wende jeweils einzelne Werte des Zustandsvektors. Das Transitionsnetz kann als Graph aufgefasst werden und die Aufgabe besteht darin, eine möglichst optimale Bewegungsabfolge zu ermitteln. Um den hochdimensionalen Konfigurationsraum und damit die Anzahl der Pfadmöglichkeiten zu reduzieren, werden die Umgebungsmodelle zunächst auf wenige essenzielle Eigenschaften (wie Sichtlinien, Hinderniskonturen, etc.) reduziert und als topologische Karte dargestellt.

Unabhängig von dem Verfahren zur Modellierung stellt sich das Problem der Graphensuche dar. Es ist die Frage zu beantworten, über welche Kanten und Zwischenknoten ein gegebener Zielknoten erreicht werden kann. Im Folgenden sollen diesbezüglich einige Algorithmen vorgestellt werden.

3.3.1 Der Algorithmus von Dijkstra

Der Algorithmus von Edsger W. Dijkstra zur Graphensuche ist ein Verfahren, welches die kürzeste Verbindung zwischen einem Startknoten und jedem anderen Knoten in einem Graphen ermittelt (Single-Source Shortest Path). Der Algorithmus stellt ein Basisverfahren dar, aus dem bereits verschiedene weitere Methoden abgeleitet wurden. An dieser Stelle wird das allgemeine Vorgehen erläutert und anhand eines einfachen Beispiels illustriert (für eine formale Beschreibung des Algorithmus siehe z. B. [Cor90]).

Es wird zunächst eine Liste Q mit allen Knoten des Graphen angelegt (siehe Algorithmus DIJKSTRA: Zeile 3). Nach der Initialisierung wird dann in einer Schleife (Zeile 4) solange jeweils der Knoten mit der kürzesten Pfadlänge aus Q entfernt und verarbeitet bis Q leer ist. Bei der Verarbeitung eines Knotens u werden zunächst die Nachbarknoten, die noch in Q verblieben sind, bestimmt. Für diese wird dann geprüft, ob sie über u auf einem kürzeren Pfad zu erreichen sind (Zeile 14), u wird dadurch als neuer Vorgängerknoten vermerkt (Zeile 16).

Abbildung 12a zeigt ein Geländeraaster, welches zur Demonstration des Dijkstra-Algorithmus die Grundlage für eine einfache Navigationsaufgabe darstellt. In der Karte sind jeweils die Höhenwerte des Geländes eingezeichnet und es soll ein Pfad vom markierten Startfeld zum Zielfeld ermittelt werden. Als Randbedingung sind dabei möglichst wenig Höheneinheiten zu überwinden.

Algorithmus 1 DIJKSTRA

1: Prozedur DIJKSTRA(G, w, s) :	'Hauptfunktion'
2: INITIALISIERE(G, s)	
3: $Q \leftarrow V(G)$	'Liste aller Knoten des Graphen'
4: solange $Q \neq \emptyset$:	
5: $u \leftarrow \text{ENTNEHME-MIN}_d(Q)$	'wähle Knoten mit minimaler Pfadlänge'
6: für alle $v \in \text{NACHBARN}(u) \cap Q$:	'alle Nachbarknoten, die noch in Q '
7: VERARBEITE(u, v, w)	
8: Prozedur INITIALISIERE(G, s) :	
9: für alle $v \in V(G)$:	
10: $d(v) \leftarrow \infty$	'setze alle Pfadlängen auf unendlich'
11: $p(v) \leftarrow \text{NULL}$	'am Anfang kein Vorgänger definiert'
12: $d(s) \leftarrow 0$	'Pfadlänge des Startknotens ist 0'
13: Prozedur VERARBEITE(u, v, w) :	
14: wenn $d(v) < d(u) + w(u, v)$:	'wenn neue Pfadlänge kürzer als aktuelle'
15: $d(v) \leftarrow d(u) + w(u, v)$	'aktualisiere die Pfadlänge'
16: $p(v) \leftarrow u$	'und setze den neuen Vorgänger'

G : Graph; V : Menge der Knoten; u, v : Knoten; s : Startknoten; $w(u, v)$: Kantenlänge; Q : Knotenliste; d : Pfadlänge; p : Vorgänger

Als erster Schritt wird die Rasterdarstellung der Karte in einen Graphen umgewandelt. Als einfache Interpretation stellen die Kanten die Verbindungen zwischen den einzelnen Feldern dar, wobei zur Vereinfachung des Beispiels keine diagonalen Bewegungen erlaubt sind. Als Pfadlänge erhalten die Kanten den Differenzbetrag aus den Höhenwerten. Das zeigt auch, dass der Begriff Pfadlänge allgemein als Bewertung des Pfades verstanden wird. Die Streckenlänge im Sinne der zurückgelegten Entfernung bleibt in diesem Beispiel unberücksichtigt.

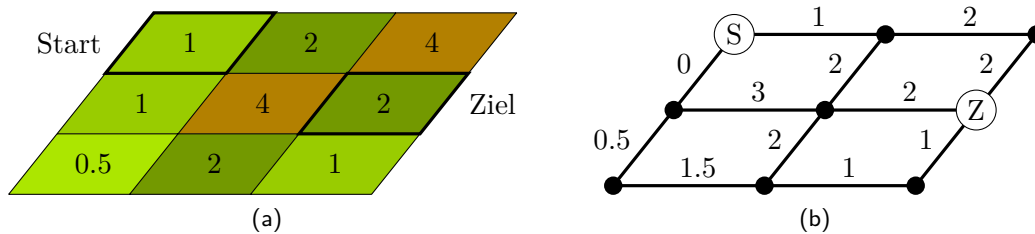
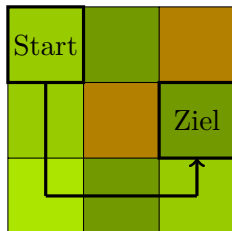
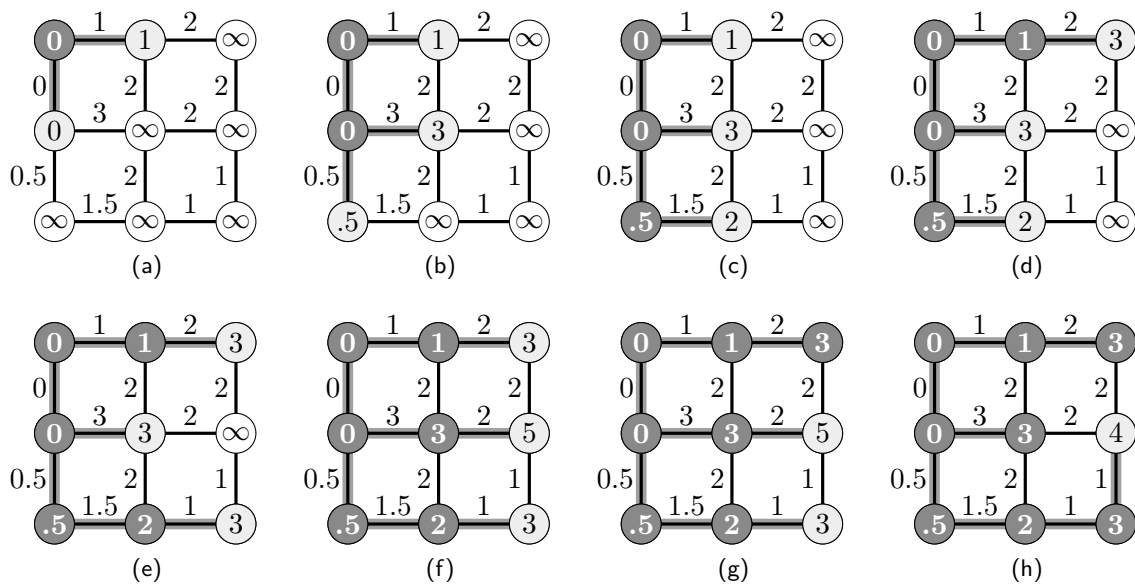


Abbildung 12. Repräsentation einer metrischen Karte als Graph. (a) Eine schematische topographische Karte mit eingezeichneten Höhenwerten. (b) Die Repräsentation der Karte als Graph. Die Kantenwerte sind die Höhendifferenzen der verbundenen Knoten. Quelle: eigene Darstellung.

In Abbildung 13a–h sind die einzelnen Verarbeitungsschritte des Dijkstra-Algorithmus für das gegebene Beispiel illustriert. Dabei wird gezeigt, wie die Knotenwerte im Verlauf der Padsuche modifiziert werden. Die resultierende optimale Strecke unter der gegebenen Randbedingung ist zudem in Abbildung 13i nochmals eingezeichnet.

Für den hier vorgestellten Dijkstra-Basisalgorithmus existieren unterschiedliche Implementierungen, mit denen zusätzlich noch verschiedene Kriterien, wie Speicherbedarf oder Geschwindigkeit, optimiert werden. Daneben gibt es aber noch weitere Verfahren zur Graphensuche, die



(i)

(a) Initialisierung des Graphen und erster Schleifendurchlauf. Schattierte Kanten kennzeichnen die aktuelle Vorgängerbeziehung. (b-g) Der Knoten mit der kleinsten Pfadlänge wird jeweils verarbeitet. Dunkle Knoten sind bereits abgearbeitet, d. h. nicht mehr in Q enthalten. (h) Der Vorgänger des Zielknotens wird wegen der kürzeren Pfadlänge aktualisiert.

Abbildung 13. Algorithmus von Dijkstra. (a-h) Es sind die einzelnen Verarbeitungsschritte zur Pfadsuche dargestellt. (i) Das Höhenmodell mit eingezeichnetem optimalem Pfad. Quelle: eigene Darstellung.

teilweise mit dem Ansatz von Dijkstra verwandt sind und die ebenfalls für Navigationsaufgaben eingesetzt werden. Von den geltenden Randbedingungen hängt es dabei ab, ob diese Ansätze Vorteile bieten. So wird beim Dijkstra-Algorithmus der Pfad von einem Startknoten zu allen anderen Knoten eines Graphen bestimmt. Soll aber nur eine Strecke zu einem bestimmten Knoten ermittelt werden (was üblicherweise bei einer Navigationsaufgabe der Fall ist), so gibt es die Möglichkeit, dass nicht vollständig alle Knoten überprüft werden müssen. Diese Idee liegt dem A*-Algorithmus zu Grunde.

Des Weiteren stellt der Dijkstra-Ansatz ein statisches Verfahren dar. Sollte sich an einer Stelle in einem Graphen eine Änderung ergeben, so müssen alle Pfade neu bestimmt werden. Bei Navigationsverfahren sind die Informationen über die Umgebung meist nicht vollständig, z. B. durch die Beschränkungen der Sensoren, Verdeckungen oder weil die Umgebung selbst sich ändert. Unter der Vorgabe, dass für einen Streckenverlauf neue Hindernisse auftreten, sollten die bisherigen Ergebnisse der Pfadplanung verwendet und nur teilweise neu bestimmt werden. Zu diesem Zweck wurde der D*-Algorithmus entwickelt. Beide der genannten Ansätze werden im Folgenden nochmals näher erläutert.

3.3.2 Der A*-Algorithmus

Der A*-Algorithmus ist eine Verallgemeinerung des Dijkstra-Verfahrens [Cho05]. Zum einen wird die Suchliste Q (siehe DIJKSTRA: Zeile 3) nicht mit der gesamten Knotenmenge initialisiert, sondern es werden sequenziell nur jeweils die Nachbarn der verarbeiteten Knoten hinzugefügt. Gleichzeitig wird für die Entnahme des nächsten Knoten (Zeile 5) der Distanzwert mit einer Heuristik kombiniert, also $u \leftarrow \text{MIN}_f(Q)$ mit $f(u) = d(u) + h(u)$. Über die Heuristik $h(u)$ werden die geschätzten Kosten berücksichtigt (z. B. die Punkt-zu-Punkt Entfernung), um vom Knoten u aus das Ziel zu erreichen.

Mit der Einführung einer Suchheuristik werden bei der Verarbeitung der Knoten diejenigen bevorzugt, die schneller zum Ziel führen. Ist ein Zielpfad gefunden, kann die Verarbeitung der restlichen Knoten abgebrochen werden. Der A*-Algorithmus ist also eine effiziente Methode, um einen optimalen Pfad zwischen genau zwei Knoten zu ermitteln.

3.3.3 Der D*-Algorithmus

Durch lokale Veränderungen muss eine Dijkstra- oder A*-Suche neu durchgeführt werden. Eine Anpassung der Graphensuche unter diesem Aspekt beschreibt der D*-Algorithmus (Dynamic A*) [Ste94; Cho05; Car07]. Ergibt sich in einem Graphen eine Kostenänderung in einem oder mehreren Knoten, so wird diese Änderung wellenförmig an die jeweiligen Nachbarknoten weitergereicht bis sich der Einfluss der lokalen Veränderung verliert. Der D*-Algorithmus basiert ebenfalls auf einer heuristischen Suche, diese wird aber im Gegensatz zur A*-Methode vom Ziel zum Startpunkt durchgeführt. Somit kennt jeder verarbeitete Knoten seine optimale Route zum Ziel, was eine Voraussetzung für die lokale Neuplanung ist, da nur die Teilrouten neu bestimmt werden müssen, die über die veränderten Knoten führen. Als weiterer Vorteil ist auch eine schnelle Neuplanung möglich für den Fall, dass der Rover von seiner geplanten Route abweicht und sich evtl. von seiner abgelegenen Position eine bessere Zielstrecke finden lässt.

3.3.4 Weitere Verfahren zur Pfadsuche

Sowohl für das ursprüngliche A* als auch für das D*-Verfahren sind verschiedene Varianten entwickelt worden. Beispiele hierfür sind die Algorithmen Incremental A*, Lifelong Planning A*, D* Lite, Focussed D* oder der auf den MER-Rovern eingesetzte Field D*. Für die Pfadplanung existieren aber auch noch weitere Methoden, die nicht auf der Repräsentation der Umgebung durch Graphen beruhen.

Ein derartiger Ansatz ist z. B. die Potenzialfeld-Methode [Lat90; Cho05]. Das Verfahren ist abgeleitet von der Idee, den Rover als ein Teilchen zu betrachten, das sich in einem Potenzialfeld bewegt. In Analogie zu elektrischen oder gravitativen Potenzialfeldern stellt das Ziel einen anziehenden Punkt dar, während Hindernisse abstoßende Kräfte ausüben. Ein Geländebereich,

der vom Rover gemieden werden soll, kann also durch einen erhöhten Energieaufwand beim Überqueren modelliert werden. Ein optimaler Pfad folgt möglichst den Gradienten, wobei zusätzlich eine Strategie notwendig ist, um lokale Minima zu überwinden.

Alle bisher vorgestellten Algorithmen sind deterministische Verfahren, sie liefern also unter gleichen Ausgangsparametern immer die gleiche Lösung. Im Unterschied dazu gibt es noch stochastische Methoden. Insbesondere bei Anwendungen mit hochdimensionalen Konfigurationsvektoren (z. B. bei der Steuerungsplanung von komplexen mechanischen Systemen) ist es meist sinnvoll, ein auf Zufalls- bzw. Wahrscheinlichkeitswerten basierendes Verfahren einzusetzen. Dadurch erhält man sehr effizient eine Lösung, die meistens auch nah am optimalen Ergebnis liegt. Für eine Beschreibung einiger solcher Methoden (z. B. die Suche mit Rapidly Exploring Random Trees RRT) siehe [LaV06]. Darüber hinaus existieren für die Rovernavigation, insbesondere zur Parameteroptimierung, auch evolutionäre Ansätze basierend auf künstlichen neuronalen Netzen (siehe z. B. [Pen09] bzw. [47]).

Durch die Betrachtung der Problemstellungen für die Navigation von planetaren Rovern wird die Komplexität einer integrierten Explorationsstrategie deutlich. So zeigt sich, dass die vorgestellten Verfahren und Lösungsansätze nicht unabhängig voneinander eingesetzt werden, sondern meist als Ergebnis eines Verarbeitungsschritts die Grundlage für die nächste Ebene der Navigationsplanung darstellen. Die Qualitätskriterien der Lokalisierung z. B. beeinflussen die Kartenerstellung und damit die Pfadsuchalgorithmen. Daraus ergibt sich ein Gesamtsystem mit vielen Konfigurations- und Parametriermöglichkeiten. Eine Optimierung ist daher oft nur schwer analytisch zu erreichen.

Simulationen können helfen, die Auswirkungen einzelner Parameteränderungen auf das Gesamtverhalten und damit auf die Ergebnisse des Verfahrens abzuschätzen. Auf diese Weise soll auch der im Rahmen dieser Diplomarbeit zu entwickelnde Navigationsansatz evaluiert und diskutiert werden. Im folgenden Kapitel werden daher zunächst die Grundlagen beschrieben, wie Landschaftsmodelle berechnet und dargestellt werden können. Diese sind die Basis für simulierte Sensordaten, die später als Eingabe für die Navigationsalgorithmen dienen sollen.

4 Landschaftsmodellierung

Die 3D-Geländesimulation fällt vor allem in das Gebiet der Computergrafik. Dieses Kapitel soll zunächst einen Überblick über die Methoden und Verfahren aus diesem Bereich geben, die es ermöglichen, virtuelle Landschaften in einer beliebigen Betrachtungsperspektive darzustellen. Darüber hinaus soll gezeigt werden, wie große Geländemodelle insbesondere für die Echtzeitvisualisierung verarbeitet werden.

Die Entwicklung der Grundlagen für die heutigen Methoden der Computergrafik begann Mitte der 1970er Jahre. Auf der Grundlage vieler neu entwickelter Techniken, aber auch basierend auf den Konzepten erster Grafikprozessoren, wurden bald auch erste Ideen für die Generierung künstlicher Landschaften entwickelt. Basierend auf der Beschreibung von fraktalen Strukturen, die von Benoît Mandelbrot entwickelt wurde, konnten bereits 1982 die ersten Verfahren gezeigt werden, um Landschaftsmodelle in unterschiedlichen Auflösungsstufen zu erstellen [Fou82].

4.1 Darstellung von 3D-Geländemodellen

Das grundlegende Verfahren für die 3D-Visualisierung besteht darin, die Oberflächen der Modelle in einfache geometrische Strukturen (meist Dreiecke) zu zerlegen (siehe Abbildung 14). Für jede dieser Teilflächen wird dann z. B. bestimmt, ob diese sichtbar oder verdeckt sind, in welcher Perspektive sie dargestellt werden müssen und in welchem Winkel sie zu den Lichtquellen stehen. Zusammen mit Oberflächeneigenschaften, wie dem Reflektionsverhalten, und der Modelltextur wird dann die jeweilige Erscheinung der Einzelflächen bestimmt. Als Ergebnis kann dann ein Gesamtbild der Szene gerendert werden, d. h., für eine vorgegebene Zielauflösung wird für alle Pixel der entsprechende Farbwert bestimmt.

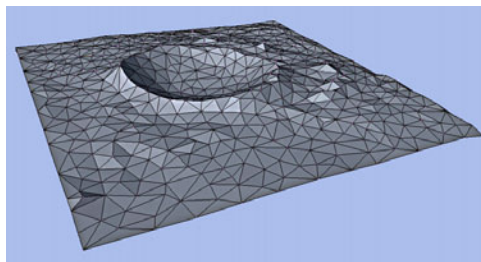


Abbildung 14. Einfaches Geländemodell eines Kraters. Schattierungen auf Grundlage der triangulierten Oberfläche. Quelle: eigene Darstellung mit Hilfe der Software Blender [9] basierend auf einem DTM-Ausschnitt von [38].

Für die Berechnung der Modellgeometrie wurden verschiedenen Verfahren entwickelt, die einen effizienten Renderingprozess und damit eine Echtzeitdarstellung ermöglichen. Einige wichtige Ansätze werden in den folgenden Abschnitten näher erläutert und sind daher in Abbildung 15 als Übersicht dargestellt. Die Abbildung zeigt schematisch den Renderingprozess, den Datenfluss sowie mögliche zusätzliche Komponenten und wo diese zum Einsatz kommen.

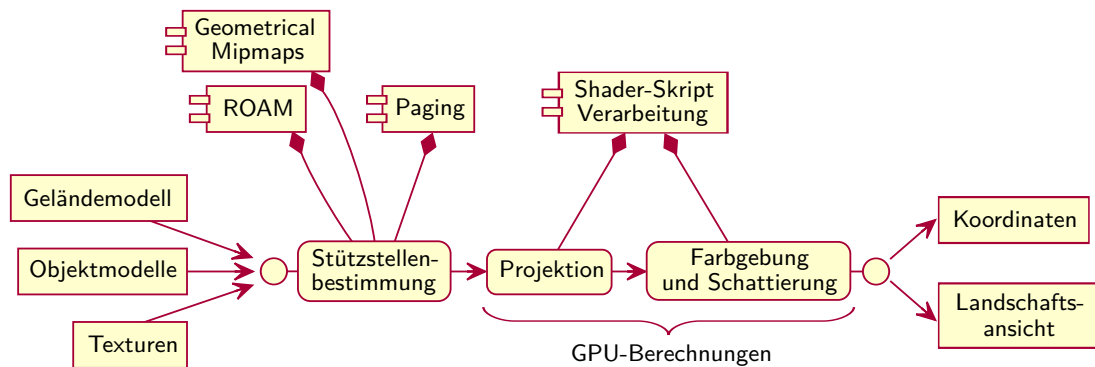


Abbildung 15. Schematischer Renderingprozess. Dargestellt sind die wichtigsten Komponenten, Verarbeitungsschritte und der Datenfluss. Quelle: eigene Darstellung.

Bei hoch aufgelösten oder großen Geländemodellen ist es zunächst notwendig, die Datensätze hinsichtlich der Betrachtungsparameter zu vereinfachen. Mitunter genügt für die in der jeweiligen Perspektive entfernt erscheinenden Flächen eine grobe Annäherung basierend auf wenigen Stützstellen. Dazu werden die Modelle in kleinere Teilbereiche zerlegt und je nach Entfernung wird die Struktur in einem unterschiedlichen Detaillierungsgrad (Level of Detail LOD) dargestellt. Darüber hinaus ist es möglich, auch bei sehr ebenen Bereichen Stützstellen zu entfernen. Nachfolgend werden zwei Lösungsansätze für das Problem der Modellvereinfachung vorgestellt, die insbesondere für die Darstellung großer Geländemodelle geeignet sind.

4.1.1 Das ROAM-Verfahren

Der etwas ältere, aber insbesondere in der Spieleentwicklung lange sehr populäre, Ansatz ist das ROAM (real-time optimally adapting meshes) Verfahren [Duc97]. Eine Grundidee dieser Methode ist zunächst der Aufbau einer effizienten Datenstruktur zur Referenzierung der Dreiecksflächen. Dazu wird ein Binärbaum verwendet, der eine hierarchische Aufteilung der Detaillierungsgrade darstellt (jeder Knoten repräsentiert ein Dreieck, das sich jeweils aus den zwei Teildreiecken seiner Nachfolger zusammensetzt). Bei jedem Ausgabebild (Frame) wird dann geprüft, wie weit jeder Pfad des Baumes verfolgt werden muss, d. h. wie oft jede Dreiecksfläche nochmals zu unterteilen ist, damit eine vorgegebene Fehlerschranke eingehalten wird. Insbesondere können schnell große Bereiche des Geländemodells von den Berechnungen ausgeschlossen werden, die sich nicht im momentanen Sichtbereich befinden. Der Vorteil des Verfahrens ist, dass bei nur kleinen Perspektivänderungen zwischen den Einzelframes, die zu überprüfenden Baumknoten sehr schnell gefunden werden können.

4.1.2 Geometrical Mipmaps

Die meisten Berechnungen für 3D-Darstellungen, vor allem die perspektivischen Transformationen und Texturabbildungen, werden heutzutage sehr effizient von Grafikprozessoren (Graphics Processing Unit GPU) durchgeführt. Viele der neueren Renderingverfahren verwenden daher sogenannte Geometrical Mipmaps [Boe00; Ulr02], weil sich die Berechnungen mit dieser Technik besser für die GPU-Verarbeitung optimieren lassen. Mipmaps (mip steht dabei für *multum in parvo*, also *viel auf kleinem Raum*) wurden zuerst für die Texturierung von 3D-Objekten eingesetzt. Neben einer sehr hoch aufgelösten Textur werden gleichzeitig noch kleinere Kopien gespeichert, wodurch sich eine Bildpyramide für das Texturbild ergibt. Abhängig davon, wie groß das Objekt in der perspektivischen Darstellung erscheint, wird dann eine entsprechende Auflösungsstufe ausgewählt. Damit werden Aliasing-Effekte und nachträgliche Filteroperationen vermieden.

Geometrical Mipmaps oder Geomipmaps übertragen die Idee der Auflösungspyramide auch auf die Objektmodelle selbst. Mit zunehmender Entfernung vom Betrachter kann ein immer einfacheres Modell zum Rendern verwendet werden. Zur Darstellung einer Landschaft wird das Geländemodell zunächst in einzelne Quadranten zerlegt. Für diese Teilmodelle werden dann unterschiedliche Auflösungsstufen mit einer jeweils reduzierten Anzahl an Stützstellen bestimmt. Die Auswahl der zu verwendenden Modellauflösung erfolgt dann in Abhängigkeit von der Perspektive und der Variation in den Höhenwerten in Verbindung mit einer vorgegebenen Fehlerobergrenze (siehe Abbildung 16b).

Zudem werden die Teilflächen oft als Quad-Tree Struktur verwaltet. Das ist ein logischer Baum über die hierarchisch aufgeteilte Gesamtfläche, vergleichbar mit einer Bildpyramide. Jede Teilfläche, die durch einen Baumknoten repräsentiert wird, ergibt sich aus den Flächen der vier Nachfolgerknoten. Damit lassen sich ähnlich wie beim ROAM-Verfahren sehr effizient die LOD-Bestimmungen durchführen und gleichzeitig können schnell die Bereiche ausgeschlossen werden, die außerhalb der Betrachterperspektive liegen.

4.1.3 Paging

Das Paging stellt keinen eigenständigen Ansatz für die Darstellung von Geländemodellen dar, sondern beschreibt eine Technik zur Speicherverwaltung, die z. B. zusammen mit Geomipmaps verwendet werden kann. Um zu entscheiden, in welcher Auflösung ein Geländequadrant dargestellt werden soll, oder um bei einem Perspektivwechsel schnell die LOD-Auflösungen anzupassen, müssen die gesamten Modelldaten verfügbar sein. Das kann bei einem großen Gelände zu einem Speicherproblem führen. Die Idee des Paging ist, die gesamte Geländefläche in einzelne Felder (Pages) aufzuteilen (Abbildung 16a). Die Felder können dann dynamisch geladen und wieder freigegeben werden. Damit ist es z. B. möglich, sich durch ein unendlich ausgedehntes Gelände zu bewegen, da in Bewegungsrichtung immer neue Modellabschnitte geladen werden. Gleichzeitig werden die Speicherbereiche der verlassenen Gebiete wieder freigegeben.

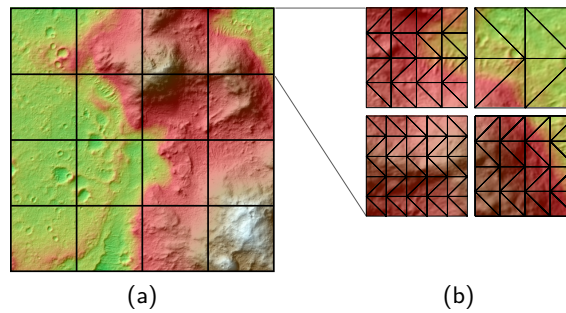


Abbildung 16. Verfahren zur Darstellung von Geländemodellen. (a) Das Höhenmodell wird in einzelne Felder (Pages) aufgeteilt, die dynamisch geladen werden. (b) Jedes Feld wird für die Darstellung nochmals in Quadranten zerlegt, für die je nach Höhenvariation und Perspektive eine unterschiedliche Auflösung gewählt wird. Quelle: eigene Darstellung basierend auf [38].

Eine andere Möglichkeit, das Paging einzusetzen, ist der Ansatz, das gesamte Gelände zunächst in einer niedrigen Auflösung zu laden und nur jeweils den Bereich um den Standpunkt als detailliertes Modell bereitzustellen. Bei einer Roversimulation könnte die Panoramaansicht des gesamten Horizontes für die Navigation notwendig sein, wobei gleichzeitig für die Kinematik eine Bodenauflösung im Zentimeterbereich gewährleistet sein muss. Darüber hinaus lässt sich der Paging-Ansatz auch auf Elemente innerhalb der 3D-Szene übertragen. Bei wiederkehrenden Objekten großer Anzahl, wie z. B. Steine oder Bäume, ist es ebenfalls ausreichend, diese nur in einem gewissen Abstand um den Standpunkt herum zu laden. In weiter Entfernung sind diese Objekte dann meist nicht mehr zu erkennen und müssen nicht dargestellt werden.

Das Geomipmap- und das Paging-Verfahren zusammen erlauben eine Anpassung des Renderingprozesses sowohl an die Rechenleistung als auch an die Hauptspeicherkapazität. Durch Beeinflussung der Darstellungsqualität lässt sich die Anzeigeleistung variieren, z. B. für eine schnelle Echtzeitvisualisierung oder die präzise Koordinatenbestimmung für die Simulation. Außerdem sind Anforderungen umsetzbar, bei denen ein großer Skalierungsbereich für die Geländemodelle notwendig ist. Ein Beispiel hierfür ist die Simulation von Landemanövern. Dafür bedarf es einen kontinuierlichen Übergang von einem sehr großen Geländebereich in niedriger Auflösung zu einer detaillierten Darstellung der Umgebung der Landestelle.

4.2 Anwendungen für die Simulation von Planetenoberflächen

Die Simulation bzw. die Visualisierung von 3D-Umgebungen stellt ein wichtiges Hilfsmittel bei der Entwicklung von Raumfahrtsystemen dar. Innerhalb des breiten Spektrums von Simulationaufgaben ist die Generierung von virtuellen Landschaften ein Teilbereich, der immer mehr an Bedeutung gewinnt. Das liegt insbesondere an den zunehmenden Autonomieanforderungen an die Systeme, aber auch an der stetigen Erweiterung der technischen Möglichkeiten. Mit realistischen Ansichten und genauen geometrischen Abbildungen von planetaren Geländemodellen können Verfahren für Rover aber auch für autonome Landefähren oder Ballonsysteme (Aerobots) entworfen und getestet werden.

In diesem Abschnitt werden verschiedene Simulations- und Visualisierungssysteme vorgestellt, die bei der NASA und der ESA bei der Konzeptionierung und Entwicklung für Raumfahrtmissionen eingesetzt werden. Es soll damit der Stand der Technik beschrieben werden, und gleichzeitig zeigen die Beispiele, wie verschiedene Applikationen integriert werden müssen, um die Funktion von komplexen Systemen wie planetare Rover zu simulieren.

4.2.1 NASA: *Maestro*, *SimScape*, *ROAMS*, *SOOPS*

Für die Operationsplanung im Missionsbetrieb von Raumsonden ist es notwendig, den Wissenschaftlern und Ingenieuren eine Vielzahl von Messdaten zur Verfügung zu stellen. Besonders für Systeme mit vielen Freiheitsgraden, wie sie planetare Rover aufweisen, entstehen deshalb hohe Anforderungen an die Visualisierung dieser Daten. Es ist zu gewährleisten, dass der Zustand des Gesamtsystems sicher erfasst werden kann, da sich weitere Entscheidungen daraus ableiten. Für diese Aufgabe wurde bei der NASA die Anwendung Science Activity Planner (SAP) entworfen [Nor05; Fox06]. Die Weiterentwicklung erfolgte unter dem Namen *Maestro*, wobei im Rahmen der MER-Mission auch eine Version dieser Software frei zur Verfügung gestellt wurde.

Maestro dient im Wesentlichen dazu, neue Befehlssequenzen für die Roversysteme zusammenzustellen (z. B. Navigation zu einem Ziel oder Positionierung des Instrumentenarms). Dazu bietet die Software verschiedene Visualisierungsmöglichkeiten bezüglich der System- und Messdaten. Unter anderem gibt es eine dreidimensionale Ansicht der Umgebung inklusive eines Rovermodells zur Darstellung der momentanen Lage und Ausrichtung (Abbildung 17a). Darüber hinaus lassen sich einfache Bewegungssimulationen durchführen, um die Befehlssequenzen vor der Ausführung zu testen.

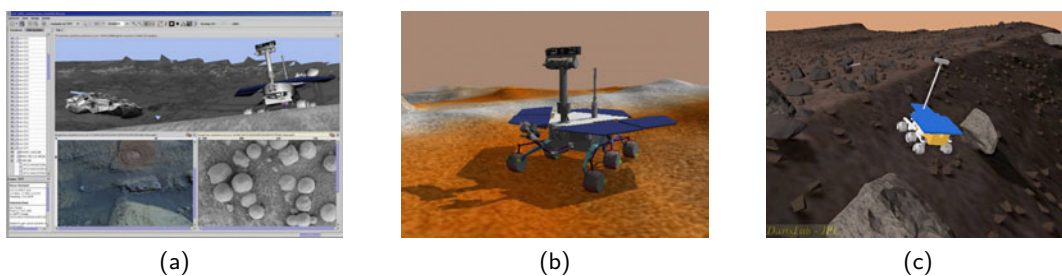


Abbildung 17. Anwendungen der NASA für die 3D-Visualisierung und Simulation. (a) Die Missionsplanungssoftware *Maestro*. (b) Aufbauend auf *SimScape* simuliert die Software *ROAMS* einen Rover auf einem planetaren Geländemodell. (c) Die *SOOPS* 'Field Test in a Box' Simulation demonstriert die autonome Navigation eines Rovermodells durch eine virtuelle Umgebung. Quellen a: [2], b: [3], c: [4].

Das Konzept für die Integration von 3D-Modellen in die Benutzerschnittstellen geht dabei zurück auf das Viz-System [Ngu01]. Damit konnten bereits die Ergebnisse der Stereobildverarbeitung aus der Mars Pathfinder Mission von 1997 als dreidimensionale Umgebung dargestellt

werden. Auch wurde bereits bei dieser Anwendung das Ziel verfolgt, die wissenschaftlichen und die technischen Analysewerkzeuge innerhalb der Steuerungsoberfläche zu integrieren.

Neben der reinen Visualisierung, ist die Darstellung von planetaren Umgebungen auch für die Umsetzung komplexer Simulationsprozesse notwendig, z. B. für die Kinematik, für Bildverarbeitungsmethoden oder Navigationsverfahren. SimScape¹ [Jai06] ist eine Anwendung, mit der sich Geländemodelle auf Grundlage von gemessenen oder künstlich generierten Höhenprofilen erstellen lassen. Darauf aufbauend kann mit Hilfe der Software ROAMS² [Jai04] ein Rover simuliert werden, der sich durch die generierte Landschaft bewegt (Abbildung 17b). So können unter anderem Messdaten und Kamerabilder für beliebige Umgebungen generiert werden.

Die technischen Möglichkeiten werden mitunter soweit genutzt, dass ein gesamtes Roversystem in einer virtuellen Umgebung simuliert wird. Für diese sogenannten Closed-Loop-Simulationen müssen viele verschiedene Komponenten aufeinander abgestimmt und verbunden werden. Innerhalb des Projektes SOOPS³ wurde für diese Anforderungen ein Konzept entwickelt. Die Algorithmen der gesamten Roverarchitektur von der Antriebssteuerung bis zur Bildverarbeitung wurden in der Softwarebibliothek CLARAty⁴ zusammengefasst und mit den Modulen SimScape und ROAMS verknüpft [Hal07]. Das Ergebnis war eine 'Field Test in a Box' Simulation, mit der basierend auf einem Landschaftsmodell getestet wurde, wie sich die Steuerung für einen Explorationsrover verhält (Abbildung 17c).

Das SOOPS-Projekt hat auch gezeigt, dass für eine realistische Simulation des Gesamtsystems eine vielschichtige Softwarearchitektur notwendig ist. Verschiedene Verarbeitungseinheiten, Sensoren, Aktoren, teilweise als reale Geräte oder ebenfalls als simulierte Komponenten, müssen auf Basis der virtuellen Umgebung Daten und Befehle austauschen. Die NASA hat daher im Rahmen des Mission Simulation Facility (MSF) Projekts auch eine Softwareplattform zur Prozesskommunikation entwickelt [Flu02], die insbesondere auf die Entwicklung von zunehmend autonomen Systemen ausgerichtet ist.

4.2.2 ESA: PANGU, 3DROV

In Vorbereitung auf die ExoMars-Mission, für die geplant ist, im Jahr 2018 ebenfalls ein Rover auf dem Mars abzusetzen, sind auch bei der ESA verschiedene Simulationsanwendungen entwickelt worden. Mit Hilfe der Software PANGU⁵ können künstliche Planetenoberflächen generiert werden [Par04; DM09]. Wie in Abbildung 18a dargestellt, werden die extraterrestrischen Landschaften aufgrund verschiedener Kriterien, wie Steinverteilung, Häufigkeit, Tiefe und Alter von Kratern sowie der Licht- und Farbgebung, entworfen. Auf Grundlage der virtuellen Terrains können z. B. Navigationsverfahren für Rover analysiert werden (Abbildung 18b).

¹SimScape Simulator of Surface Characteristics and Attributes for Planetary Environments

²ROAMS Rover Analysis, Modeling and Simulation

³SOOPS Science Operations On Planetary Surfaces

⁴CLARAty Coupled Layer Architecture for Robotic Autonomation

⁵PANGU Planet and Asteroid Natural Scene Generation Utility

Eine andere Anwendung von PANGU ist die Simulation der Steuerung von Landern, wobei neben der visuellen Darstellung der Geländemodelle auch Sensorinformationen wie Lidar-Daten (Light detection and ranging) generiert werden können.

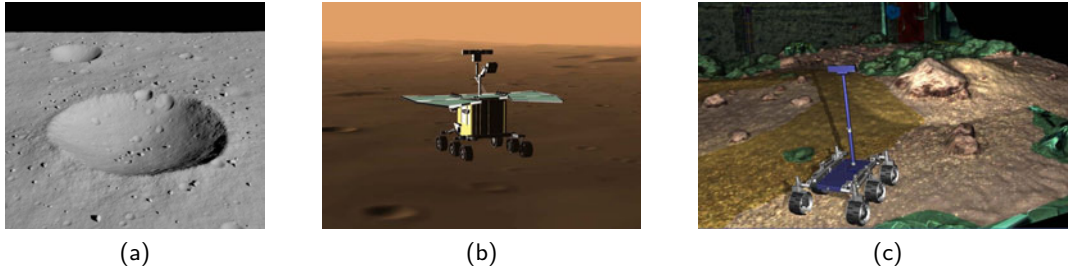


Abbildung 18. Anwendungen der ESA für die 3D-Visualisierung und Simulation. (a) Mit Hilfe der Software PANGU können künstliche planetare Landschaften generiert werden. (b) Eine Anwendung für diese virtuellen Planetenoberflächen ist die Simulation und Analyse von Roversystemen. (c) Eine dreidimensionale Roverumgebung, dargestellt durch das Visualisierungsmodul der 3DROV Softwareplattform. Quellen a: [DM09], b: [McC08], c: [Kap09].

Eine weitere ESA-Anwendung, die speziell als Unterstützung für die Roverentwicklung entworfen wurde, ist 3DROV [Pou08]. Die Software ist eine Simulationsplattform, durch die verschiedene Modellierungskomponenten (z. B. für das Antriebs-, Thermal- und Energieversorgungssystem) über eine Kommunikations- und Steuerungsarchitektur verbunden werden können. Darüber hinaus bietet das System eine zentrale Benutzerschnittstelle und eine grafische 3D-Visualisierung (Abbildung 18c). In der Anzeige der Roverumgebung ist es möglich, verschiedene Systemparameter oder Ergebnisse der Datenverarbeitung, wie den Sichtbarkeitsbereich der Kameras oder klassifizierte Geländebereiche, darzustellen. Darüber hinaus lassen aus den Ansichten auch Sensordaten für Simulationen generieren.

4.3 Entwurf einer Simulationsumgebung für Geländemodelle

Im Rahmen dieser Diplomarbeit soll eine Simulationsumgebung für Planetenoberflächen entworfen werden, die sich an den vorgestellten Anwendungen aus der Raumfahrtssystementwicklung orientiert. Neben der visuellen Darstellung der Umgebung soll es möglich sein, auf Basis der Geländemodelle Navigationsverfahren zu demonstrieren und zu analysieren. Für die Umsetzung der Software sollen zunächst einige Anforderungen bestimmt werden.

4.3.1 Anforderungen

Wie die Beispielanwendungen aus dem vorigen Abschnitt zeigen, dient die Darstellung von dreidimensionalen Geländemodellen neben der Visualisierung, z. B. für den Missionsbetrieb, auch als Analyseverfahren in der Konzeptionsphase. Das Ziel für die Entwicklung einer Simulationssoftware soll es daher sein, eine Plattform zu entwerfen, die als einheitliche Komponente für

verschiedene Aufgabenbereiche eingesetzt werden kann. So soll es möglich sein, eine generierte Landschaft in Echtzeit zu betrachten und eventuell sogar verschiedene Metainformationen über die Modelle einzublenden. Auch die Anbindung an ein Geoinformationssystem (GIS) ist denkbar, um vorliegende Analyseergebnisse für das Gelände, wie Spektral- oder Thermalemissionsdaten, über die 3D-Ansicht abrufbar zu machen.

Neben der visuellen Darstellung soll die Simulationssoftware den Zugriff auf die Modellgeometrie ermöglichen. Mit Hilfe der berechneten Geländekoordinaten in verschiedenen Ansichten können z. B. Algorithmen der Stereobildverarbeitung verifiziert werden. Zudem stellen die perspektivischen Abbildungen der Planetenoberfläche die Grundlage für die Demonstration des Navigationskonzepts dar, das im 6. Kapitel dieser Arbeit beschrieben wird. Im Folgenden sind die wichtigsten Softwareanforderungen, insbesondere auch für die spätere Diskussion der Ergebnisse, aufgelistet.

a) **Größe des Geländemodells**

Es soll ein Gebiet von $2\text{ km} \times 2\text{ km}$ dargestellt werden.

b) **Auflösung des Geländemodells**

Die Mindestanforderung für die räumliche Auflösung des Geländemodells ist 1 m.

c) **Geländeparameter**

Die in der Modellierung zu berücksichtigenden Parameter sind Bodenbeschaffenheit, Unebenheiten und Anstieg des Geländes sowie auftretende Hindernisse.

Bodenbeschaffenheit: Damit wird die Befahrbarkeit des Untergrundes modelliert. Es können z. B. sandige Gebiete gekennzeichnet werden, die zu umfahren sind.

Unebenheiten: Gemeint sind im Wesentlichen kleinere Hindernisse, die aber überfahren werden können. Dennoch stellen sie ein Risiko dar. Gebiete mit großen Unebenheiten, wie Geröllfelder, sollten deshalb ebenfalls gemieden werden.

Anstieg: Für die Roverdynamik gibt es einen maximalen Anstieg, der nicht überschritten werden darf, damit das Fahrzeug nicht umkippt. Der Anstieg daher ein wichtiges Kriterium für die Pfadplanung.

Hindernisse: Hindernisse, die nicht überfahren werden können, müssen ebenfalls bei der Pfadplanung berücksichtigt werden. Es müssen entsprechende Ausweichstrecken gefunden werden.

d) **Visualisierung**

Es muss eine Echtzeitbetrachtung des Geländemodells (evtl. durch dynamische Reduzierung der Genauigkeit) möglich sein.

e) **Simulation**

Für die Weiterverarbeitung der Geländekoordinaten hat die Genauigkeit eine hohe Priorität. Neben der Visualisierung soll es daher einen Simulationsmodus geben, der für eine präzise Geometrieberechnung ausgelegt ist, die Darstellungsgeschwindigkeit kann hierbei vernachlässigt werden.

f) **Schnittstellen**

Neben der interaktiven Bedienung als Visualisierungsumgebung muss die Simulationsumgebung auch eine Schnittstelle zur Softwaresteuerung vorsehen, über die Befehle und Daten (z. B. gerenderte Bilder oder Modellkoordinaten) ausgetauscht werden können.

Für die Umsetzung einer Software basierend auf diesen Anforderungen wurde die Grafikkbibliothek OGRE 3D [40] verwendet. Im nächsten Abschnitt werden daher kurz das Konzept und die wichtigsten Funktionen der verschiedenen OGRE-Komponenten vorgestellt.

4.3.2 Konzept und Funktionen der OGRE 3D Grafikkbibliothek

Ziel des Open Source Projekts OGRE (Object-Oriented Graphics Rendering Engine) ist die Entwicklung einer plattformunabhängigen Softwarebibliothek zur Darstellung von 3D-Szenen. Die verwendete Programmiersprache ist C++, verschiedene Adaptionen für andere Sprachen, wie C# oder Python, sind aber ebenfalls verfügbar.

Ein Kernkonzept der OGRE-Komponenten ist die Hardware-Abstraktion. Die speziellen Eigenschaften unterschiedlicher Grafikprozessoren oder 3D-Programmierschnittstellen, wie OpenGL oder DirectX, müssen nicht berücksichtigt werden, da die entsprechenden Module je nach Zielsystem von der Bibliothek automatisch bereitgestellt werden. Der Schwerpunkt der Programmierung liegt damit auf der Modellierung der einzelnen 3D-Objekte sowie auf der Beschreibung einer Szene, die durch die Relationen zwischen den Objekten definiert wird.

Neben einer verbreiteten Anwendung im Computerspielebereich ist die OGRE-Bibliothek bereits auch zur Evaluierung neuer Verfahren in wissenschaftlichen Projekten verwendet worden. Ein Beispiel ist die Simulation von unbemannten Unterwasserfahrzeugen [Glo09]. Aber auch im Raumfahrtbereich wurden bereits verschiedene Visualisierungsaufgaben auf Grundlage der Open Source Bibliothek umgesetzt. So hat die NASA verschiedene Konzeptideen anhand virtueller Missionsszenarien veranschaulicht und analysiert [Dam07]. Abbildung 19 zeigt ein Mondlandungsmanöver und einen Space Shuttle Einsatz. Die Aufnahmen wurden mit einer auf OGRE 3D basierenden Plattform erstellt [12]. Aber auch die ESA verwendet für die Visualisierungskomponenten der 3DROV Software (Abbildung 18c) diese Grafikkbibliothek [18].

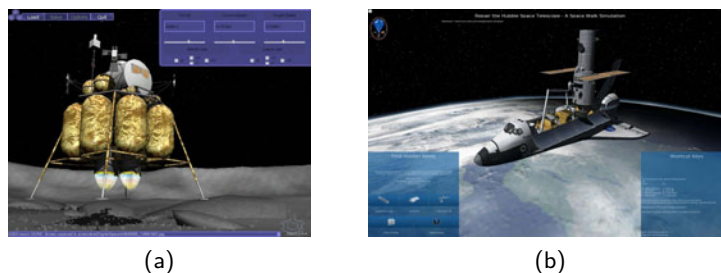


Abbildung 19. Beispiele für die Visualisierung mit OGRE 3D. (a) Ein virtuelles Mondlandema- növer im Rahmen des NASA-Konzepts Lunar Surface Access Module (LSAM) aus dem Jahr 2005. (b) Die Simulation eines Reparatereinsatzes am Hubble Teleskop 2009. Quellen a: [14], b: [13].

In Abbildung 20 ist schematisch ein UML-Klassendiagramm der wichtigsten OGRE-Komponenten dargestellt. Das zentrale Element einer Anwendung ist die Klasse `Root`. Beim Erzeugen des `Root`-Objekts werden die notwendigen Initialisierungen vorgenommen sowie auch das Anwendungsfenster (`RenderWindow`) definiert. Da die 3D-Darstellung vom Zielsystem abhängig ist, gibt es verschiedene Ausführungen der Render-Komponenten (z. B. für OpenGL und DirectX), wobei die jeweils passende Implementierung als Plug-in geladen wird. Die Ausgabekomponente benötigt zudem eine Referenz auf eine Kamera, da diese die Perspektive bestimmt. Neben dem Programmfenster kann auch der Speicherbereich eines Textur-Objekts als Renderziel gewählt werden. Damit sind Effekte wie Spiegelungen möglich oder die Technik wird verwendet, wenn mehrere Ansichten einer Szene gleichzeitig erzeugt werden sollen.

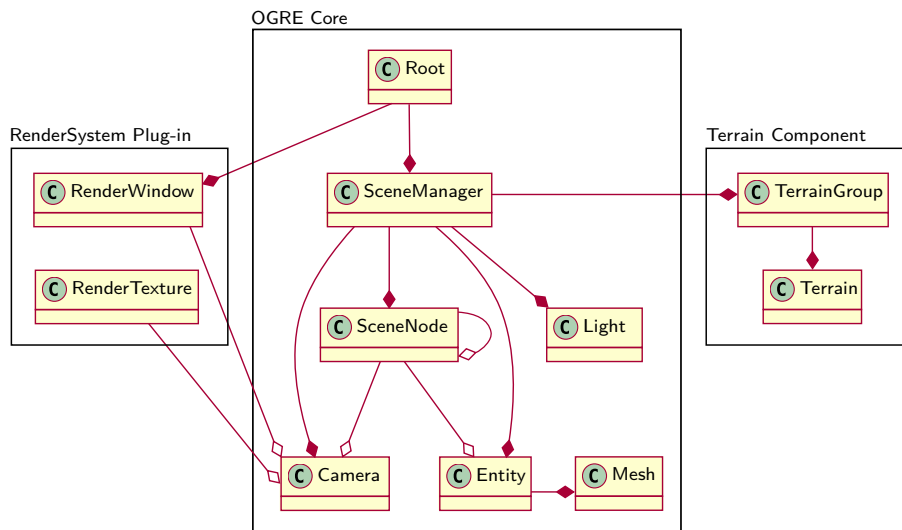


Abbildung 20. Schematisches UML-Diagramm der OGRE 3D Bibliothek. Dargestellt sind die Beziehungen zwischen den wichtigsten Klassen. Quelle: eigene Darstellung.

Eine weitere wichtige Komponente ist das `SceneManager`-Objekt. Dieses dient der Verwaltung aller Szenen-Elemente, liefert also dem Render-System als Inhalt die entsprechenden Eigenschaften und Beziehungen für das darzustellende Gesamtmodell. Unterhalb des `SceneManager`s wird dazu eine hierarchische Struktur aufgebaut, die alle Objekte einer Szene enthält. Dazu zählen neben den elementaren 3D-Modellen und Geometrien auch die Lichtquellen und Kameras. Diesen Ansatz der hierarchischen Relation zur Beschreibung einer Szene nennt man allgemein einen Scene Graph.

Mit Hilfe der `SceneNode` Objekte können auch geometrische Relationen innerhalb der Baumstruktur abgebildet werden. Diese bieten lokale Koordinatensysteme, in welchen die Mesh- oder Kameraobjekte platziert werden. Durch diese abgeleiteten Positionen und Richtungen ist es z. B. möglich, eine Kamera relativ zu einem Rover-Modell zu verorten. Wird das Rover-Modell bewegt, so ändert sich automatisch die Perspektive, da die Kamera mitgeführt wird.

Geometriebestimmung mit Hilfe von Shader-Skripten

Neben der visuellen Darstellung einer Szene ist es für die Roversimulation auch notwendig, die genaue Geometrie des Umgebungsmodells relativ zu einem beliebigen Standpunkt und Blickwinkel zu ermitteln. Dazu muss die Projektion der 3D-Objekte, einschließlich des Geländemodells, auf die Bildebene der virtuellen Kamera bestimmt werden, was die Berechnung von mehreren Koordinatentransformationen erfordert. Grafikprozessoren sind für diese hochdimensionalen Matrixoperationen optimal ausgelegt. Die Nutzung von Berechnungsergebnissen der Grafikkarte stellt damit ein effizientes Verfahren zur Koordinatenbestimmung bezüglich verschiedener Perspektiven dar.

Die Grafikberechnung innerhalb einer GPU beinhaltet mehrere Verarbeitungsschritte, die von sogenannten Shader-Modulen ausgeführt werden. So bestimmt der Vertex-Shader zunächst die Geometrie einer Szene, während der Pixel-Shader anschließend die Farbgebung der Objekte (aus der Kombination von Material, Textur und Beleuchtung) festlegt. Die für diese Arbeit umgesetzte Idee ist, die Shader zu modifizieren, um die Berechnungsergebnisse der Grafikkarte für die Anwendung verfügbar zu machen. Eine von der Firma Nvidia entworfene Skriptsprache für die dafür notwendige Anpassung ist Cg (als Abkürzung für C for graphics). Mit Hilfe eines Cg-Skripts für den Vertex-Shader lassen sich die fertig berechneten Koordinaten in einem Register zwischenspeichern. Ein modifizierter Pixel-Shader kann auf dieses Register zugreifen und die Daten in einen Ausgabebereich (z. B. eine Textur) schreiben. OGRE bietet bereits verschiedene Funktionen zur einfachen Einbindung derartiger Shader-Skripte.

4.3.3 Geländedarstellung basierend auf MRO-Höhenprofilen

Als Geländemodell für die simulierte Planetenoberfläche sollen die Höhenprofile aus der Mars Reconnaissance Orbiter (MRO) Mission verwendet werden. Das Zielgebiet der ExoPlanet-Beispielmission, welches im Wesentlichen das Operationsgebiet des MER-Rovers Spirit umfasst, ist als hoch aufgelöstes DTM verfügbar [38]. Darüber hinaus wurden auch die Stereobilder der HiRISE-Kamera veröffentlicht, die zur Bestimmung der Höhenwerte verwendet wurden. Abbildung 21a zeigt ein Orthobild. Die resultierenden Geländemodelle sind in den Abbildungen 21b und 21c dargestellt. Einmal als Relief und daneben als farbcodiertes Höhenprofil.

Die HiRISE-Aufnahmen für dieses Gebiet haben eine Bodenauflösung von ca. 0,25 m pro Pixel. Der Rasterabstand für das Höhenprofil beträgt 1 m. Um die Qualität dieser Höhendaten sowie die Leistungsfähigkeit der Geländemodellierung mit Hilfe der Grafikkarte OGRE 3D zu demonstrieren, wurde zunächst eine Ansicht generiert, die auch als Aufnahme aus der MER-Mission existiert. Abbildung 21d zeigt das McMurdo-Panorama, aufgenommen vom Rover Spirit im Jahr 2006. Im Vergleich dazu ist in Abbildung 21e eine virtuelle 360-Grad-Ansicht bezüglich des gleichen Standorts dargestellt. Sie ist das Ergebnis des Renderingprozesses auf Grundlage des MRO-Höhenprofils sowie dem abgebildeten Orthobild als Textur. Die Position und die entsprechenden Blickrichtungen sind zudem in Abbildung 21c gekennzeichnet.

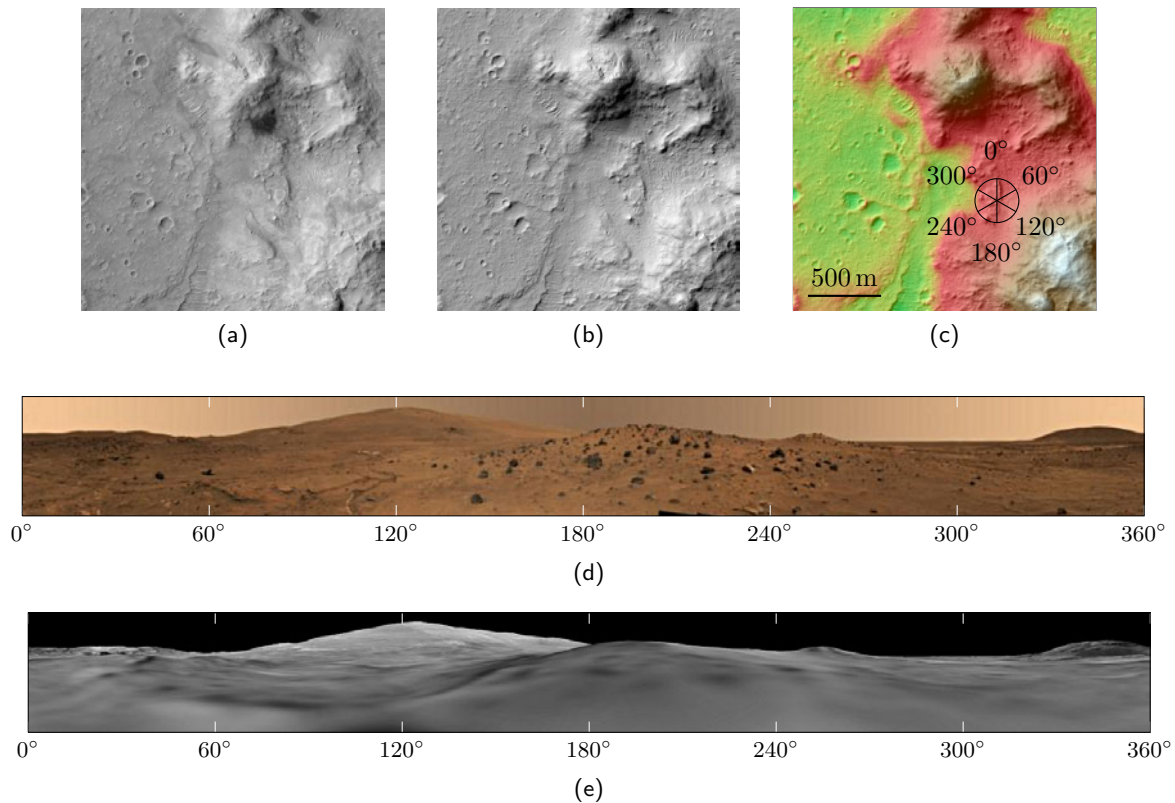


Abbildung 21. Geländemodellierung auf Grundlage von MRO HiRISE Daten. (a) Aufnahme der Orbiterkamera als Orthobild. (b) Das Höhenmodell als schattiertes Relief. (c) Das farbcodierte Höhenprofil. Eingezeichnet sind Standort und Azimutwinkel der Panoramaaufnahmen. (d) Das McMurdo-Panorama, aufgenommen vom Rover Spirit. (e) Zum Vergleich die gerenderte Panoramaansicht. Quellen a–c: [38], d: [34], e: eigene Darstellung basierend auf [38].

Die hohe Auflösung der HiRISE-Bilder ist im Panoramavergleich gut zu erkennen. So sind selbst kleinere Steine und Schatten als dunkle Flecken sichtbar. Im Folgenden soll kurz erläutert werden, wie das Geländemodell in der dargestellten Ansicht generiert wurde. Insbesondere soll gezeigt werden, wie die MRO-Geländemodelle als Grundlage für die Geländemodellierung mit OGRE 3D verwendet werden können.

Das Subsystem von OGRE 3D für die Geländedarstellung wurde 2009 vollständig überarbeitet und steht seit der Version 1.7 offiziell zur Verfügung. Das Konzept orientiert sich an der in Abschnitt 4.1.2 vorgestellten Geomipmap-Technik. Dazu kann das gesamte Geländemodell in einzelne Quadranten geteilt werden, wobei jedes dieser Felder eine eigenständige Modellinstanz darstellt und durch die Klasse `Terrain` (siehe Abbildung 20) repräsentiert wird. Unter Umständen ist auch eine Teilung des Geländemodells notwendig, wenn für die Grafikkarte eine maximale Größe für eine einzelne Modelltextur vorgegeben ist (das verwendete Orthobild hat eine Auflösung von 8192×8192 Pixeln). Darüber hinaus können die Daten der einzelnen `Terrain`-Objekte dynamisch geladen und freigegeben werden, womit auch eine Paging-Technik (vgl. Abschnitt 4.1.3) zur Verfügung steht. Zur Verwaltung aller `Terrain`-Objekte dient eine Instanz der Klasse `TerrainGroup`.

Die Höhenprofile für das Geländemodell werden dem OGRE Terrain-System als Float-Array übergeben. Die MRO-Höhendaten liegen im VICAR-Datenformat vor. Dieses allgemeine Format für Binärdaten ist als Teil des gleichnamigen Bildverarbeitungssystems (VICAR Video Image Communication And Retrieval) entstanden [24]. Aus der Spezifikation dieses Datenformats (siehe [Dee92]) geht hervor, dass VICAR Dateien mit einem Text-Header beginnen. Darin sind hauptsächlich verschiedene Metainformationen, wie dem Instrument und dem Zeitpunkt der Aufnahme oder Angaben zum geltenden Koordinatensystem, aber auch alle Verarbeitungsschritte, die ausgehend von den Rohdaten vorgenommen wurden, dokumentiert. Darüber hinaus finden sich hier auch Angaben zur Interpretierung der Daten, also die Datentypen und evtl. auch Offset- und Skalierungsparameter. Mit Hilfe dieser Header-Informationen ist es möglich, die Höhendaten in das gewünschte Ausgabeformat zu transcodieren.

Ein sehr effizientes Hilfsmittel zur Bearbeitung von großen Bildern (das Raster aus Float-Höhenwerten kann äquivalent zu Pixelbildern verarbeitet werden) ist die Python Imaging Library [42]. Für diese Bibliothek lassen sich Plug-ins anlegen, um beliebige Datenformate einzulesen. Danach lassen sich verschiedene Bildoperationen, wie das Ausschneiden eines Bereiches oder Skalierungen, durchführen, wobei die Ausgabe im gewünschten Zielformat erfolgt. Die internen Funktionen sind dabei in C programmiert, so dass die Python Bibliothek auf auch großen Daten sehr schnell arbeitet. Auf diese Weise konnten die Höhenmodelle für die Darstellung durch die OGRE-Komponenten vorbereitet werden.

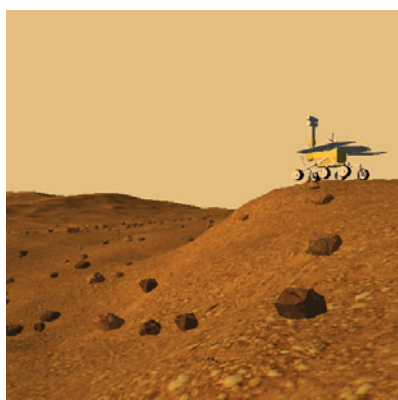
Für die Panoramadarstellung in Abbildung 21e wurden von dem Modell mehrere Aufnahmen in verschiedenen Richtungen gerendert. Da die Brennweite und der Öffnungswinkel der Kamera bekannt sind, können die Einzelbilder mit einer entsprechenden Software (z. B. [22]) zu einer Panoramaprojektion zusammengefügt werden.

4.3.4 Modellierung der Geländeparameter

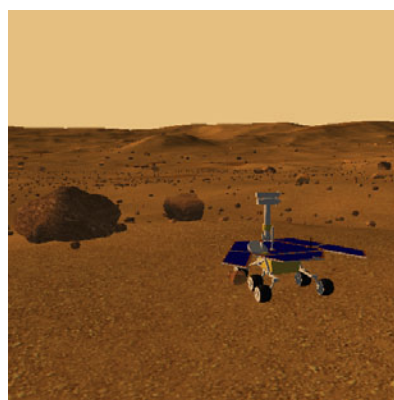
Nachdem die Darstellung des Geländemodells umgesetzt ist, muss für eine realistische Farban-sicht zunächst die Textur angepasst werden. In der Darstellung der Ergebnisse in Abbildung 22 ist zu erkennen, dass im Nahbereich eine sehr detaillierte Bodentextur verwendet wurde, aber gleichzeitig auch großflächige Hell-Dunkel Strukturen in den entfernten Bereichen vorhanden sind. Die hoch aufgelöste Ansicht des Untergrunds ist notwendig, um z. B. Algorithmen für die Stereobildverarbeitung zu testen, da für die Geometriebestimmung genügend markante Punkte in den Aufnahmen gefunden werden müssen. Dazu wird ein periodisch wiederkehrendes Bodenmuster verwendet, das zudem noch Normalenvektordaten der Oberfläche enthält, womit ein leicht räumlicher Eindruck erzeugt werden kann.

Die Schattierungen des Geländes werden auf Grundlage der definierten Lichtquelle bestimmt. Zudem wurde das Orthobild in die Berechnung einbezogen, da dieses neben den Schatten auch die unterschiedliche Helligkeit des Bodens beinhaltet, was der Gesamttextur nochmals einen realistischeren Eindruck verleiht.

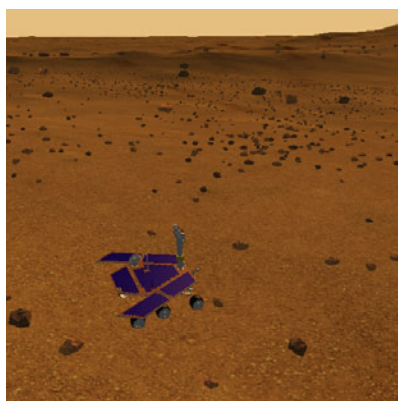
Für die Simulation von Navigationsverfahren müssen außerdem gemäß den Anforderungen aus Abschnitt 4.3.1 noch die verschiedenen Geländeparameter modelliert werden. Die jeweilige Umsetzung ist in Abbildung 22 dargestellt und wird im Folgenden beschrieben, wobei auch auf verschiedene Erweiterungsmöglichkeiten eingegangen wird. Die unterschiedlichen Perspektiven sind maßstabsgetreu und der abgebildete Rover (Modell basierend auf [21]) dient dem Größenvergleich. Zur Bewertung der Simulationsergebnisse wurde eine Panoramaaufnahme der Mars Pathfinder Landestelle hinzugefügt. Hierbei handelt es sich um eine sehr steinige Umgebung, wie sie auch die künstliche Landschaftsgenerierung zum Ziel hatte.



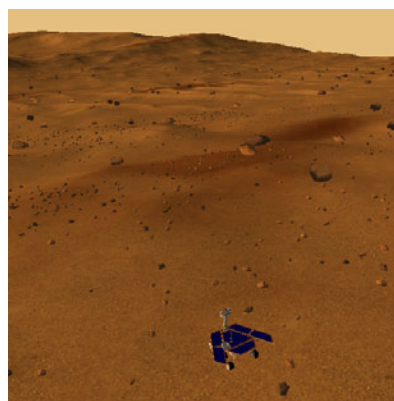
(a)



(b)



(c)



(d)



(e)

Abbildung 22. Simulation unterschiedlicher Geländeparameter: (a) Anstiege, (b) Hindernisse, (c) Unebenheiten und (d) die Untergrundbeschaffenheit. (e) Zum Vergleich eine Panoramaaufnahme der Mars Pathfinder Landestelle. Quellen a-d: eigene Darstellung, e: [32].

a) Geländeanstieg

Der Anstieg des Geländes (siehe Abbildung 22a) ist bereits über die Höhenwerte vorgegeben. Die einzelnen Stützstellen werden für den Pixel-Shader während des Renderingprozesses linear interpoliert, wobei der Rasterabstand in der maximalen Detailansicht der MRO-Auflösung von 1 m entspricht. Sind spezifische Anstiegswerte für eine Simulationsaufgabe vorgegeben, so besteht eine Möglichkeit darin, die Höhenwerte zu skalieren. Ein weiterer Ansatz, der z. B. auch für die Anforderung einer größeren räumlichen Auflösung der Höhenwerte gewählt werden kann, ist die Verkleinerung des Modellrasters durch ein beliebiges Interpolationsverfahren. Anschließend kann ein Rauschanteil addiert werden, wobei sich ein natürlich wirkendes Rauschsignal mit Hilfe des Perlin Noise Verfahrens (vgl. [Bur08]) erzeugen lässt. Alternativ dazu sind auch Methoden entwickelt worden, um ein Geländemodell komplett automatisch zu erzeugen [Dac06; Lat08]. Entweder durch zufallsbasierende Verfahren oder durch die Nachbildung von Erosionsprozessen bis hin zur Simulation von Bodendeformationen z. B. durch Krateranschläge. In der ESA-Software zur Generierung von Planetenoberflächen PANGU (siehe Abschnitt 4.2.2) sind derartige auf geologischen Prozessen basierenden Techniken Bestandteil der Modellierung [Par04].

b) Hindernisse

Zur Simulation von Hindernissen, die für einen Rover zu umfahrende Objekte darstellen, werden der virtuellen Landschaft Steinmodelle hinzugefügt. Abbildung 22b zeigt diese großen Steine, die zufällig über das Gelände verteilt sind, im Verhältnis zum Rovermodell. Dabei handelt es sich immer um das gleiche 3D-Modell, jedoch mit verschiedenen Texturen und einer variierenden Skalierung und Ausrichtung. Das zu Grunde liegende Modell wurde mit der freien Software Blender [9] erstellt. Werden mehrere individuelle Steinmodelle benötigt, so muss evtl. ein zusätzliches Verfahren entwickelt werden, womit eine schnelle interaktive Generierung von Steinformen möglich ist (wie z. B. [Mit03]). Zudem existieren auch Ansätze, um natürlich wirkende Gesteinsstrukturen algorithmisch zu erzeugen (siehe [Dac06; Pey09]).

Möglich ist auch, anstelle der zufälligen Positionierung der Steine, eine empirische Verteilungsfunktion zu verwenden. Erste Analysen von Steingrößen und -verteilung auf der Marsoberfläche wurden als Ergebnis der Viking-Missionen vorgestellt. Aber auch auf aktuelle Bilder der HiRISE-Kamera des Mars Reconnaissance Orbiters wurden Steinerkennungsverfahren angewendet, um ihre Häufigkeiten je nach Größe zu charakterisieren (siehe dazu [Gol01; Gol08]).

c) Unebenheiten

Eine weitere wichtige Eigenschaft einer Planetenoberfläche sind vorkommenden Unebenheiten durch kleinere Steine und Geröll. Derartige Objekte stellen nicht direkt ein Hindernis dar, da ein Rover sie überwinden kann. Die notwendigen Manöver bedeuten für das Fahrwerk dennoch ein Risiko, weshalb unwegsame Gebiete wenn möglich gemieden werden sollen. Wie Abbildung 22c zeigt, wurden in der simulierten Landschaft Felder mit unterschiedlicher Verteilung (basierend auf Zufallswerten) von kleineren Steinen erzeugt.

Wie bereits bei den großen Steinen, die als Repräsentation von Hindernissen dienen, wird auch bei den kleinen Steinen nur ein einzelnes 3D-Modell zur Darstellung verwendet. Der Unterschied zwischen den Objekten ergibt sich nur durch die Skalierung und Lage. Neben der vereinfachten Verwaltung der Modelle liegt ein Vorteil der Verwendung von Objektkopien in der Geschwindigkeit des Renderingprozesses.

Das Übertragen der Modellkoordinaten und der Texturen in den Grafikprozessor sowie das Initialisieren der Datenstrukturen stellt eine vergleichsweise langsame Operation dar, so dass die Darstellung von sehr vielen individuellen 3D-Objekten möglicherweise nicht mehr in Echtzeit möglich ist. Dagegen ist die Verarbeitung großer Koordinatenmengen mit jeweils einer einmaligen Initialisierung der Grafikpipeline ein sehr effizienter Prozess. Möglichst viele Einzelmodelle werden daher zu einer Gesamtstruktur zusammengefasst und als eigenständige Geometrieinstanz (auch als Batch bezeichnet) der GPU übergeben. Darüber hinaus wird in der Planetensimulation ein Objekt-Paging Verfahren verwendet, so dass z. B. die kleinen Steine nur in einem vorgegebenen Umkreis (100 m) dargestellt werden. Für die OGRE-Bibliothek existiert für diese Funktion eine Erweiterung [49], in der sowohl die Batch-Verarbeitung als auch die Objekt-Paging-Methode implementiert ist.

Für eine Geländesimulation mit der Anforderung einer möglichst realistischen Planetenumgebung könnte es zudem ein Ansatz sein, die Geröllverteilung von Kraterformen auf der Oberfläche abzuleiten. Da die kleinen Felsteile und Steine das Auswurfmaterial bei einem Einschlag darstellen, könnten hier Modelle zur Kraterbildung herangezogen werden. Die zu Grunde liegenden Prozesse sind jedoch sehr komplex und es konnte in der Literatur keine Beschreibung einer Simulationsmethode (abgesehen von der Kraterbildung selbst und evtl. der damit einhergehenden Sekundärkraterentstehung) gefunden werden. Eine Alternative besteht darin, die Steinverteilung in Aufnahmen verschiedener Explorationsmissionen auszuwerten und wie bereits bei der Modellierung von Hindernissen beschrieben, empirische Verteilungswerte abzuleiten.

Mit den beschriebenen Renderingverfahren und den gegebenen Möglichkeiten der Detailabstufung ist es auch denkbar, die Auflösung des Geländemodells bis auf die Größenordnung des Radabstandes zu erhöhen. Damit können Unebenheiten direkt in den Höhenprofilen abgebildet werden. Eine derart präzise Untergrundmodellierung bis in den Zentimeterbereich ist jedoch für den Test eines einfachen Navigationskonzepts nicht notwendig, könnte aber durchaus eine Anforderung bei der Kinematiksimulation und bei der Analyse der Fahrwerkssteuerung sein.

d) Bodenbeschaffenheit

Für eine sichere Navigation muss neben Hindernissen auch die Untergrundbeschaffenheit berücksichtigt werden. Beim Befahren von losem sandigem Boden besteht die Gefahr, dass sich die Räder darin festfahren, wie es sich beim Mars Exploration Rover Spirit im Jahr 2009 ereignete. Wie in Abbildung 22d zu erkennen ist, wurde in der Softwareumsetzung eine dunkle Texturfarbe zur Markierung von kritischen Bodenstrukturen verwendet.

Der Effekt wurde durch die Verwendung von zwei verschiedenen Texturen erreicht, die basierend auf einer Transparenzmaske kombiniert wurden (auch bezeichnet als Alpha-Blending). Die Pixelwerte dieser Bildmaske, die als ein zufälliges Muster generiert wurde, können als Indikator für die Befahrbarkeit des Untergrundes interpretiert werden. Der Wert kann direkt als Parameter in die Pfadplanung einfließen. Als erweiterte Möglichkeit ist auch eine Modellierung als Reibungs- oder allgemein als Traktionsparameter denkbar. Die Simulation kann basierend darauf, das physikalische Verhalten des Rovers bestimmen. Aus der sich ergebenden Raddynamik muss die Fahrwerkssteuerung dann zunächst Schlupfdaten ermitteln, die als Parameter dem Navigationsmodul zur Verfügung gestellt werden. In der Praxis werden zur Untergrundklassifizierung verschiedene Verfahren gleichzeitig eingesetzt. Zum einen können sandige Gebiete visuell detektiert werden, zum anderen werden sie, wie beschrieben, erst während des Befahrens erkannt.

Wie auch schon bei den anderen Geländeparametern, besteht bei der Charakterisierung der Bodenbeschaffenheit die Möglichkeit, die Modellierung von geologischen Prozessen abzuleiten. Die in dieser Arbeit konzipierte Simulationsumgebung ist auf eine ebene, steinige Umgebung ausgerichtet. Planetare Oberflächen weisen aber auch Regionen auf, die von Sand- und Staubverwehungen geprägt sind. Die numerische Simulation von Sanddünen aufgrund von gegebenem Ausgangsmaterial und verschiedenen Windbedingungen (siehe z. B. [Par07]) kann hierbei eine Methode sein, um die Struktur des Untergrunds und damit auch die physikalischen Eigenschaften zu bestimmen. Darüber hinaus lassen sich auch Formgebung und die sich daraus ergebenden Geländeanstiege ableiten.

4.3.5 Ergebnis der Geländesimulation

Die Aufnahmen aus Abbildung 22 stellen das Ergebnis der Softwareumsetzung zur Modellierung von Planetenoberflächen dar. Um nochmals den visuellen Eindruck der simulierten Landschaft wiederzugeben, wurden für Abbildung 23 einzelne Bilder aus einem Anflug auf das Operationsgebiet des Rovers ausgewählt. Zum Vergleich und zur Orientierung wurde auf das Gelände zunächst die bei [38] verwendete Höhenfarbskala projiziert. Daneben ist die simulierte Landschaft dargestellt. Darin sind auch die Oberflächendetails, wie die unterschiedliche Untergrundbeschaffenheit oder die Steinmodelle, zu erkennen (für eine animierte Darstellung siehe [20]).

Neben der visuellen Darstellung der virtuellen Planetenoberfläche war eine weitere Anforderung die Bereitstellung der Modellkoordinaten. Wie in Abschnitt 4.3.2 beschrieben, werden die einzelnen Vektorkomponenten bezüglich der aktuellen Perspektive während des Renderingprozesses von der GPU bestimmt und können mit Hilfe modifizierter Shader-Skripte verfügbar gemacht werden. Auf Grundlage dieser räumlichen Daten ist es dann möglich, sowohl Tiefenkarten von der Roverumgebung, als auch Sichtbarkeitskarten der Umgebung zu erstellen.

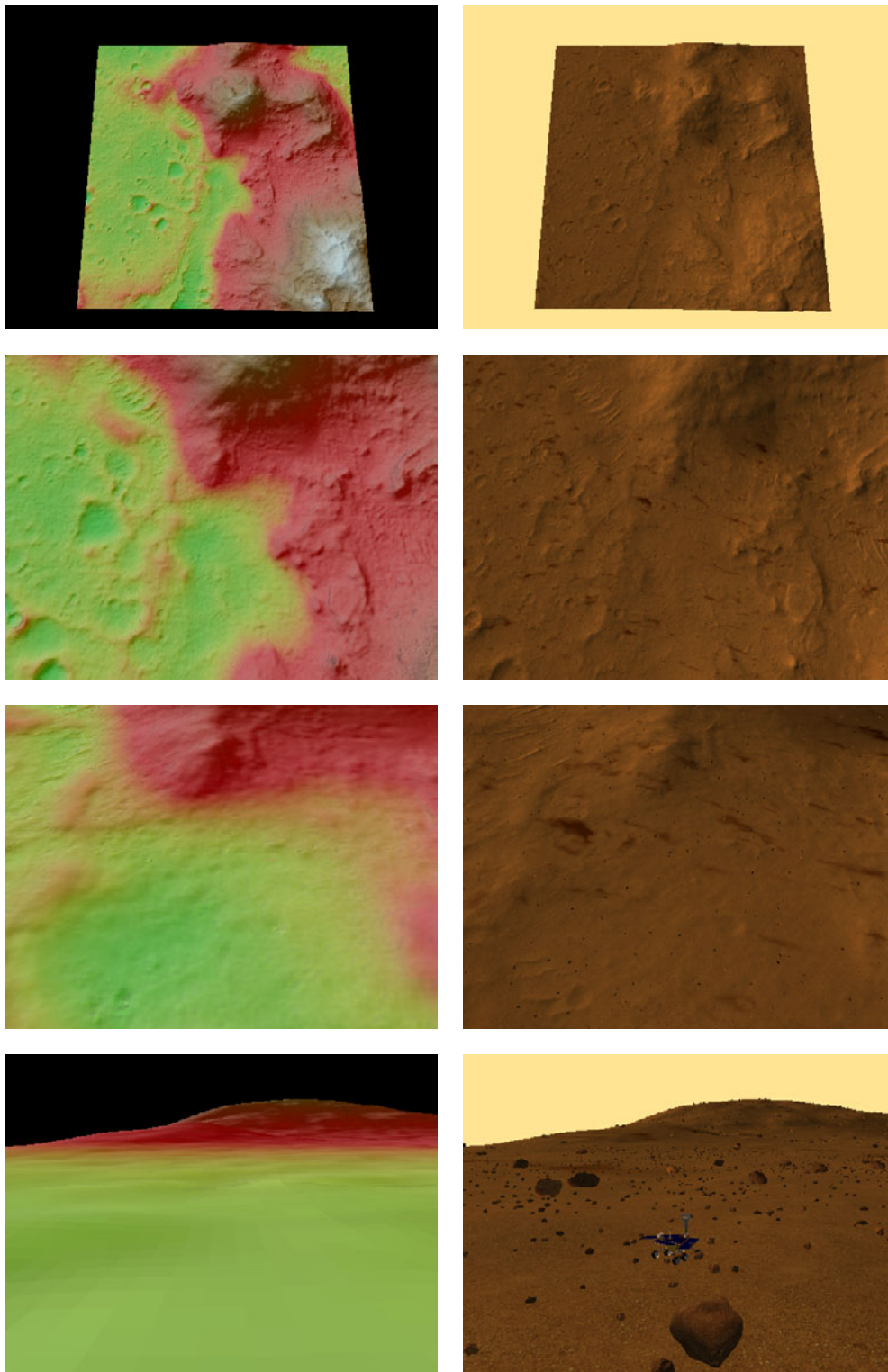


Abbildung 23. Ergebnis der Geländesimulation. Dargestellt sind Ausschnitte aus einem animierten Anflug auf das Operationsgebiet. Quelle: eigene Darstellung (siehe auch [20]).

Abbildung 24 zeigt am Beispiel des McMurdo-Panoramas das Ergebnis dieser Verarbeitung. Die Tiefenkarte (Abbildung 24b) repräsentiert das Ergebnis der Stereobildanalyse. Sie beschreibt den Abstand der einzelnen Bildpunkte vom Sensorsystem. Für die Navigation von Bedeutung ist dann die Projektion dieser Koordinaten in die Geländeebene (Abbildung 24c, für die Position und Blickrichtungen siehe auch Abbildung 21c). Basierend auf diesen Sichtbarkeitskarten kann eine Pfadplanung z. B. zur Hindernisvermeidung erfolgen.

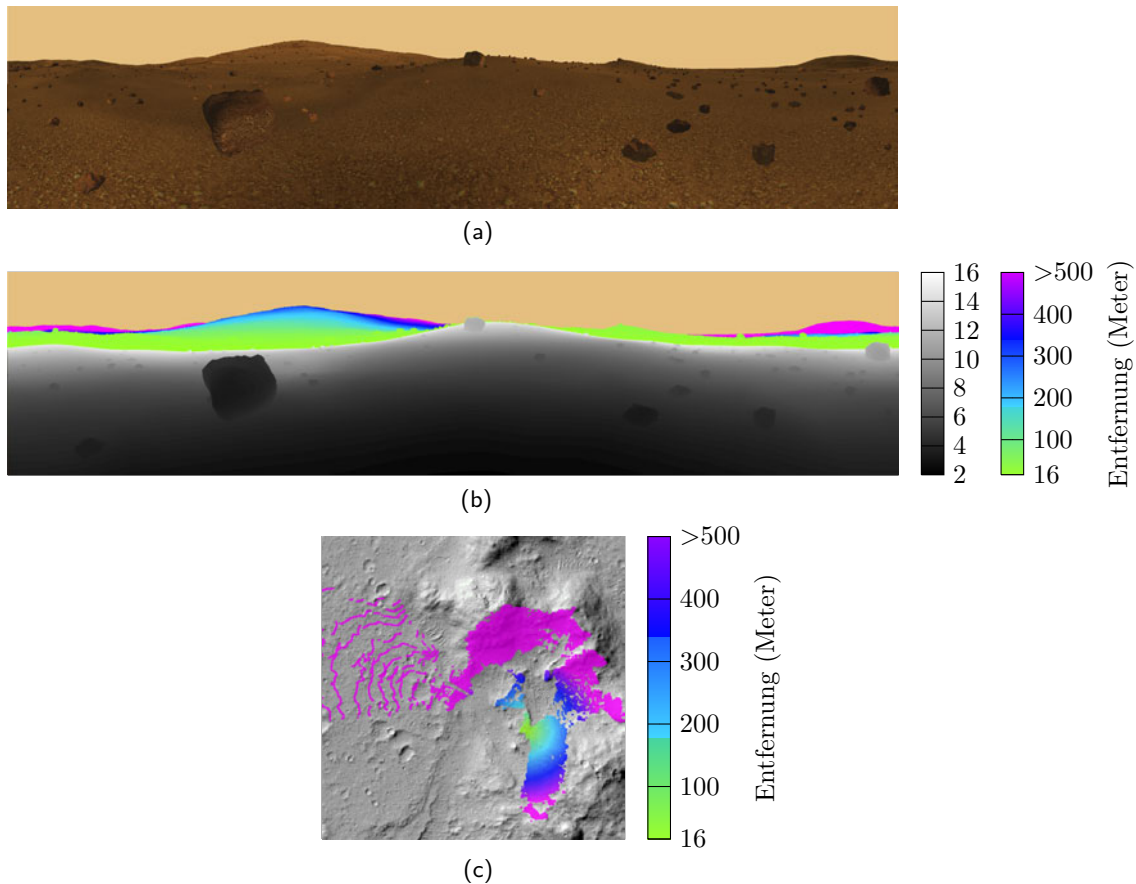


Abbildung 24. McMurdo-Panorama als Tiefenkarte. (a) Die Panoramaaufnahme aus der Simulation. (b) Tiefeninformationen in der Panoramaansicht. (c) Die Sichtbarkeitskarte als Projektion der Bildpixel auf die Geländeebene. Quelle: eigene Darstellung.

Die Implementierung der Geländesimulation beinhaltet somit neben der Möglichkeit der Visualisierung auch entsprechende Funktionen zur Bestimmung der Modellkoordinaten. Bezüglich der formulierten Anforderungen war es notwendig, die Simulationsdaten sowohl als interaktive Darstellung als auch über eine Softwareschnittstelle zur Verfügung zu stellen. Der Zugriff auf die Bild- und Modelldaten ist notwendig, um Navigationsanwendungen auf Grundlage von generierten Planetenoberflächen zu entwickeln und zu testen. Das Konzept und die Umsetzung für die Datenschnittstelle werden daher im Abschnitt über die Integration der einzelnen Komponenten zur Simulation der Rovernavigation (siehe Abschnitt 5.1.4) beschrieben.

5 Navigationskonzept für planetare Rover

Nachdem im Kapitel 3 bereits ein Überblick über die Problemstellungen und verschiedene Grundlagenverfahren für die autonome Navigation gegeben wurde, wird im Folgenden ein konkretes Explorationskonzept erarbeitet und in der Umsetzung beschrieben. Als Ergebnis wird eine Closed-Loop-Simulation für verschiedenen Rover basierend auf der vorgestellten Anwendung zur Geländemodellierung durchgeführt.

5.1 Entwurf einer Explorationsstrategie

Die Anforderungen an das Navigationsverfahren leiten sich aus der ExoPlanet-Mission ab (siehe Abschnitt 2.2.1). Ziel ist, einen Algorithmus für die Pfadplanung zu entwickeln, der es ermöglicht, auf einer Planetenoberfläche ein zusammenhängendes Gebiet bei möglichst kurzer Gesamtstreckenlänge zu erfassen. Gleichzeitig müssen die lokalen Geländeeigenschaften bezüglich ihrer Befahrbarkeit berücksichtigt werden. Treten Hindernisse oder kritische Anstiege bzw. Bodenbeschaffenheiten auf, so muss das Umfahren dieser Bereiche sichergestellt werden.

5.1.1 Anforderungen und Problemstellung

Das zu erarbeitende Konzept für die autonome Rovernavigation soll als Ergebnis dieser Arbeit umgesetzt und validiert werden. Die zu implementierenden Verfahren dienen vor allem der Machbarkeitsanalyse. Der Schwerpunkt liegt daher auf der Integration der Navigationsalgorithmen mit der Geländesimulation. Für diesen Demonstrationszweck soll ein möglichst einfaches Explorationskonzept entwickelt werden. Gleichzeitig sollen die Möglichkeiten der autonomen Navigation und verteilten Exploration aufgezeigt werden. Das Verfahren muss daher eine in sich geschlossene Anwendung darstellen, die eine Closed-Loop-Simulation ermöglicht. Für diese Randvorgaben und unter Berücksichtigung der ExoPlanet-Missionsziele werden nachfolgend die wesentlichen Anforderungen formuliert.

a) Sensordaten

Als Datenbasis für die Navigationsalgorithmen dienen die Koordinaten der lokalen Umgebung bezüglich der als bekannt angenommenen Position. Die Klassifizierung der einzelnen Koordinatenvektoren nach Bodenbeschaffenheit und Unebenheiten wird als Ergebnis

einer entsprechenden Vorverarbeitung ebenfalls vorausgesetzt. Zudem wird davon ausgegangen, dass Hindernisse bereits verortet sind und ein Anstiegsprofil der Umgebung verfügbar ist.

b) **Pfadplanung**

Für die Pfadplanung sollen folgende Geländeparameter berücksichtigt werden: Geländeanstieg, Hindernisse, Unebenheiten und Bodenbeschaffenheit. Hinsichtlich dieser Kriterien ist zu entscheiden, ob ein Areal umfahren (Hindernisse, große Anstiege) oder möglichst gemieden (ungünstige Untergrundbedingungen, kleinere Anstiege) werden soll. Daraus ist eine optimale Route abzuleiten.

c) **Autonomie**

Auf Grundlage der Eingabedaten soll die Navigation autonom, d. h. ohne externe Unterstützung oder Steuerung, erfolgen. Unter der Voraussetzung, dass die Umgebungsparameter vom Sensorsystem des Rovers automatisch bereitgestellt werden können, soll der Navigationsprozess selbständig ablaufen.

d) **Exploration**

Ziel des zu entwickelnden Verfahrens ist es, ein zusammenhängendes Gebiet ($2\text{ km} \times 2\text{ km}$) flächendeckend zu erfassen. Es sollen keine Vorkenntnisse über das Gelände notwendig sein. Als Randbedingung soll eine möglichst geringe Gesamtstreckenlänge zurückgelegt werden. Deshalb ist ein Durchfahren von bereits erkundeten Gebieten zu vermeiden.

e) **Kooperation**

Die Explorationsstrategie soll auf ein verteiltes System, bestehend aus mehreren gleichzeitig operierenden Rovern, erweiterbar sein. Die Fahrzeuge sollen dazu ihre bereits erfassten Gebiete untereinander abgleichen, was eine kompakte Repräsentation für diese Daten erfordert. Das auf diese Weise kooperierende Gesamtsystem soll ebenfalls in der Lage sein, autonom zu agieren.

Wie bereits beschrieben, sollen als Ergebnis dieser Arbeit die notwendigen Komponenten einer integrierten Anwendung zur Simulation autonomer Navigationsverfahren basierend auf einer virtuellen Planetenoberfläche entwickelt werden. Für die Anforderungen an die Pfadplanungsalgorithmen bleibt daher das Sensorsystem für die Umgebungserfassung und die Vorverarbeitung (z. B. der Stereo-Vision-Prozess oder die Merkmalsextraktion und Klassifizierung) unberücksichtigt. Die notwendigen Informationen werden direkt aus dem Geländemodell abgeleitet. Für eine erweiterte Analyse des Navigationsprozesses bietet die 3D-Geländesimulation die Möglichkeit, diese Verarbeitungsschritte ebenfalls zu integrieren. Durch die genaue Kenntnis der Modellkoordinaten kann die Simulationsumgebung insbesondere dazu genutzt werden, verschiedene Verfahren zu evaluieren und untereinander zu vergleichen.

Des Weiteren bleiben die Messungenauigkeiten der Sensoren und damit das SLAM-Problem (vgl. Abschnitt 3.1.3) unberücksichtigt. Die Roverposition wird immer als genau bekannt angenommen. Diese Einschränkung stellt ebenfalls eine Vereinfachung dar und muss nicht notwendigerweise gelten. Visuelle Odometrieverfahren wie das Bundle Adjustment (siehe hierzu Abschnitt 3.2.2) können ebenfalls in den Simulationsprozess eingebunden werden. So zeigt

z. B. die Arbeit von [Lin10], in der auf Grundlage der hier beschriebenen Geländemodelle ein Koordinaten-Matchingverfahren zur Selbstlokalisierung von Rovern vorgeschlagen wird, dass die Integration verschiedener Algorithmen in den simulierten Navigationsprozess möglich ist.

Betrachtet man das Hauptziel für das Explorationskonzept, ein Gebiet flächendeckend zu erfassen, so liegt zunächst als Strategie nahe, das Gelände in ein Raster aufzuteilen und systematisch (z. B. zeilenweise) zu erkunden. Dieser triviale Ansatz hat aber entscheidende Nachteile. So ergeben sich Probleme, wenn einzelne Rasterzellen unpassierbar sind, besonders ungünstig sind labyrinthartige Umgebungen. Darüber hinaus ist es schwierig, mehrere Rover untereinander zu koordinieren. Möglich wäre, jedes Fahrzeug ausschließlich in einem Teilgebiet einzusetzen. Insgesamt erhält man damit aber ein sehr statisches Verfahren. Es existieren kaum Möglichkeiten, operationelle Vorgaben zu berücksichtigen. Aufgabenstellungen könnten sein, dass alle Rover zusammen ein bestimmtes Teilgebiet sehr schnell und detailliert erfassen oder dass sich andererseits alle Fahrzeuge maximal voneinander entfernen, um einen groben Überblick über eine große Fläche zu geben. Soll zudem ein Wechsel zwischen derartigen Operationsszenarien möglich sein, so ist dafür ein statisches Rasterverfahren unbrauchbar.

Das Prinzip der Rastereinteilung kann dennoch als Grundlage für Explorationsstrategien und Kartierungsaufgaben dienen. Ein Beispiel hierfür ist das Occupancy-Grid-Verfahren. Das gesamte Gelände wird in sehr kleine Zellen (je nach Anwendung mit einer Größe von 5–50 cm) eingeteilt. Auf Grundlage der Sensordaten werden die Zellen als erfasste Gebiete gekennzeichnet. Darüber hinaus werden zusätzliche Informationen gespeichert, die für die Pfadplanung relevant sind, also ob es sich z. B. um ein Hindernis handelt oder wie die Untergrundbeschaffenheit klassifiziert wurde. Diese Darstellung der Umgebung, die überdies sehr robust gegenüber Lokalisierungs- und Messfehlern ist, eignet sich gut für Verfahren der Frontier-Based-Exploration (siehe Abschnitt 3.2.4). Ein Nachteil ist jedoch der Speicherbedarf der Kartendaten, insbesondere wenn es darum geht, die Umgebungsinformationen unter mehreren Rovern zu teilen. Auch wenn es Ansätze gibt, die Rasterkarten z. B. mit Hilfe der Wavelettransformation zu komprimieren [Ygu05], erscheint eine hybride Methode unter Verwendung von sowohl metrischen als auch topologischen Modellen für die ExoPlanet-Zielstellung geeigneter.

5.1.2 Das SRT-Verfahren

Als Grundlage für die Entwicklung eines ExoPlanet-Explorationskonzepts soll eine Methode dienen, die auf dem allgemeinen Rapidly-Exploring Random Tree (RRT) Prinzip (siehe z. B. [LaV06]) beruht und insbesondere für sensorbasierte Systeme entwickelt wurde. Beim Sensor-based Random Tree (SRT) Verfahren [Ori04; Fre05] werden zunächst die Sensordaten der Umgebung ausgewertet und wie bei der Frontier-Based-Exploration die Grenzbereiche selektiert, in deren Richtung sich noch unbekannte Bereiche befinden. Wie Abbildung 25a illustriert, wird danach (zufällig) ein Punkt auf dieser Grenzlinie bestimmt, der das Ziel für den nächsten Streckenabschnitt darstellt. Hier angekommen, werden erneut Sensordaten erfasst und ein weiterer Zielpunkt ausgewählt.

Aus den Standorten, an denen jeweils die lokale Umgebung erfasst wurde, und den an diesen Positionen ausgewählten Zielpunkten kann eine Baumstruktur aufgebaut werden. Sollte sich während der Gebietserfassung an einer Stelle kein Grenzbereich mehr zu unbekanntem Bereichen finden, so wird auf Grundlage dieses Explorationsbaumes zum jeweiligen Vorgängerstandort zurückgekehrt, um hier einen noch zu erkundenden Punkt auszuwählen. Abbildung 25b zeigt die zurückgelegte Strecke während einer Innenraumerfassung. Zu erkennen ist auch eine Position, zu der zurücknavigiert wurde, da sich in einer Richtung keine neuen Bereiche mehr erschlossen haben.

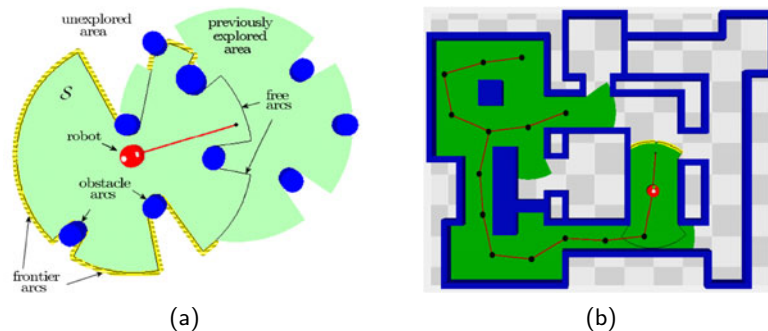


Abbildung 25. Prinzip der SRT-Exploration. (a) Dargestellt sind die Ergebnisse einer Abstandsmessung als erfasster Bereich (grün) zwischen mehreren Hindernissen (blau). Aus dem Grenzbereich zu der noch unbekanntem Region (gelb) wird der nächste Zielpunkt ausgewählt. (b) Innenraumerkundung basierend auf dem SRT-Verfahren. Dargestellt sind der bereits erfasste Bereich (grün) und die zurückgelegte Strecke als Baumstruktur. Quellen a, b: [Fre07].

Lokal arbeitet das SRT-Verfahren auf einem metrischen Umgebungsmodell. Als Grundlage dient der vom Sensorsystem erfasste Bereich, der auch als Local Safe Region (LSR) bezeichnet wird (Abbildung 26a). Für den beschriebenen Explorationsbaum, der als topologische Repräsentation des erkundeten Gebiets betrachtet werden kann, ist die Information über die Form dieser Fläche notwendig, um die Grenzbereiche zu unbekanntem Bereichen zu bestimmen. Allerdings genügt dafür eine vereinfachte Beschreibung. Das Originalverfahren sieht für eine reduzierte Darstellung der Umgebungsdaten zwei Möglichkeiten vor. Die Erste besteht darin, den Radius bis zum ersten auftretenden Hindernis als Parameter zu verwenden, was als SRT-Ball bezeichnet wird (Abbildung 26b). Als Nachteil ergibt sich aber, dass die erfasste Umgebung jeweils unterrepräsentiert wird und sich somit die notwendige Gesamtstrecke für die Exploration erhöht. Eine Alternative ist die SRT-Star-Form, die sich durch Aufteilung des Messkreises in Winkelabschnitte mit jeweils einem Abstandsparameter ergibt (Abbildung 26c).

Das SRT-Explorationsverfahren soll im Folgenden aufgegriffen werden, um eine Explorationsstrategie für die Anforderungen der ExoPlanet-Mission zu entwickeln. Zunächst werden aber nochmals die wesentlichen Argumente aufgeführt, die zu der Auswahl dieses Verfahrens geführt haben. Wie bereits erwähnt, sind die Methoden, die mit einem metrischen oder rasterbasierten Umgebungsmodell arbeiten, auf Grund des Speicherbedarfs, insbesondere mit Hinblick auf eine verteilte Anwendung, nicht geeignet. Zudem ist für diese Karten in Matrixdarstellung der Rechenaufwand für objektbezogene Operationen hoch (z. B. die Detektion von Bereichs-

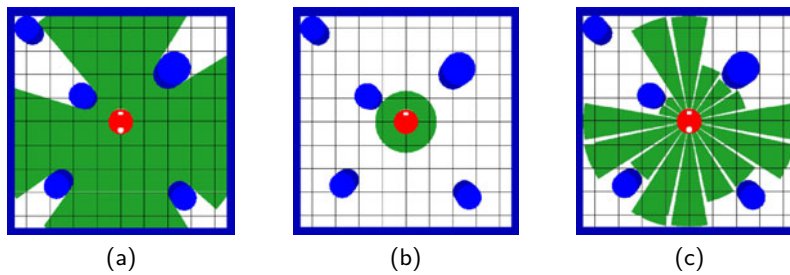


Abbildung 26. Repräsentation der metrischen Umgebung. (a) Der von den Sensoren erfasste Bereich. (b) Die Repräsentation der erkundeten Umgebung als Kreis. (c) Die Modellierung durch Kreissegmente. Quellen a–c: [Fre07].

überschneidungen). Mit der Parametrisierung, wie sie für topologische Modelle vorgenommen wird, sind indes schnelle Überprüfungen möglich. Bei vielen hybriden Verfahren ist jedoch die Methode für diese Informationsreduzierung (die meiste auf stochastischen Klassifikationsverfahren beruht) sehr komplex. Unter diesen Aspekten ergeben sich als Motivation für die Verwendung des SRT-Ansatzes folgende Punkte:

- **Hybride Umgebungsmodellierung**
Aus den hoch aufgelösten lokalen Sensordaten wird eine kompakte Baumstruktur abgeleitet. Eine Speicherung größerer Datenmengen ist nicht notwendig.
- **Einfaches Grundprinzip**
Die topologische Repräsentation der erkundeten Bereiche basiert auf wenigen Parametern (z. B. für einen Kreis oder Kreissegmente). Diese sind einfach, d. h. ohne hohe Anforderungen an Rechenleistung, zu bestimmen.
- **Erweiterungsfähig für eine Anwendung im freien Gelände**
Das Grundprinzip des SRT-Verfahrens ist sehr allgemein formuliert und deshalb auch auf komplexe Umgebungen anwendbar.
- **Einsetzbar für eine dezentrale Koordinierung mehrerer Systeme**
Der sich bei diesem Verfahren ergebende Explorationsbaum ist eine sehr kompakte Datenstruktur, die auch über geringe Kommunikationsbandbreiten zwischen mehreren Systemen austauschbar ist.

Vor allem die zwei zuerst genannten Punkte, aus denen sich ein insgesamt geringer Bedarf an Speicher- und Rechenkapazität für dieses Verfahren ableiten lässt, sind wichtige Aspekte im Zusammenhang mit den begrenzten Hardwareressourcen der Rover. Bezüglich der Erweiterungsfähigkeiten weisen die Autoren in der Diskussion zum SRT-Verfahren selbst auf die bestehenden Möglichkeiten hin: „In principle, the SRT method can be used for sensorbased exploration of high-dimensional configuration spaces [...]. The main issue would become the generation of a configuration-space Local Safe Region from the available sensor measures.“ [Ori04] Auch eine Anwendung des Verfahrens für verteilte Systeme aus mehreren Explorationssonden ist bereits gezeigt worden [Fra07]. Dabei wurde eine gemeinsame Datenstruktur verwendet, die zwischen allen beteiligten Fahrzeugen synchronisiert wird. Der Sensor-based Random Tree (SRT) wurde dabei zu einem Sensor-based Random Graph (SRG) erweitert.

5.1.3 Beschreibung der ExoPlanet-Explorationsstrategie

Im Folgenden wird die Explorationsstrategie beschrieben, die bezüglich der Anforderungen der ExoPlanet-Mission aus dem SRT-Verfahren abgeleitet wurde. Als Grundlage für die Pfadplanung wird vom Sensorsystem die lokale Umgebung erfasst. Bei planetaren Rovern werden dazu in der Regel Stereo-Kamerasysteme eingesetzt. Es wird davon ausgegangen, dass für das Umgebungsmodell bereits eine Segmentierung und Klassifizierung bezüglich der relevanten Geländekriterien vorgenommen wurde.

Abbildung 27a zeigt ein Rovermodell in einer planetaren Umgebung, die mit Hilfe der in Kapitel 4 vorgestellten Geländesimulation generiert wurde. Die erfasste Umgebungskarte, die ebenfalls von der Simulationssoftware bereitgestellt wird, ist in Abbildung 27b dargestellt. Die Rasterdaten haben eine Auflösung von 0,5 m pro Pixel, der maximale Messabstand beträgt 50 m. Der vom Rover befahrbare Bereich ist grün eingefärbt. Rot sind Hindernisse (große Steine) gekennzeichnet und orange die Bodenunebenheiten in Form von kleineren Steinen. Treten Anstiege auf, so wird je nach Winkel ein über Hellgrün nach Gelb verlaufender Farbbereich verwendet (z. B. beim Kraterrand in Abbildung 27b und 28a). Eine weitere simulierte Geländeeigenschaft ist die Untergrundbeschaffenheit. Je ungünstiger eine Stelle bezüglich dieses Parameters eingeschätzt wird (z. B. bei sandigem Boden), desto weiter verschiebt sich der zugehörige Farbwert nach Dunkelgrün (siehe als Beispiel Abbildung 28c).

Als erster Verarbeitungsschritt werden mit Hilfe eines Erosionsfilters alle vereinzelt und damit unsicheren Messpunkte entfernt. Eine Dilatationsoperation erweitert die Hindernisbereiche, da die direkte Umgebung der Hindernisse auf Grund einer Kollisionsgefahr ebenfalls nicht befahren werden darf. Für die Bodenunebenheiten wird ein glättender Filterkern verwendet, da sich eine kontinuierliche Verteilung bei diesem Parameter stabilisierend auf die Pfadsuche auswirkt, eine mögliche Route besitzt dann auch einen glatteren Verlauf. Das Ergebnis dieser Filteroperationen ist in Abbildung 27c dargestellt. In Analogie zum SRT-Verfahren kann der resultierende Navigationsbereich als sichere lokale Umgebung bezeichnet werden.

Im nächsten Schritt wird durch ein einfaches Segmentierungsverfahren das zusammenhängende Gebiet um die Roverposition herum ermittelt. Damit erhält man die Menge aller erreichbaren Punkte, die in Abbildung 27d als weiß umrandeter Bereich dargestellt ist (für weitere Beispiele siehe Abbildungen 28c und d). Die sichere Umgebung wurde nochmals auf die erreichbare lokale Umgebung reduziert. Die Liste der Randpunkte dieses Bereichs umfasst gleichzeitig auch die potenziellen Zielpunkte für den nächsten Navigationsabschnitt. Aus dieser Liste wird daher zufällig eine kleinere Anzahl von Zielkoordinaten (10–20) ausgewählt.

Unter Anwendung des Dijkstra-Algorithmus (siehe Abschnitt 3.3.1) wird daraufhin jeweils der optimale Pfad zwischen der aktuellen Position (im Mittelpunkt der Navigationskarte) und diesen Zielkoordinaten ermittelt. Optimal heißt dabei, dass die Bewertung bzw. die Kostenfunktion (siehe Abschnitt 3.2.3) der Strecke gegenüber allen anderen möglichen Routen minimal

ist. Als Kriterien fließen in diese Bewertung mit ein die Streckenlänge, die Bodenebenheiten, der Geländeanstieg und die Untergrundbeschaffenheit. Wird für den Anstieg oder die Untergrundbeschaffenheit ein Maximalwert überschritten oder wurde ein Hindernis erkannt, so werden die entsprechenden Rasterzellen von der Pfadplanung ausgeschlossen. Als Beispiel ist in Abbildung 27d ein möglicher Pfad zu einem Zielpunkt eingezeichnet.

Das Ziel für einen Streckenabschnitt muss beim SRT-Verfahren außerhalb des erfassten Bereichs liegen. Die notwendige Überprüfung basiert auf dem Explorationsbaum, der alle lokalen Umgebungen als vereinfachte Formen enthält. Hierzu wird eine kreisförmige Repräsentation verwendet. Um den Radius zu ermitteln, wird für alle Randpunkte der erreichbaren Umgebung der Abstand zum Rover bestimmt und daraus der Mittelwert berechnet. In Abbildung 27d ist der resultierende Kreis eingezeichnet. Die Pfadliste zu den zufällig ausgewählten Zielen wird bezüglich der Gesamtpfadbewertung (oder besser nach dem Verhältnis aus Pfadlänge und Bewertung) sortiert, da Routen mit einfachen Navigationsmanövern vorzuziehen sind. In dieser Reihenfolge wird für jedes Streckenziel geprüft, ob sich dieses innerhalb der Kreisregionen des Explorationsbaums befindet. Ist das nicht der Fall, so wird das Ziel für den nächsten Navigationsabschnitt verwendet. Befinden sich alle Zielpunkte der Liste innerhalb von erkundeten Bereichen, so wird zum vorherigen Standort zurückgekehrt. Die einzelnen Verarbeitungsschritte für die beschriebene Explorationsstrategie werden im Folgenden in einer algorithmischen Beschreibung nochmals zusammengefasst.

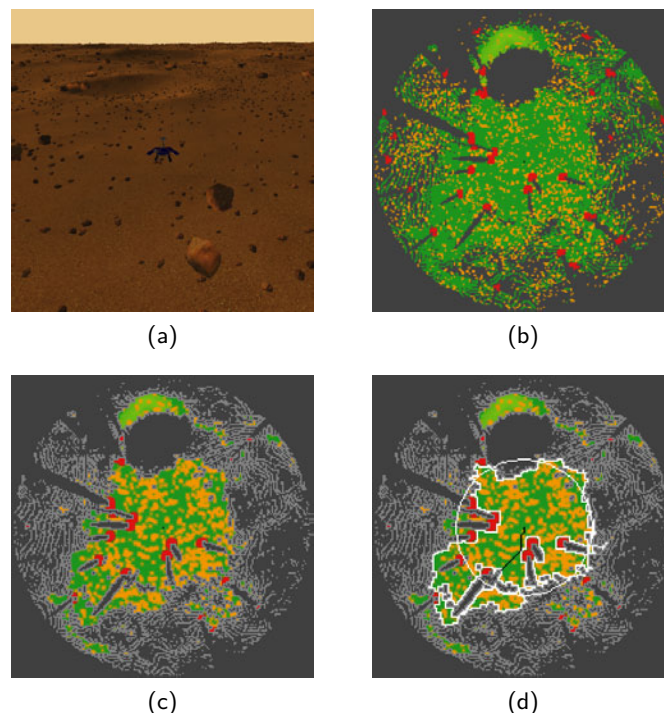


Abbildung 27. Prinzip der ExoPlanet-Explorationsstrategie. (a) Die Position des Rovers innerhalb des generierten Geländes. (b) Die lokale Umgebung basierend auf simulierten Sensordaten. (c) Der sichere Navigationsbereich. (d) Die erreichbare Umgebung (weiß umrandet), der mittlere Abstand (weißer Kreis) und der ausgewählte Pfad (schwarz). Quelle: eigene Darstellung.

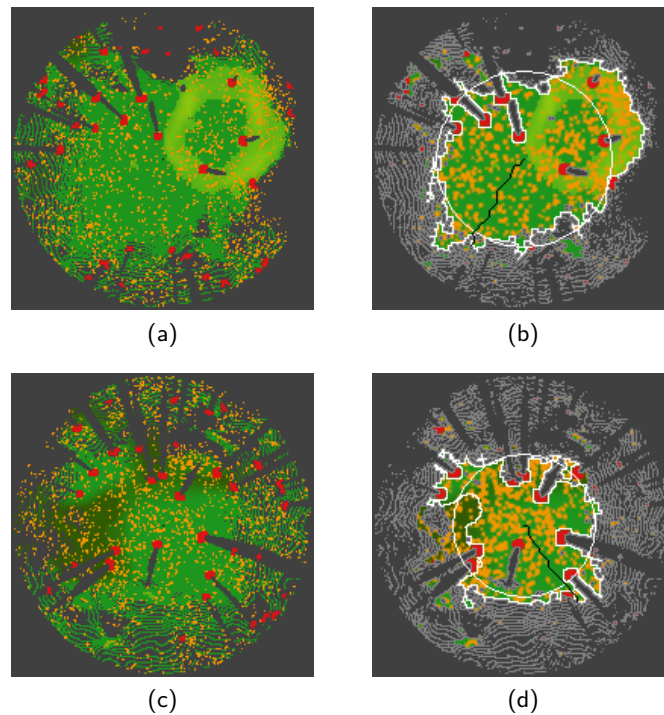


Abbildung 28. Weitere Beispiele für Navigationskarten. (a, c) Simulierte Sensordaten. (b, d) Die jeweils zugehörige sichere lokale Umgebung. Quelle: eigene Darstellung.

Die Exploration läuft in einer Schleife ab, in der jeweils das nächste Ziel und der optimale Pfad dahin bestimmt werden (siehe Algorithmus EXPLORATION: Zeile 4). Die Pfadplanung beginnt mit der Umgebungserfassung und einer Vorverarbeitung der Rasterdaten (Zeile 12, 13 bzw. 29). Basierend auf der resultierenden Karte werden die erreichbare Umgebung sowie der zugehörige Randbereich und dessen mittlerer Abstand von der aktuellen Position bestimmt (Zeile 14–16).

Anschließend werden zufällig einige Zielkoordinaten aus dem Randbereich ausgewählt und mit Hilfe des Dijkstra-Algorithmus jeweils der optimale Zielpfad durch die erfasste Umgebung ermittelt (Zeile 17–20). Die sich ergebende Liste von Pfaden wird bezüglich niedriger Gesamtstreckenkosten sortiert und in dieser Reihenfolge wird geprüft, ob das Pfadziel in dem vom Rover noch nicht erfassten Gebiet liegt (Zeile 21–24). Trifft dies zu, so wird der Explorationsbaum erweitert und der gefundene Pfad für den nächsten Navigationsabschnitt verwendet (Zeile 25, 26). Sollte sich unter den potenziellen Zielen keine Koordinate befinden, die nicht innerhalb des bereits erkundeten Gebiets liegt, so wird dem Explorationsbaum lediglich die Information über die aktuelle lokale Umgebung hinzugefügt. Ein weiterführender Pfad konnte für die aktuelle Position nicht ausgemacht werden. (Zeile 27, 28).

Wenn als Ergebnis der Pfadplanungsfunktion ein folgender Streckenabschnitt gefunden wurde, kann der Rover weiter diesen Pfad entlang navigieren (Zeile 7). Konnte bezüglich der aktuellen Position kein neues Ziel gefunden werden, so bewegt sich der Rover entlang der im Explorationsbaum gespeicherten Route zur vorherigen Position zurück (Zeile 9, 10), der Schlei-

fendurchlauf zur Pfadsuche beginnt daraufhin erneut. Bei der Rückwärtsnavigation bezüglich des Explorationsbaums ist noch darauf zu achten, dass für die Ausgangsposition (also den Wurzelknoten) keine Vorgängerposition existiert. Sollte der Rover wieder an dieser Stelle angekommen sein, so stoppt der Explorationsalgorithmus ebenfalls. Es kann dann aber davon ausgegangen werden, dass das komplette Gebiet hinreichend flächendeckend erkundet wurde.

Algorithmus 2 EXPLORATION

1:	Prozedur EXPLORATION :	'Hauptfunktion'
2:	$p \leftarrow$ LOKALISIERE	'initialisiere Position'
3:	$T \leftarrow T(p)$	'initialisiere Explorationsbaum'
4:	solange nicht 'Stopp' :	'Hauptschleife'
5:	$(P, T) \leftarrow$ PLANEPFAD(T)	
6:	wenn $P \neq$ NULL :	
7:	FAHRENTLANG(P)	'navigiere zum nächsten Ziel'
8:	sonst :	
9:	$P \leftarrow P(T.VORGÄNGER)$	
10:	FAHRERÜCKWÄRTSENTLANG(P)	'gehe zur vorherigen Position zurück'
11:	Prozedur PLANEPFAD(T) :	
12:	$M \leftarrow$ ERSTELLEKARTE	
13:	$M_S \leftarrow$ FILTER(M)	'ermittle sichere lokale Umgebung'
14:	$M_E \leftarrow M_E(M_S)$	'ermittle erreichbare lokale Umgebung'
15:	$M_R \leftarrow \partial M_E$	'ermittle Randpunkte'
16:	$r \leftarrow \emptyset$ ABSTAND(M_R)	'ermittle mittleren Radius'
17:	$Z \leftarrow$ ENTNEHMEX(M_R)	'wähle zufällig x Zielpunkte aus, z. B. $x=10$ '
18:	$\hat{P} \leftarrow \emptyset$	
19:	für alle $z \in Z$:	
20:	$\hat{P} \leftarrow \hat{P} \cup$ DIJKSTRA(M_E, z)	'ermittle jeweils den optimalen Pfad'
21:	solange $\hat{P} \neq \emptyset$:	'für alle potenziellen Pfade'
22:	$P \leftarrow$ ENTNEHME-MIN _{K} (\hat{P})	'Reihenfolge sortiert nach Pfadkosten'
23:	$z \leftarrow Z(P)$	'wähle den Zielpunkt des Pfades aus'
24:	wenn $z \notin A(T)$:	'liegt Ziel im unbekanntem Gebiet?'
25:	$T \leftarrow T.PLUS(r, P)$	'erweitere Explorationsbaum'
26:	Rückgabe : (P, T)	'Pfad gefunden'
27:	$T \leftarrow T.PLUS(r)$	'erweitere Explorationsbaum ohne Pfad'
28:	Rückgabe : (NULL, T)	'kein Pfad gefunden'
29:	Prozedur FILTER(M) :	
30:	$M \leftarrow$ EROSION(M)	'entferne unsichere Messwerte'
31:	$M \leftarrow$ DILATATION(M)	'erweitere Hindernisbereiche'
32:	$M \leftarrow$ GLÄTTUNG(M)	'kontinuierlicher Verlauf für Unebenheiten'
33:	Rückgabe : M	

p : Position; T : Explorationsbaum; M : lokale Umgebungskarte; M_S : sichere Umgebung; M_E : erreichbare Umgebung; M_R : Randbereich von M_E ; r : mittlerer Abstand zu M_R ; P : Pfad; \hat{P} : Pfadliste; MIN _{K} : minimaler Wert der Kostenfunktion; Z : potenzielle Ziele; z : Zielkoordinate

Der Dijkstra-Algorithmus wurde für dieses Verfahren ausgewählt, da für mehrere Ziele bei gleicher Ausgangsposition ein Pfad bestimmt werden soll. Der Effizienzvorteil der A*-Methode geht dadurch verloren. Für ein verteiltes System wird der Explorationsbaum untereinander ausgetauscht. Die Überprüfung, ob eine Position im bereits erkundeten Gebiet liegt (EXPLORATION: Zeile 24), wird dann bezüglich aller Explorationsbäume durchgeführt.

5.1.4 Integration von Geländesimulation und Navigationsanwendung

Im folgenden Abschnitt wird beschrieben, wie für die praktische Umsetzung und die Evaluation des erarbeiteten Explorationskonzepts, eine Integration der Navigationsalgorithmen mit der Geländesimulation vorgenommen wurde. Dazu wurde die Softwareentwicklung in zwei Teilprojekte gegliedert. Unter der Bezeichnung *ExoSim* wurden alle notwendigen Komponenten zusammengeführt, die für die Geländegenerierung und Sensorsimulation notwendig sind. Das Projekt *ExoPath* beinhaltet die Module zur Simulation der Navigationsalgorithmen und greift dazu auf *ExoSim* zurück. Abbildung 29 zeigt eine Übersicht über alle Komponenten.

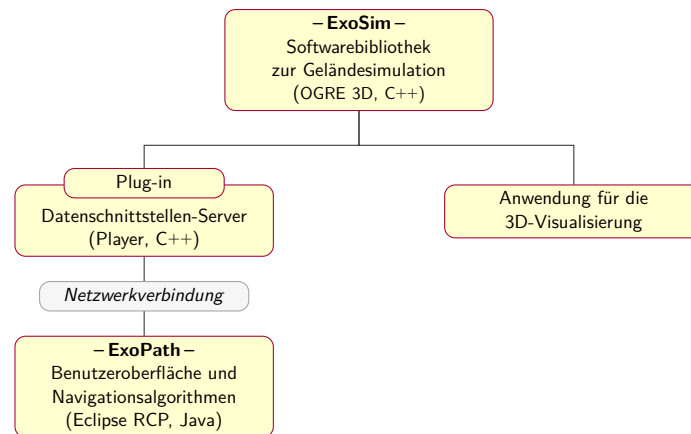


Abbildung 29. Komponenten der integrierten Simulationsplattform. Die Geländesimulation kann sowohl als eigenständige Binärmodule zur 3D-Visualisierung verwendet werden als auch als Plug-in für den Player-Server, der netzwerktransparent allgemeine Roboter-Datenschnittstellen zu Verfügung stellt. Quelle: eigene Darstellung.

Das Ergebnis der ExoSim-Entwicklung, die wie in Kapitel 4 beschrieben auf OGRE 3D basiert, sind zwei verschiedene Binärmodule. Als eigenständige Applikation (*ExoSim.exe*) ausgeführt, wird ein Anwendungsfenster geöffnet, in dem sich ein Benutzer interaktiv und in Echtzeit durch die generierte Landschaft bewegen kann. In dieser Form kann die ExoSim-Anwendung als 3D-Visualisierungsprogramm für räumliche Umgebungen bzw. raumbezogene Informationen dienen. Darüber hinaus wird die Geländesimulation als Softwarebibliothek (*ExoSim.dll*) zur Verfügung gestellt. Es können darin verschiedene Funktionen aufgerufen werden, um den Simulationsprozess zu starten und zu steuern. Außerdem ist der Zugriff auf Daten, wie die gerenderten Bilder oder die Modellkoordinaten, möglich. Somit können externe Softwarekomponenten mit der Simulationsumgebung kommunizieren.

Um die möglichen Funktionsaufrufe der ExoSim-Bibliothek in eine einheitliche Schnittstellenbeschreibung zu kapseln, wurde auf das Open Source Projekt Player [45] zurückgegriffen. Das Konzept für die Player-Entwicklung beinhaltet einen Server, der allgemeine Schnittstellen für Geräte aus dem Robotikbereich zur Verfügung stellt. Für jede Schnittstelle sind Formate zur Kommandierung und für den Datenaustausch festgelegt. Beispiele für solche Module sind Kameras, Abstandssensoren oder Positionierungseinheiten, aber auch reine Datenbeschreibun-

gen wie 2D/3D-Punktwolken oder Navigationskarten (für eine ausführliche Schnittstellenbeschreibung siehe [46]). Die Idee hinter diesem Ansatz ist, den Schnittstellen austauschbare Gerätetreiber zuzuordnen. So können Algorithmen unabhängig von der Hardware entworfen und anschließend auf verschiedenen Systemen oder basierend auf (teilweise) simulierten Daten getestet werden.

Die ExoSim-Bibliothek ist als Plug-in in den Player-Server integriert und kann dadurch über das Player-Protokoll gesteuert werden. Ein weiterer Vorteil, der sich aus dieser Integration ergibt, ist die netzwerktransparente Kommunikation. Die Geländesimulation, die eine rechenintensive Anwendung darstellt, kann somit auf einem dafür optimierten Rechnersystem laufen und über eine Netzwerkverbindung z. B. mit der Benutzeroberfläche kommunizieren und Daten austauschen. Für die Navigationsalgorithmen ist es sogar möglich, diese auf der Roverhardware auszuführen, wobei dennoch simulierte oder auch nur teilweise simulierte Sensorinformationen verwendet werden können.

Wie bereits erwähnt, ist die ExoPath-Software ebenfalls modular aufgebaut. Sie besteht im Wesentlichen aus den Komponenten für die Benutzeroberfläche (Graphical User Interface GUI) und den Navigationsalgorithmen. Dazu kommt ein Clientmodul zur Kommunikation mit dem Player-Server. Für die Explorationssimulation werden mehrere Player-Schnittstellen verwendet, z. B. für Kameradaten, die Position und Ausrichtung des Rovers sowie das Punktdatenformat für die Umgebungskarten. Der prinzipielle Ablauf der Kommunikation zwischen den beteiligten Softwarekomponenten für einen Simulationslauf wird in Abbildung 30 illustriert. Dabei wird zunächst die Initialisierungsphase beschrieben (Abschnitt Init) und darunter der Datenaustausch für das Explorationsverfahren (Abschnitt Explore).

Die Kommunikation beginnt mit einem Verbindungsaufbau zum Player-Server (**connect**). Anschließend werden von der Client-Komponente die Schnittstelleninformationen und verschiedene Initialisierungsdaten erfragt (**request**). Mit der Herstellung einer Verbindung blockiert der Serverprozess und lädt das ExoSim-Plug-in, welches daraufhin gestartet wird. Nachdem sich die Simulationsumgebung initialisiert hat, beantwortet der Server die ausstehenden Anfragen des Clients (**reply**). Dieser ist dadurch in Kenntnis gesetzt, dass das Simulationsmodul bereit ist, und benachrichtigt daraufhin alle anderen beteiligten Komponenten (**notify**). Darunter z. B. die Benutzeroberfläche oder die Navigationsanwendung, die daraufhin Umgebungsdaten abfragen kann.

Während der simulierten Exploration wird der Rover durch die generierte Landschaft bewegt (**moveTo/setPosition**) und es müssen Umgebungsdaten abgefragt werden (**getData**). Bei einer Datenanfrage löst der Server im ExoSim-Modul einen neuen Renderingdurchlauf aus und stellt im Anschluss daran Datenpakete für die entsprechenden Schnittstellen zusammen (**packData**). Diese werden separat an den Client übertragen, z. B. in Form von Kamerabildern (**publishCamImg**) oder Punktdaten der Umgebung (**publishRangePoints**). Die Daten werden empfangen und es erfolgt wiederum die Benachrichtigung der übrigen Module. Wie im vorigen Abschnitt beschrieben, kann der Navigationsalgorithmus auf Basis dieser Eingabedaten

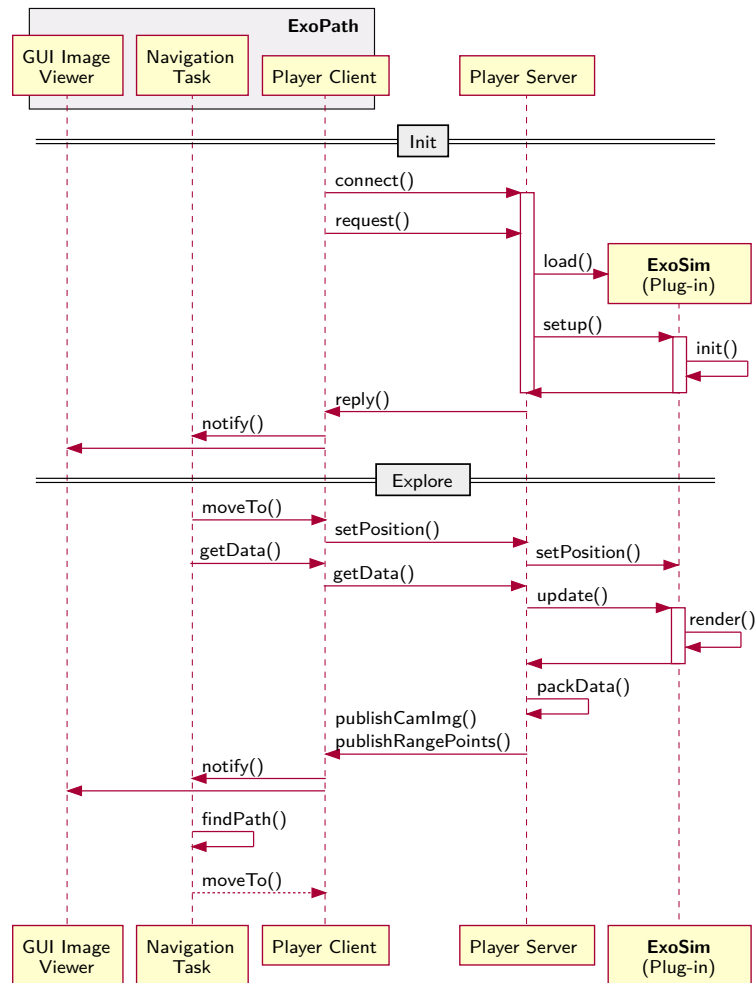


Abbildung 30. Sequenzdiagramm für den Simulationsprozess. Dargestellt ist die Kommunikation zwischen den wichtigsten Softwaremodulen während der Initialisierungsphase und für den Explorationsprozess. Quelle: eigene Darstellung.

einen neuen Zielpfad bestimmen (`findPath`) und den Rover erneut in Bewegung setzen. Zur Überprüfung oder Visualisierung der Navigation ist es möglich, das Rovermodell auf jeden einzelnen Pfadpunkt zu setzen. In diesem Fall würde eine komplette Umgebungskarte nur am Ende jedes Streckenabschnitts angefragt und generiert.

Zur Überwachung der Closed-Loop-Simulation wurde für die ExoPath-Anwendung eine Benutzeroberfläche entwickelt, mit der unterschiedliche Kamerabilder oder Informationen über den Verlauf des Explorationsprozesses dargestellt werden können. Um die gesamte Funktionalität für ein aufteilbares Anwendungsfenster und die damit verbundene Interaktion nicht neu implementieren zu müssen, wurde als Grundlage die Eclipse Rich Client Platform [19] verwendet. Mit diesem Framework ist es auf einfache Weise möglich, eine Benutzeroberfläche mit mehreren Ansichten zu entwickeln, die beliebig angeordnet werden können. Das ExoPath-Anwendungsfenster ist in Abbildung 31 dargestellt. Als Teilfenster stehen hierbei zur Verfügung: 1. ein simuliertes Umgebungsbild der Roverkamera, 2. eine Draufsicht auf das Gelände bezüglich der aktuellen Position, 3. die erfasste Umgebungskarte mit den Ergebnissen des Na-

vigationsalgorithmus sowie 4. eine Abbildung des gesamten Gebietes mit einer Darstellung der bereits erfassten Bereiche und der zurückgelegten Gesamtstrecke, abgeleitet aus dem Explorationsbaum. Der Gebietsanzeige liegt das MRO-Orthobild zu Grunde, wobei in der Anzeige ein schnelles Verschieben und Zoomen für das hoch aufgelöste Bild möglich ist.

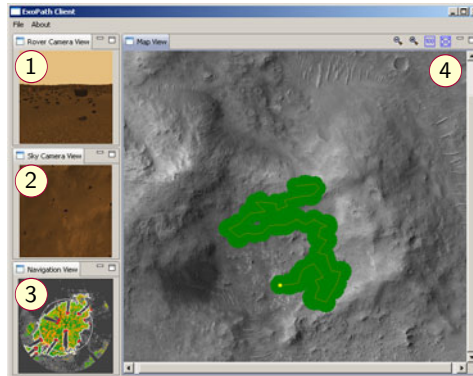


Abbildung 31. Die ExoPath-Benutzeroberfläche. Die einzelnen Teilfenster zeigen 1. das Bild der Roverkamera, 2. eine Geländedraufsicht bezüglich der aktuellen Position, 3. die Sensordaten mit Ergebnissen des Navigationsalgorithmus und 4. eine Darstellung des gesamten Explorationsgebiets und des bereits erfassten Bereichs. Quelle: eigene Darstellung.

Die Entwicklung der Benutzeroberfläche bzw. der ExoPath-Komponenten erfolgte in der Programmiersprache Java – im Unterschied zur Geländesimulation, die in C++ geschrieben ist. Für die Client-Server-Kommunikation, um auf die Player-Schnittstellen zugreifen zu können, wurde in die GUI ein weiteres ein Open Source Projekt integriert. Der Javaclient [25] implementiert bereits die verschiedenen Datentypen und übernimmt die Codierung der Daten für die Netzwerkübertragung.

5.2 Durchführung der Simulation

Im Folgenden werden die Ergebnisse der durchgeführten Closed-Loop-Simulation für die autonome Rovernavigation vorgestellt. Für die Umgebungserfassung wurde ein Kamerasystem simuliert, welches sich in einer Höhe von 1,40 m befindet. Diese Höhe entspricht der Masthöhe der Mars Exploration Rover, liegt aber auch für Mikro-Rover im konstruktiv möglichen Bereich. So hat die Versuchsplattform Rocky 7 der NASA eine Masthöhe von 1,50 m [Mat97].

5.2.1 Exploration mit einem Rover

Ein erster Versuch wurde mit einem einzelnen Explorationsrover durchgeführt. Daher wurde das Missionsgebiet auf eine Breite von 1 km reduziert. Abbildung 32a zeigt die Fläche, welche dem rechten oberen Viertel des ExoPlanet-Missionsgebiets entspricht. Daneben in Abbildung 32b ist noch einmal der gleiche Bereich als farbiges Höhenprofil abgebildet.

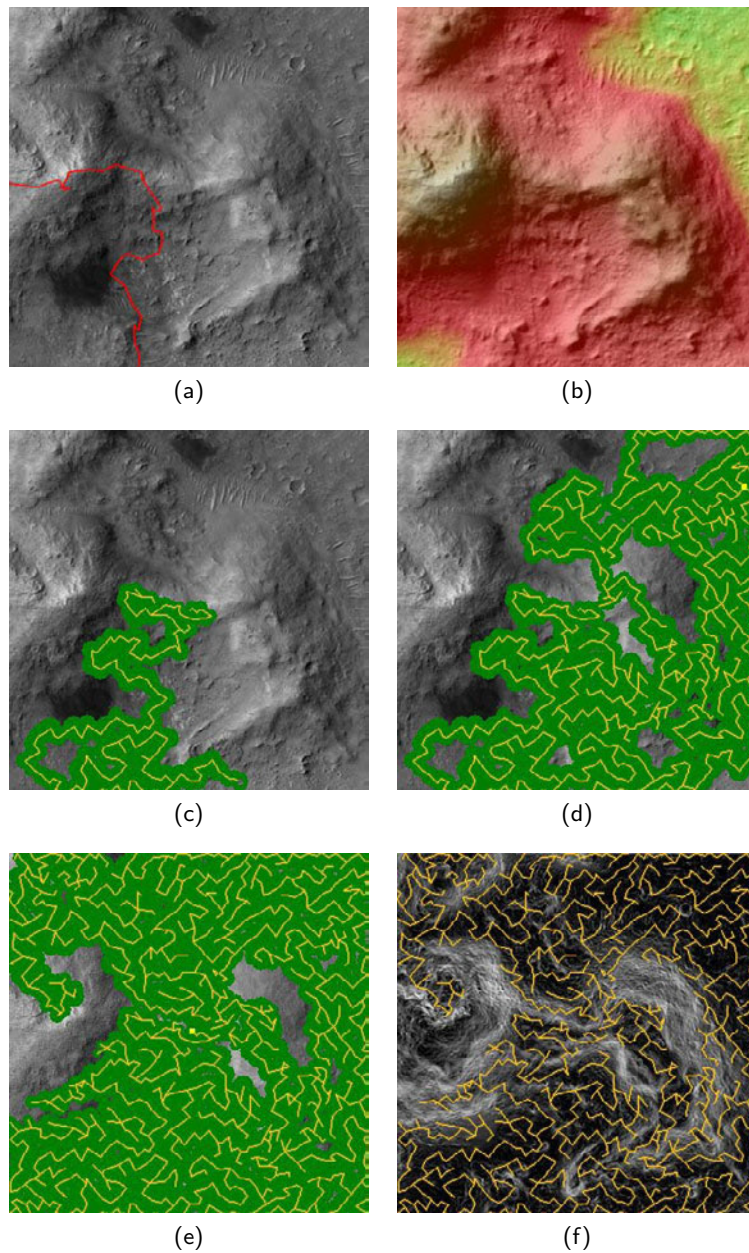


Abbildung 32. Ergebnis der simulierten Exploration. (a) Das zu Grunde liegende Explorationsgebiet mit einer Abmessung von $1\text{ km} \times 1\text{ km}$. Zudem ist die Route des MER-Rovers Spirit eingezeichnet. (b) Das Zielgebiet als farbiges Höhenprofil. (c) Eine Momentaufnahme der autonomen Exploration nach 5 km und (d) nach 25 km zurückgelegter Gesamtstrecke. (e) Das insgesamt erkundete Gebiet. (f) Die Explorationsroute im Vergleich zum Geländegradien. Quelle: eigene Darstellung basierend auf [38] und [33].

Des Weiteren ist für den simulierten Rover in den Abbildungen der Verlauf der Geländeerfassung dokumentiert, wobei jeweils nach 5 km und 25 km zurückgelegter Gesamtstrecke das vom Explorationsfahrzeug erfasste Gebiet dargestellt ist (Abbildungen 32c und d). In diesen Momentaufnahmen zeigen sich bereits einige Leistungsmerkmale des zu Grunde liegenden Verfahrens. Mit Hilfe einer autonomen Navigationsstrategie ist es möglich, eine gute Flächenab-

deckung für ein planetares Explorationsgebiet zu erreichen. Abhängig von der Sichtweite der Sensoren kommt es nicht zu Überschneidungen bei der Geländeerfassung. Gleichzeitig wurden bei der Navigation auf Hindernisse, zu große Anstiege oder die Untergrundbeschaffenheit Rücksicht genommen. Bereits nach 5 km zurückgelegter Strecke konnte der simulierte Rover damit ein relativ großes Gebiet erfassen. Unter Berücksichtigung der Anforderungen (siehe Abschnitt 5.1.1) zeigt sich in einem ersten Fazit bereits die Eignung des vorgestellten Verfahrens für diese Explorationsaufgabe.

Geht man bei der Tiefenerfassung mit Hilfe eines Stereo-Kamerasystems von einer zuverlässigen Messung bis zu einem Abstand von 20 m aus, erhält man jeweils eine lokale Umgebung mit einem Durchmesser von 40 m. Um damit ein Gebiet von 1 km Kantenlänge systematisch als Raster zu erfassen, müssen ungefähr 25 km zurückgelegt werden. Diese Streckenlänge wurde daher auch für die zweite Momentaufnahme der Geländeexploration gewählt. Es ist zu erkennen, dass ein weiter Bereich des Zielgebiets erfasst wurde, aber noch keine flächendeckende Erkundung erreicht ist. Ein Grund dafür sind die notwendigen Ausweichmanöver um nicht befahrbare Geländeabschnitte. Eine Navigationsroute ist damit länger als die Punktverbindung zweier Koordinaten. Zudem ist es bei dem eingesetzten Verfahren möglich, dass der Rover in eine Sackgasse gerät und zurücknavigieren muss. Das kostet ebenfalls zusätzliche Streckenlänge, erlaubt aber eine autonome Routenplanung auch in labyrinthartigen Umgebungen.

In Abbildung 32e ist die Gesamtfläche dargestellt, die von dem simulierten Rover erfasst werden konnte. Der Navigationsalgorithmus wurde beendet, nachdem das Fahrzeug an seinen Ausgangspunkt zurückgekehrt ist. Es wurden alle entstandenen Richtungsarme des Explorationsbaums durchlaufen und hinsichtlich weiterer Erkundungsmöglichkeiten überprüft. Um nochmals das zu Grunde liegende Pfadplanungsverfahren zu verifizieren, wurde in Abbildung 32e vom Höhenmodell des Zielgebiets eine Gradientendarstellung abgeleitet. Der Grauwert der Pixel entspricht demnach qualitativ dem lokalen Geländeanstieg. Anhand dieser Darstellung ist zu erkennen, dass der Rover nicht in stark ansteigende oder abfallende Gebiete gefahren ist. Da es sich bei der ausgewählten Marsregion nicht um eine übermäßig bergige Landschaft handelt (der relative Höhenunterschied für das Gebiet beträgt ca. 100 m), wurde der maximal befahrbare Anstieg des Rovers auf 15 Grad begrenzt. Dass das ausgewählte Gelände durch einen Explorationsrover überwunden werden kann, hat die MER-Mission gezeigt. Bei dem Berg im linken Bildbereich handelt es sich um den Husband Hill, über dessen Gipfel der MER-Rover Spirit gefahren ist. Die dabei zurückgelegte Strecke ist zum Vergleich in Abbildung 32a dargestellt.

Zur Erkundung des gesamten Zielgebiets wurde eine Strecke von 60 km zurückgelegt. Dies stellt für einen Explorationsrover (insbesondere für einen Mikro-Rover in einem unwegsamen Gelände) eine sehr weite Entfernung dar. Dieser große Distanzwert hängt mit dem Ansatz zur Routenplanung beim SRT-Verfahren zusammen. Für die Navigation werden nur Daten der direkten Umgebung verwendet. Existiert bezüglich einer Position kein weiterführender Pfad, so wird zur vorherigen Stelle zurückgekehrt und ein neuer Versuch zur Erkundung des unbekann-

ten Bereichs unternommen. Durch diese Einschränkung auf lokale Informationen, zusammen mit der Repräsentation der Explorationshistorie als Baumstruktur, nimmt die Anzahl an Verbindungsstrecken (Navigationsabschnitte, auf denen kein neues Gebiet erschlossen wird) zu. Insbesondere zum Ende der Exploration, wenn sich der bereits erfassten Bereiche vergrößert, ist dieses Verhalten zu beobachten.

Am Ende der Gebietserkundung kehrt der Rover zu seinem Ausgangspunkt zurück. Betrachtet man also den gesamten Explorationsverlauf, so wurde durch das Zurückkehren jede Teilstrecke zwei Mal durchlaufen. Das würde für die Abschätzung bezüglich der Rastermethode 50 km Gesamtstrecke bedeuten (Ausweichmanöver sind dabei nicht berücksichtigt). Der Unterschied zu den 60 km, die beim SRT-Verfahren insgesamt zurückgelegt wurden, ist unter diesem Aspekt nicht mehr sehr groß. Dennoch ist eine Schlussfolgerung, dass für ein Verfahren mit zu minimierenden Gesamtstreckenlängen die Baumstruktur keine optimale Planungsgrundlage darstellt. Mögliche Ansätze zur Vermeidung von zu vielen Verbindungsstrecken werden am Ende dieses Kapitels im Rahmen der Erweiterungsmöglichkeiten diskutiert.

5.2.2 Exploration mit mehreren Rovern

Der zweite Versuch wurde basierend auf dem ExoPlanet-Missionsgebiet durchgeführt. Dazu wurden vier Explorationsrover simuliert, die gleichzeitig im Einsatz sind. Die Abbildungen 33b und c zeigt die zurückgelegte Strecke und das dabei erkundete Gebiet nach einer Gesamtdistanz von 25 km bzw. nach 100 km. Die Momentaufnahmen wurden wiederum gewählt, da nach 100 km theoretisch eine Erfassung der gesamten Fläche möglich ist. Bei dem Simulationslauf basierend auf dem SRT-Verfahren wurden dagegen insgesamt ca. 260 km zurückgelegt (vgl. Abbildung 33d). Dabei wurden jedoch immer nur sichere Pfadabschnitte gewählt, wie auch die wegen zu steilem Gefälle ausgelassenen Bereiche zeigen.

Am Ende der Exploration wurden vom ersten Rover (in Abbildung 33 grün dargestellt) insgesamt 100 km zurückgelegt. Der zweite Rover (blau) ist 60 km gefahren, der dritte (rot) 40 km und der vierte (magenta) ebenfalls 60 km. Der sich ergebene Unterschied in den Distanzwerten liegt am Verlauf der zufällig gewählten Streckenabschnitte. Wie bereits in Abbildung 33c zu erkennen, wurde das Gebiet für den mit Rot markierten Rover von zwei Seiten eingegrenzt, somit ist das Fahrzeug bereits früher zu seinem Ausgangspunkt zurückgekehrt, da keine unbekannt Bereiche mehr erschlossen werden konnten, ohne den bereits erfassten Bereich der anderen Fahrzeuge zu durchqueren. Dieses Verhalten ist jedoch mit der vereinfachten Implementierung zu begründen. Die Rover tauschen nur die Information über die bereits erfassten Bereiche aus, verwenden aber keine gemeinsame Datenstruktur, was prinzipiell möglich ist. Welche Ansätze zur Verbesserung hierfür bestehen, wird im Anschluss zusammen mit den generellen Schlussfolgerungen betrachtet.

Eine weitere Beobachtung ergibt sich im Vergleich der beiden Simulationsergebnisse mit dem Routenverlauf der MER-Rovers Spirit über den Husband Hill Gipfel (vgl. Abbildung 32a). So konnte beim ersten Versuch mit dem Einzelsystem ein Verbindungsstück dieser Route nicht gefunden werden. Im zweiten Durchlauf ist dagegen zu erkennen, dass der Gipfelbereich komplett erschlossen wurde. Das liegt wiederum an der Zufallsselektion, die dem Verfahren zu Grunde liegt. Weitere Versuche müssen zeigen, ob eventuell eine Methode, welche alle und nicht nur ausgewählte Möglichkeiten in Betracht zieht, trotz erhöhtem Berechnungsaufwand besser geeignet ist.

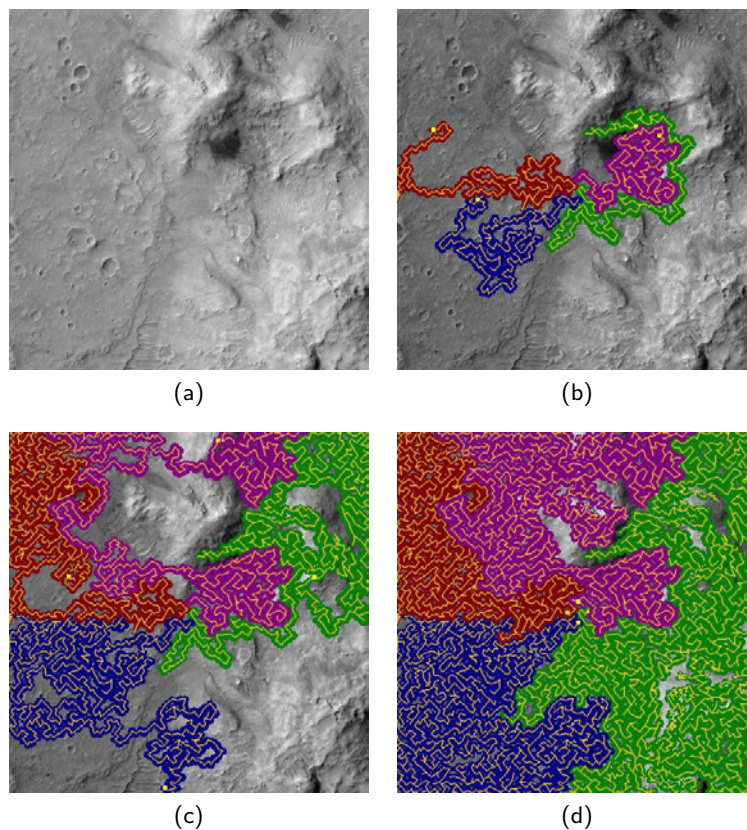


Abbildung 33. Ergebnis der simulierten Exploration mit mehreren Rovern. (a) Das ExoPlanet-Missionsgebiet mit einer Fläche von $2\text{ km} \times 2\text{ km}$. Als Momentaufnahmen der Simulation sind der erkundete Bereich und die zurückgelegte Gesamtstrecke der vier Rover nach (b) zusammen 25 km und (c) 100 km dargestellt. (d) Das insgesamt erfasste Gebiet. Quelle: eigene Darstellung.

6 Bewertung und Diskussion

In den Kapiteln 3 und 4 dieser Arbeit wurden die Grundlagen für die Entwicklung und Simulation einer planetaren Explorationsstrategie aufgezeigt. Darauf aufbauend wurde ein eigener Navigationsalgorithmus entworfen und verschiedene Softwarekomponenten implementiert, um die Funktionsweise des Verfahrens zu demonstrieren. Es ist eine integrierte Anwendung entstanden, die basierend auf einem generierten Landschaftsmodell Sensordaten als Eingabe für die simulierte Rovernavigation bereitstellt. Im Folgenden werden die Ergebnisse der präsentierten Simulationsläufe bewertet und diskutiert. Dazu werden zunächst die gestellten Anforderungen der jeweiligen Umsetzung gegenübergestellt. Anschließend erfolgt eine kritische Einschätzung der Leistungsfähigkeit und es werden darüber hinaus einige verfahrensverbessernde Erweiterungsmöglichkeiten aufgezeigt.

6.1 Bewertung der Ergebnisse

Die Leistungsfähigkeit der Geländesimulation, die unter dem Projektnamen ExoSim entwickelt wurde, ist bereits in Abschnitt 4.3.5 an einigen Beispielen demonstriert worden. Für die folgende abschließende Bewertung werden insbesondere die verschiedenen Aspekte, die bei der Entwicklung berücksichtigt werden sollten, nochmals aufgegriffen.

Größe und Auflösung des Geländemodells

Die Landschaftsmodellierung für das ExoPlanet-Zielgebiet mit der Fläche von $2\text{ km} \times 2\text{ km}$ und einer Rasterauflösung von 1 m, was der Auflösung der MRO-Höhenprofile entspricht, konnte erreicht werden. In dieser Konfiguration wurde eine planetare Umgebung generiert, die als Grundlage für die simulierte Exploration mit mehreren Rovern diente und die als Ergebnis vorgestellt wurde.

Geländeparameter

Als Geländeparameter wurden Hindernisse, Unebenheiten und die Bodenbeschaffenheit modelliert, wobei die Umsetzung dem in Abschnitt 4.3.4 beschriebenen Konzept gefolgt ist. Das hier ebenfalls beschriebene Objekt-Paging-Verfahren (mit dem z. B. die Steinmodelle nur in einem vorgegebenen Radius um den Standort dargestellt werden) konnte in die Geländesimulation integriert werden. So war es möglich, über das gesamte Explorationsgebiet ca. 10.000 große Steine als Hindernisse und 800.000 kleine Steine als Unebenheiten zu verteilen. Die

Anstiege ergaben sich dagegen direkt aus dem Höhenmodell. Für den Versuchslauf mit einem einzelnen Rover wurde zudem illustriert, wie der Streckenverlauf von diesem Parameter beeinflusst wurde (vgl. Abbildung 32f).

Visualisierung

Die verwendete Grafikkbibliothek OGRE 3D kann für den Renderingprozess einen Qualitätsparameter berücksichtigen, um die Auflösung des Geländemodells für eine schnelle Visualisierung herabzusetzen. Wird zudem für das zu Grunde liegende Höhenmodell ein Paging-Verfahren eingesetzt, so kann sich ein Benutzer durch eine beliebig große Landschaft bewegen. Zudem wurde der Anwendungsfall der interaktiven Visualisierung bei der Entwicklung berücksichtigt. Die Geländesimulation kann zu diesem Zweck als eigenständige Applikation oder als Softwarebibliothek genutzt werden, wobei in beiden Fällen die gleiche Codebasis verwendet wird. Der Unterschied besteht nur bei der Parametrierung (z. B. bezüglich der Darstellungsqualität).

Simulation und Schnittstellen

Wird die Geländesimulation in Form der Softwarebibliothek verwendet, so kann der Renderingprozess über verschiedene Funktionsaufrufe beeinflusst werden. Der Schwerpunkt liegt hierbei nicht auf der Echtzeit-Bilddarstellung, sondern auf einer präzisen Positionierung, Modellberechnung bzw. auf der Datenausgabe. Für die Bibliotheksfunktionen wurde mit Hilfe des Player-Servers und des zu Grunde liegenden Protokolls eine definierte Zugriffsmöglichkeit geschaffen, um Befehle und Daten mit der Exo-Sim Anwendung auszutauschen. Hierüber stehen auch simulierte Sensordaten zur Verfügung, die in Form von Koordinaten aus den Modellgeometrien abgeleitet wurden und bereits hinsichtlich der beschriebenen Geländeparameter klassifiziert sind.

Die vorgestellten Simulationsergebnisse konnten basierend auf einem planetaren Geländemodell die Funktionsweise des entwickelten Navigationsverfahrens veranschaulichen. Die Implementierung dieses Verfahrens erfolgte unter dem Projektnamen ExoSim und sollte es den Explorationsrovern ermöglichen, autonom ein unbekanntes Gebiet flächendeckend zu erfassen. Für die Entwicklung der dafür notwendigen Algorithmen galten ebenfalls verschiedene Anforderungen, die im Folgenden bewertet werden.

Sensordaten

Die Sensordaten für den Navigationsalgorithmus werden von der Geländesimulation bereitgestellt. Die Umgebungskarten beinhalten bereits die Informationen über die Position von Hindernissen und die verschiedenen Geländeeigenschaften. Alle weiteren Verarbeitungsschritte beruhen ausschließlich auf diesen Daten, weitere Eingaben sind nicht notwendig.

Pfadplanung

Die Pfadplanung beruht auf dem Dijkstra-Algorithmus. Da mehrere alternative Routen bestimmt werden, die zudem zufällig aus allen möglichen Zielpunkten ausgewählt sind, ist dieses Verfahren besser geeignet als z. B. der A*-Algorithmus. Wie in den Anforderungen

formuliert, werden nicht befahrbare Bereiche für die Streckenplanung ausgeschlossen und schwierige Untergrundbeschaffenheiten wenn möglich vermieden.

Autonomie

Das vorgestellte Explorationsverfahren funktioniert vollständig autonom und bedarf keiner externen Steuerung. Das Grundprinzip beruht auf einer zufälligen Richtungsauswahl, wobei durch die Bestimmung mehrerer Routen und der Sortierung nach der Schwierigkeit der Pfade, ein einfacheres und damit schnelleres Vorankommen gefördert wird.

Kooperation

Wie der Versuchslauf mit mehreren Explorationsrovern zeigt, eignet sich das vorgestellte Verfahren auch für die dezentrale Koordinierung von verteilten Systemen. Da lediglich Umgebungsdaten ausgetauscht werden, bleibt die Autonomie aller beteiligten Systeme erhalten. Zudem ist die Repräsentation der Umgebungen mit einer sehr kompakten Darstellung möglich. Die Explorationsbäume, die untereinander abgeglichen werden, bestehen für jede Position jeweils aus zwei Koordinatenwerten, einem mittleren Umgebungsradius und einer Referenz auf die Vorgängerposition. Die sich daraus ergebende Datenmenge ist über begrenzte Bandbreiten schnell übertragbar und kann auch in mehrfacher Form, wie bei dem verteilten System notwendig, von den Onboard-Systemen der Rover verarbeitet werden. Zudem ist die Explorationsstrategie skalierbar, falls zusätzliche Rover hinzukommen.

Wie diese Zusammenfassung der Anforderungen zusammen mit den genannten Aspekten zur Umsetzung und dem Hinweis auf die vorgestellten Ergebnisse zeigt, konnte das Ziel für die Demonstration einer simulierten Rovernavigation erreicht werden. Um einen praktischen Bezug herzustellen, wurden die Konzepte für die Entwicklung einer Geländesimulation und für die autonome Explorationsstrategie an den Vorgaben einer Beispielmission ausgerichtet. Insbesondere die Ergebnisse der Closed-Loop-Simulation konnten zeigen, dass der entwickelte Lösungsvorschlag einen geeigneten Ansatz zur Erreichung der ExoPlanet-Missionsziele darstellt. Für einen praktischen Test dieses Verfahrens mit einem Roversystem im freien Gelände müssen für den Navigationsalgorithmus noch einige Erweiterungen und Verbesserungen vorgenommen werden. Die wesentlichen Probleme und Möglichkeiten werden im Folgenden diskutiert.

1. Für die Simulation wurde von idealen Messdaten ausgegangen.

Der verwendete Navigationsalgorithmus basiert auf einer von den Roversensoren erfassten Umgebungskarte. Für die Simulation werden die Koordinaten direkt aus dem Geländemodell abgeleitet. Sie stellen somit einen idealen Datensatz dar. Für einen realen Einsatz des Verfahrens muss der Algorithmus noch um eine Stereobildverarbeitung zur Koordinatenbestimmung sowie Methoden zur Objektsegmentierung und Klassifikation zur Erkennung von Hindernissen ergänzt werden. Für die Einschätzung der Untergrundbeschaffenheit werden in der Praxis neben der Kamerabildanalyse auch Traktionsdaten vom Antriebssystem verwendet – auch diese sollten vom Navigationsalgorithmus berücksichtigt werden. Wie bereits beim Konzeptentwurf diskutiert, stellt die Simulationsumgebung für die Entwicklung dieser Erweiterungen

ein Hilfsmittel dar. So können z. B. nur Stereobildpaare (sogar unter Berücksichtigung von Abbildungsfehlern) generiert werden. Die Modelldaten erlauben dann die Überprüfung der Analyseergebnisse für die ermittelten Koordinaten oder die Klassifizierung.

Ein zweiter wesentlicher Aspekt, der sich für reale Messdaten ergibt, ist die Notwendigkeit der Lokalisierung. Für die Erstellung der Umgebungskarten muss die genaue Position und Ausrichtung des Rovers bekannt sein. Diese Informationen können aber meist nur aus einer korrekten Kartengrundlage abgeleitet werden. Im Kapitel über die Rovernavigation wurden verschiedene Ansätze dargestellt (z. B. visuelle Odometrie, Bundle-Adjustment oder Lokalisierung anhand von Landmarken und der Wide-Baseline Stereoanalyse). Für einen realen Versuch müssen derartige Methoden ebenfalls berücksichtigt werden. An dieser Stelle sei nochmals auf die Arbeit von [Lin10] verwiesen, in der ein Lokalisierungsverfahren beschrieben wird, das ebenfalls anhand der hier vorgestellten Simulationsumgebung demonstriert wurde.

2. Die Pfadplanung basiert auf einem Rastermodell und erlaubt nur diskrete Richtungen.

Für den verwendeten Dijkstra-Algorithmus wurde aus der Umgebungskarte ein Graph abgeleitet, der horizontale und diagonale Bewegungsrichtungen erlaubt. Für einen kontinuierlichen Streckenverlauf muss daher noch eine Interpolation der Pfadpunkte vorgenommen werden. Zudem wurde die Manövrierfähigkeit des Rovermodells nicht berücksichtigt. Für Systeme mit einem Einachsantrieb ist z. B. kein Wenden auf der Stelle möglich. Derartige Randbedingungen müssen ebenfalls in das Planungsverfahren einfließen.

3. Bei den Explorationsrouten werden zu viele Verbindungsstrecken zurückgelegt.

Die Simulationsversuche haben gezeigt, dass mit der Zunahme der erfassten Bereiche immer häufiger Verbindungsstrecken zurückgelegt werden, also Navigationsabschnitte, auf denen keine neuen Gebiete erfasst werden. In der Bewertung der Ergebnisse wurden bereits die Ursachen genannt: für die Pfadplanung werden nur lokale Informationen berücksichtigt und basierend auf einer Baumstruktur bewegt sich der Rover jeweils zum vorherigen Standort zurück, falls in einer eingeschlagenen Richtung kein Vorankommen mehr gewährleistet ist. Dieser Ansatz hat den Vorteil, dass ein Explorationsfahrzeug autonom weiteroperieren kann, auch wenn eine Sackgasse auftritt. Außerdem besitzt das Verfahren die Eigenschaft, dass die Fahrzeuge am Ende des Explorationsprozesses zum Ausgangspunkt zurückkehren.

Für ein Verfahren, welches auf minimale Fahrdistanzen ausgelegt ist und in dieser Form für die planetare Exploration benötigt wird, sollte der Explorationsbaum durch einen Explorationsgraphen ersetzt werden. In diesem könnten auch Zwischenverbindungen bezüglich naheliegender Positionen eingefügt werden. So kann sich z. B. für die Rückkehr zum Ausgangsort eine direkte Verbindung ergeben. In einer Sackgassensituation sollte dann die gesamte Graphenstruktur mit den jeweils eingetragenen Abkürzungsrouten betrachtet werden, um schnell zu einer neuen Explorationsfront zu gelangen. Pfadoptimierungsverfahren können hier ebenfalls helfen, ein Ziel zu finden, bei dem ein ausgeglichenes Verhältnis zwischen zurückzulegender Strecke und Informationszugewinn besteht. In Abschnitt 3.2.4 wurde kurz auf das Prinzip dieser Frontier-Based-Methoden eingegangen.

4. Die autonome Routenplanung ist unvorhersagbar.

Bei der Betrachtung der Simulationsergebnisse wird ebenfalls deutlich, dass die Richtungswahl bei der Explorationsstrategie sehr willkürlich erscheint. Der Grund für diesen Streckenverlauf ist das Prinzip der zufälligen Richtungsauswahl. Damit stellt sich zur Diskussion, ob nicht einfache verhaltensbasierte (Behavior-based) Steuerungen besser geeignet sind.

Dem zu Grunde liegenden SRT-Verfahren können jedoch weitere Parameter zur Gewichtung der Zufallsselektion übergeben werden. Für die vorgestellte Implementierung wurde dieser Aspekt bereits berücksichtigt, indem Routen mit einer geringen Pfadkostenbewertung, d. h. einem niedrigen Schwierigkeitsgrad, vorgezogen werden. In dieser Form können noch weitere Randbedingungen einfließen, wie z. B. eine Exploration, die sich auf ein bestimmtes Gebiet konzentriert oder Bergregionen bzw. steinige Bereiche vorzieht. Auch eine Konfiguration, die eine bestimmte Richtung beizubehalten versucht, ist denkbar, womit sich einer systematischen rasterbasierten Exploration angenähert werden kann. Die zufällige Richtungswahl kann somit durch ein übergeordnetes Ziel eingeschränkt werden. Zusammen mit globalen Informationen, die durch den Explorationsgraphen zur Verfügung stehen, sind hinsichtlich dieser Ziele Optimierungen möglich, was ein Vorteil gegenüber rein verhaltensbasierten Systemen darstellt.

5. Planetare Missionen sind nur mit Vorkenntnissen über das Zielgebiet durchführbar.

Insbesondere für die Konstruktion des Landesystems ist es notwendig, Informationen über das Missionsgebiet und über grundlegende Geländeparameter zu besitzen. Die Annahme einer Exploration ohne Vorkenntnisse entspricht daher nicht der Realität. Das Ziel einer flächendeckenden Erfassung, wie es die ExoPlanet-Beispielmission beschreibt, ist dagegen für die geologische Forschung prinzipiell denkbar. In die dafür entwickelte Explorationsstrategie können vorhandene Kartendaten durchaus mit einfließen. Die Informationen können wiederum zur Gewichtung der Richtungsauswahl verwendet werden. Außerdem können bei der Erweiterung des Explorationsgraphen mögliche Sackgassen vorausgesehen oder gezielt Abkürzungsrouten erschlossen werden. Mit einer groben Geländevorkenntnis ist es zudem möglich, eine topologische Repräsentation des Gesamtgebiets abzuleiten, die wie in der Arbeit von [Kle07] beschrieben zur Koordinierung mehrerer Fahrzeuge eingesetzt werden kann.

6. Beim Einsatz mehrerer Explorationsrover kommt es zu gegenseitigen Beeinträchtigungen.

Bei der Beschreibung der Ergebnisse wurde bereits darauf hingewiesen, dass es beim gleichzeitigen Einsatz mehrerer Rover zu einer gegenseitigen Beeinträchtigung kommen kann. Ist das Explorationsgebiet eines Rovers, wie in Abbildung 33 zu sehen, von bereits erfassten Bereichen der anderen Fahrzeuge umschlossen, so findet sich nach kurzer Zeit kein unbekanntes Gebiet mehr. Die Anforderung nach Kooperation ist nur als eine Form der Kommunikation umgesetzt. Mit einer gemeinsamen Datenstruktur wäre es möglich, dass sich die Fahrzeuge neue Explorationsgebiete untereinander zuweisen und dadurch auch wirklich im Sinne der Zielstellung kooperieren.

6.2 Schlussfolgerungen und Fazit

Ausgehend von den dargestellten Simulationsergebnissen, zusammen mit der Bewertung der praktischen Umsetzung bezüglich der Anforderungen, wurde die Funktionalität der Geländesimulation und des Navigationsverfahrens erwiesen. Mit der Softwareimplementierung und der Demonstration der erarbeiteten Konzepte konnte die Aufgabenstellung hinsichtlich beider Anwendungen erfüllt werden.

Die Diskussion um die Erweiterungsmöglichkeiten und Verbesserungen hat gezeigt, dass es sich bei der Explorationsstrategie, die für die Versuchsläufe verwendet wurde, zunächst um einen vereinfachten Ansatz handelt. Es ist dabei von verschiedenen Annahmen ausgegangen worden, wie den idealen und bereits klassifizierten Messdaten. Zudem wurde mit der verwendeten Baumstruktur eine einfache Möglichkeit zur Repräsentation der erfassten Umgebung gewählt, die aber hinsichtlich der zurückgelegten Strecken Nachteile bietet. Diese Vereinfachungen wurden mit Hinblick auf die Aufgabenstellung dieser Diplomarbeit getroffen, die im Ergebnis eine funktionsfähige Closed-Loop-Simulation mit allen zugehörigen Komponenten präsentieren sollte.

Die Reduzierung des Navigationskonzepts auf eine funktionierende, aber noch nicht optimierte Implementierung ist zudem für eine wissenschaftliche Evaluierung des Verfahrens sinnvoll. So kann der Algorithmus nacheinander um verschiedene Komponenten erweitert und basierend auf dem Vergleich der Versuchsergebnisse beurteilt werden. Insbesondere bei einer Vielzahl von Optimierungsparametern, wie sie bei Pfadplanungsverfahren auftreten, ist eine getrennte Modifikation und Bewertung der Auswirkungen möglich. Auch wenn Alternativen diskutiert werden sollen, können bereits mit einfachen Simulationsergebnissen Vor- und Nachteile illustriert werden.

Ein Aspekt, der bei der Diskussion über die vorgeschlagenen Algorithmen für die autonome Navigation bisher nicht berücksichtigt wurde, ist die Ausführung auf der Hardware-Plattform der Explorationsfahrzeuge. Als grober Richtwert kann für einen planetaren Rover eine Prozessorarchitektur angenommen werden, die um den Faktor 40 langsamer arbeitet, als aktuelle Desktop-Computer. So war der Mikro-Rover Sojourner mit einer 0.1 MHz CPU und 512 kB Hauptspeicher ausgestattet. Die Mars Exploration Rover verfügen über 20 MHz bei 128 MB RAM und der MSL-Rover wird voraussichtlich mit einem 200 MHz-Prozessor und 256 MB Speicher bestückt [Baj08]. Zudem werden viele Algorithmen direkt als logische Schaltung oder auf einer FPGA (Field Programmable Gate Array) implementiert. So stellen einfache Standardoperatoren, wie Filter oder Bildtransformationen, kein Problem dar. Derartige Funktionen werden bereits erfolgreich im Rahmen der MER-Mission eingesetzt. In [Gol02] sind die Ausführungszeiten der Navigationsroutinen auf der Roverhardware aufgelistet und denen eines herkömmlichen Computers gegenübergestellt, wodurch sich auch der Richtwert für den Geschwindigkeitsvergleich bestätigt.

Das entwickelte Navigationsverfahren, welches aus dem SRT-Ansatz abgeleitet wurde, verwendet ebenfalls nur einfache Bildoperationen. Die Pfadplanung, basierend auf dem Dijkstra-Algorithmus, kann zudem parallelisiert und wie erwähnt in einen FPGA-Chip integriert werden [Tom01]. Auch die Speicherressourcen genügen auf Grund der einfachen Modellierung der Baumstruktur für das Explorationsverfahren. Bei einer optimierten Programmierung stellt die beschränkte Ausführungsplattform also kein Hindernis für den Einsatz des Verfahrens dar, wobei die im Rahmen dieser Arbeit entstandene Implementierung nicht als Vergleich herangezogen werden kann, da sie lediglich der Evaluierung des Konzepts dient.

Über die Geländesimulation kann zusammenfassend gesagt werden, dass sie die zum Test des Navigationsverfahrens notwendigen Eingabeparameter liefern konnte. Erweiterungen für eine möglichst realistische Modellierung von planetaren Landschaften wurden im Abschnitt 4.3.4 im Rahmen des Umsetzungskonzepts vorgeschlagen. Für eine grundlegende Validierung der Simulationsergebnisse sollte eine reale Versuchsumgebung zum Vergleich herangezogen werden. Ein Rover-Testgelände, wie es in der Arbeit von [Dup08] verwendet wurde, kann als Modell erfasst werden, um die Simulation hinsichtlich vergleichbarer Ergebnisse zu überprüfen.

Für eine Erweiterung der Anwendungsmöglichkeit einer planetaren Landschaftssimulation sollten auch physikalische Eigenschaften der Umgebungen berücksichtigt werden. So kann die Roverdynamik basierend auf Schwerkraft und Reibungskräften bis hin zum Traktionsverhalten des Systems modelliert werden. Mit einer Beschreibung der kinematischen Beziehungen z. B. für das Fahrwerk wäre sogar eine Closed-Loop-Simulation des kompletten Rovers möglich. Als Basis für die Modellierung kann eine sogenannte Physik-Engine (wie [39] oder [8]) in die Anwendung integriert werden. Eine weitere notwendige Erweiterung stellen Umgebungsinformationen wie der Sonnenstand dar. Für Roversysteme, die mit Solarenergie angetrieben werden, ergibt daraus für die Pfadplanung eine zusätzliche technische Randbedingung. Mit einer geplanten Route muss immer auch die Energieversorgung sichergestellt werden, es dürfen also keine längeren Schattenabschnitte enthalten sein. Dieser Aspekt muss daher ebenfalls von den Navigationsalgorithmen berücksichtigt werden.

Die Geländesimulation wurde mit dem Ziel entwickelt, ein Navigationsverfahren zu demonstrieren. Dafür wurden auf dem Höhenmodell zufällig Steinmodelle verteilt und dargestellt, der Benutzer kann auf die Verteilung keinen Einfluss nehmen. Für eine allgemeine Simulationsanwendung sollte es möglich sein, Höhenmodelle, Texturen, die Verteilung von Objekten sowie Metainformationen, wie Spektraldaten, oder physikalische Eigenschaften, wie Traktionsparameter, vorzugeben. Für diesen Zweck eignen sich Datenbanken, die auf räumlich referenzierte Datensätze spezialisiert sind und die daher auch als Grundlage für Geoinformationssysteme verwendet werden. Ein Beispiel für eine derartige GIS-Datenbank ist SpatiaLite [43]. Die ebenfalls als Open Source Projekt entwickelte Datenbank kann als Softwarebibliothek in die eigene Anwendung integriert werden. Bei der Entwicklung der ExoSim-Applikation wurden dahingehend einige Tests durchgeführt, die gezeigt haben, dass die SpatiaLite-Komponenten einfach einzubinden sind und damit die Möglichkeit für die Auslagerung der Modelldaten besteht.

Abschließend soll noch darauf hingewiesen werden, dass weitere Ergebnisse dieser Diplomarbeit auf einer Projektseite im Internet zur Verfügung stehen (siehe [20]). Darunter auch einige Demonstrationsvideos und Animationen, die nochmals einen Eindruck von der Geländesimulation vermitteln. Zudem wird hier auch der Quellcode der Anwendungen ExoSim und ExoPath bereitgestellt.

6.3 Ausblick

Mit dem implementierten Explorationsverfahren war es möglich, die Leistungsfähigkeit der SRT-Methode zu demonstrieren. Zudem konnten anhand der Simulationsergebnisse Erweiterungsmöglichkeiten vorgeschlagen und diskutiert werden. In dieser Arbeit wurden die wesentlichen Grundlagen beschrieben, um ein solches erweitertes Navigationskonzept umzusetzen. An dieser Stelle werden noch einmal wesentliche Aspekte zusammengefasst, die bei der Weiterentwicklung dieses Verfahrens im nächsten Iterationsschritt berücksichtigt werden sollten.

- Der Navigationsalgorithmus sollte durch eine Komponente für die Stereo-Bildverarbeitung sowie zur Steinerkennung und Untergrundklassifikation ergänzt werden. Als Basis für die notwendige Bildprozessierung kann auf die OpenCV-Bibliothek [41] zurückgegriffen werden. Zudem sollte ein SLAM-Verfahren integriert werden. Für verschiedene Standardverfahren stehen auch hier bereits quellenoffene Implementierungen zur Verfügung, wie z. B. als Bestandteil des Mobile Robot Programming Toolkit [26].
- Der Dijkstra-Algorithmus, als Grundlage für die Pfadplanung, ist gut geeignet, um basierend auf vollständigen Umgebungsdaten eine optimale Route zu finden. Aufgrund von Messungenauigkeiten oder Bereichsverdeckungen oder um weniger Rundum-Aufnahmen erstellen zu müssen, sollte während der Navigation eine zusätzliche dynamische Pfadoptimierungsmethode wie das D*-Verfahren eingesetzt werden. Damit wäre es auch möglich, die Streckenvorgabe nochmals zu interpolieren und hinsichtlich der Manövrierfähigkeit des Roversystems zu optimieren.
- Der Sensor-based Random Tree (SRT) sollte, wie in der Arbeit von [Fra07] gezeigt, zu einem Sensor-based Random Graph (SRG) erweitert werden. Damit können Verbindungsstrecken hinzugefügt werden, um langes Zurücknavigieren zu vermeiden. Zudem ist diese Repräsentation zur Koordinierung von mehreren Roversystemen besser geeignet.
- Die zufällige Richtungsauswahl sollte stärker bezüglich eines übergeordneten Ziels gewichtet werden. Hierbei sollten zusätzliche Vorabinformationen über das Gelände einfließen. Zudem sollte nicht nur die Rückkehr zum vorherigen Standort möglich sein, sondern immer jeweils der gesamte Explorationsgraph betrachtet werden, um ein optimales Verhältnis zwischen Streckenlänge und Informationszugewinn zu gewährleisten.

Auch für die Geländesimulation wurden verschiedene Erweiterungsmöglichkeiten aufgezeigt. Teilweise lassen sich diese Zusatzfunktionen unter Verwendung von frei verfügbaren Softwaremodulen einfach umsetzen, erschließen dadurch aber für die Simulationsumgebung neue

Anwendungsmöglichkeiten. Sinnvolle Modifikationen sind diesbezüglich die Integration einer Physik-Engine, die Simulation von Sonnenstand und Licht-/Schattenverhältnissen sowie die Auslagerung der Modelldaten in eine GIS-Datenbank. Zudem sollte eine variable Einstellung der maximalen Bodenauflösung möglich sein – für eine Kinematiksimulation ist die verwendete Rasterbreite von 1 m zu groß. Es müssen daher geeignete Interpolations- und Rauschgenerierungsverfahren, z. B. basierend auf der Perlin Noise Funktion, integriert werden.

Insgesamt ist es mit einer derart erweiterten Software zur Generierung von planetaren Landschaften möglich, Simulationen für die Rovernavigation, für Landemanöver oder sogar für Orbiterinstrumente durchzuführen. Es können Sensordaten simuliert werden, wie Kamerabilder oder Lidar-Distanzen. Die Evaluierung von Algorithmen ist ebenso eine Anwendung, wie die 3D-Visualisierung der Modelle.

In dieser Arbeit wurde neben der Konzept- und Softwareentwicklung auch ein umfassender Überblick über die Literatur in den verschiedenen Schwerpunktbereichen vermittelt. Insbesondere beim Thema Rovernavigation zeigt sich, dass viele Arbeiten im Rahmen der Mars Exploration Rover Mission veröffentlicht wurden oder sich auf diese beziehen. Damit ist die MER-Mission ein gutes Beispiel dafür, wie Raumfahrt als Impulsgeber für die Wissenschaft und Technik fungieren kann, und dass erfolgreiche Projekte, zusammen mit verfügbaren Daten, immer auch Inspiration für neue Möglichkeiten sind.

Die praktische Umsetzung des Navigationsalgorithmus als Simulationsanwendung hat darüber hinaus den Nutzer der Open Source Entwicklung verdeutlicht. So war es in kurzer Zeit möglich, eine Software zur Verarbeitung und Darstellung von komplexen 3D-Modellen, eine Netzwerk-Kommunikationsschnittstelle und eine Benutzeroberfläche zu entwickeln. Gleichzeitig konnten durch den verfügbaren Quellcode Änderungen für die eigenen Anforderungen, wie die Ausgabe der Objektkoordinaten, vorgenommen werden. Wie bereits in der Einleitung formuliert, sollten die Open Source Grundsätze stärker im Bereich der Raumfahrt berücksichtigt werden. Teilweise ist auch bereits ein Weg in diese Richtung zu erkennen. So stellt die NASA einige Projekte unter einer eigenen freien Lizenz zur Verfügung [28] und hat 2011 im Rahmen der Open Government Initiative auch eine erste Konferenz zu diesem Thema veranstaltet [29].

Die für die nächsten Jahre geplanten Rovermissionen Mars Science Laboratory und ExoMars werden auch auf dem Gebiet der autonomen Navigation zu neuen Erkenntnissen führen. Möglicherweise werden im Rahmen der ESA Cosmic Vision Agenda oder der NASA Programme zur Erforschung des Sonnensystems noch weiter entlegene Explorationsziele, wie der Saturnmond Titan, ausgewählt, oder es werden verteilte Systeme für Mikro-Rover, Ballone oder andere Arten von Flugrobotern eingesetzt. Mit derartigen Szenarien geht auch die Notwendigkeit an zunehmender Autonomie für die Explorationssonden einher. So ist diese Diplomarbeit auch als ein Beitrag für eine weitere Entwicklung auf diesem Weg zu sehen.

Literatur

- [Ale06] Alexander, D. A. u. a., „Processing of Mars Exploration Rover imagery for science and operations planning“, *Journal of Geophysical Research* 111(E2), 2006, DOI: 10.1029/2005JE002462 (→ Seite 16)
- [Baj08] Bajracharya, M. u. a., „Autonomy for Mars Rovers: Past, Present, and Future“, *Computer* 41(12), 2008, DOI: 10.1109/MC.2008.479 (→ Seite 72)
- [Bel03] Bell, J. F. I. u. a., „Mars Exploration Rover Athena Panoramic Camera (Pancam) investigation“, *Journal of Geophysical Research* 108(E12), 2003, DOI: 10.1029/2003JE002070 (→ Seite 16)
- [Boe00] Boer, W. H. de, „Fast Terrain Rendering Using Geometrical MipMapping“, 2000, URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.2737> (→ Seite 33)
- [Bur08] Burger, W., „Gradientenbasierte Rauschfunktionen und Perlin Noise“, Technischer Bericht HGBTR08-02, School of Informatics, Communications und Media, Upper Austria University of Applied Sciences, 2008, URL: <http://staff.fh-hagenberg.at/burger/> (→ Seite 45)
- [Car07] Carsten, J. u. a., „Global Path Planning on Board the Mars Exploration Rovers“, *IEEE Aerospace Conference*, Big Sky, März 2007, DOI: 10.1109/AERO.2007.352683 (→ Seite 29)
- [Cha96] Chatila, R. u. a., „Autonomous Mobile Robot Navigation for Planet Exploration: The EDEN Project“, *IEEE International Conference on Robotics and Automation (ICRA), Workshop on Planetary Rover Technology and Systems*, Minneapolis, Apr. 1996 (→ Seite 12)
- [Che08] Chen, W., „Wide Baseline Mapping in Mars Exploration“, Master Thesis, Ohio State University, Department of Civil, Environmental Engineering und Geodetic Science, 2008 (→ Seite 20)
- [Cho05] Choset, H. M., *Principles of robot motion: theory, algorithms, and implementation*, MIT Press, 2005 (→ Seiten 24, 25, 29)
- [Cor90] Cormen, T. H. u. a., *Introduction to Algorithms*, The MIT Press, 1990 (→ Seite 26)
- [Dac06] Dachsbacher, C., „Interactive Terrain Rendering: Towards Realism with Procedural Models and Graphics Hardware“, Dissertation, Universität Erlangen–Nürnberg, Technische Fakultät, 2006 (→ Seite 45)

- [Dam07] Damer, B. u. a., „Simulation-Based Lunar Telerobotics Design, Acquisition and Training Platform for Virtual Exploration“ Final Technical Report, NASA, DigitalSpace, 2007, URL: <http://www.digitalspace.com/reports/sbir04-phase2-finalreport/index.html> (→ Seite 39)
- [Dee92] Deen, R. G., „The VICAR file format“, 1992, URL: http://www-mipl.jpl.nasa.gov/external/VICAR_file_fmt.pdf (→ Seite 43)
- [Di08] Di, K. u. a., „Photogrammetric processing of rover imagery of the 2003 Mars Exploration Rover mission“, *ISPRS Journal of Photogrammetry and Remote Sensing* 63(2), 2008, DOI: 10.1016/j.isprsjprs.2007.07.007 (→ Seite 20)
- [DM09] Dubois-Matra, O. u. a., „Testing and Validation of Planetary Vision Based Navigation Systems with Pangu“, *21st International Symposium on Space Flight Dynamics (ISSFD)*, Toulouse, Sep. 2009 (→ Seiten 36, 37)
- [Duc97] Duchaineau, M. u. a., „ROAMing Terrain: Real-time Optimally Adapting Meshes“, *8th Conference on Visualization*, Phoenix, Okt. 1997 (→ Seite 32)
- [Dup08] Dupuis, E. u. a., „Autonomous Over-The-Horizon Rover Navigation“, *10th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA)*, Noordwijk, Nov. 2008 (→ Seiten 23, 73)
- [Flu02] Fluckiger, L. u. a., „A new simulation framework for autonomy in robotic missions“, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Okt. 2002, DOI: 10.1109/IRDS.2002.1041733 (→ Seite 36)
- [Fou82] Fournier, A. u. a., „Computer Rendering of Stochastic Models“, *Communications of the ACM* 25(6), 1982, DOI: 10.1145/358523.358553 (→ Seite 31)
- [Fox06] Fox, J. M. u. a., „Advances in distributed operations and mission activity planning for Mars surface exploration“, *AIAA 9th International Conference on Space Operations (SpaceOps)*, Rom, Juni 2006, HDL: 2014/39854 (→ Seite 35)
- [Fra07] Franchi, A. u. a., „A Decentralized Strategy for Cooperative Robot Exploration“, *International Conference on Robot Communication and Coordination (RoboComm)*, Athen, Okt. 2007 (→ Seiten 54, 74)
- [Fre05] Freda, L. u. a., „Frontier-Based Probabilistic Strategies for Sensor-Based Exploration“, *IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Apr. 2005, DOI: 10.1109/ROBOT.2005.1570713 (→ Seite 52)
- [Fre07] Freda, L. u. a., *Randomized Strategies for Sensor-Based Robot Exploration*, Projektpräsentation, 2007, URL: <http://www.dis.uniroma1.it/~labrob/pub/slides/RT-Slides.pdf> (→ Seiten 53, 54)
- [Glo04] Glotzbach, T., „Adaptive autonomy: a suggestion for the definition of the notation ‘autonomy’ in mobile robotics“, *IEEE International Conference on Control Applications*, Taipei, Sep. 2004, DOI: 10.1109/CCA.2004.1387487 (→ Seiten 12, 13)

- [Glo09] Glotzbach, T. u. a., „CViewVR: A High-performance Visualization Tool for Team-oriented Missions of Unmanned Marine Vehicles“, *8th International Conference on Computer Applications and Information Technology in the Maritime Industries (COMPIT)*, Budapest, Mai 2009 (→ Seite 39)
- [Gol01] Golombek, M. u. a., „Crater and rock hazard modeling for Mars landing“, *Space 2001 Conference and Exposition*, Albuquerque, Aug. 2001, HDL: 2014/12872 (→ Seite 45)
- [Gol02] Goldberg, S. B. u. a., „Stereo Vision and Rover Navigation Software for Planetary Exploration“, *IEEE Aerospace Conference Proceedings*, Big Sky, März 2002, DOI: 10.1109/AERO.2002.1035370 (→ Seite 72)
- [Gol08] Golombek, M. P. u. a., „Size-frequency distributions of rocks on the northern plains of Mars with special reference to Phoenix landing surfaces“, *Journal of Geophysical Research* 113(E3), 2008, DOI: 10.1029/2007JE003065 (→ Seite 45)
- [Hal07] Haldemann, A. F. C. u. a., „Simulation to Evaluate Autonomous Behaviors for Mobile Planetary Surface Science Missions“, *IEEE Aerospace Conference*, Big Sky, März 2007, DOI: 10.1109/AERO.2007.352698 (→ Seite 36)
- [Hay05] Hayati, S. A., „The Mars Technology Program“, *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, München, Sep. 2005, HDL: 2014/37742 (→ Seite 21)
- [Hon09] Honary, E. u. a., „Robotic experiments with cooperative aerobots and underwater swarms“, *Robotica* 27(1), 2009, DOI: 10.1017/S0263574708004505 (→ Seite 13)
- [Iag04] Iagnemma, K. u. a., *Mobile Robots in Rough Terrain - Estimation, Motion Planning, and Control with Application to Planetary Rovers*, Springer, 2004, DOI: 10.1007/b94718 (→ Seite 25)
- [Ish07] Ishigami, G. u. a., „Path Planning for Planetary Exploration Rovers and Its Evaluation based on Wheel Slip Dynamics“, *IEEE International Conference on Robotics and Automation (ICRA)*, Rom, Apr. 2007, DOI: 10.1109/ROBOT.2007.363672 (→ Seite 23)
- [Jai04] Jain, A. u. a., „Recent developments in the ROAMS planetary rover simulation environment“, *IEEE Aerospace Conference*, Big Sky, März 2004, DOI: 10.1109/AERO.2004.1367686 (→ Seite 36)
- [Jai06] Jain, A. u. a., „SimScape Terrain Modeling Toolkit“, *IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, Pasadena, Aug. 2006, DOI: 10.1109/SMC-IT.2006.67 (→ Seite 36)
- [Jia97] Jiang, X. u. a., *Dreidimensionales Computersehen: Gewinnung und Analyse von Tiefenbildern*, Springer, 1997 (→ Seite 16)

- [Kap09] Kapellos, K. u. a., „3DROV: A Planetary Rover Design Tool based on SIMSAT v4“, *European Ground System Architecture Workshop (ESAW)*, Darmstadt, Mai 2009 (→ Seite 37)
- [Kar10] Karahayit, O., „Mars Exploration Rover Mission - Rover Localization and Topographic Mapping“, Master Thesis, Ohio State University, 2010 (→ Seiten 18, 20)
- [Kim09] Kim, W. S. u. a., „Targeted Driving Using Visual Tracking on Mars: From Research to Flight“, *Journal of Field Robotics* 26(3), 2009, DOI: 10.1002/rob.20283 (→ Seite 23)
- [Kle07] Kleiner, A. D., „Mapping and Exploration for Search and Rescue with Humans and Mobile Robots“, Dissertation, Universität Freiburg, Institut für Informatik, 2007 (→ Seiten 25, 71)
- [Kol10] Kolling, A. u. a., „Solving Pursuit-Evasion Problems on Height Maps“, *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Mai 2010 (→ Seite 24)
- [Kui91] Kuipers, B. u. a., „A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations“, *Robotics and Autonomous Systems* 8(1-2), 1991, DOI: 10.1016/0921-8890(91)90014-C (→ Seite 24)
- [Lam08] Lamon, P., *3D Position Tracking and Control for all-Terrain Robots*, Springer, 2008, DOI: 10.1007/978-3-540-78287-2 (→ Seiten 10, 18, 19)
- [Lat08] Lathan, M., „GPU-basierte Simulation dynamischer Terrains“, Diplomarbeit, Technische Universität Dresden, Fachgebiet Computergrafik, 2008 (→ Seite 45)
- [Lat90] Latombe, J., *Robot motion planning*, Springer, 1990 (→ Seiten 25, 29)
- [LaV06] LaValle, S. M., *Planning Algorithms*, Cambridge University Press, 2006 (→ Seiten 23, 24, 30, 52)
- [Li06] Li, R. u. a., „Spirit rover localization and topographic mapping at the landing site of Gusev crater, Mars“, *Journal of Geophysical Research* 111(E2), 2006, DOI: 10.1029/2005JE002483 (→ Seite 21)
- [Lin10] Lingenauber, M., „Implementierung und Analyse eines automatischen Matchingverfahrens für die Selbstlokalisierung planetarer Rover“, Diplomarbeit, Technische Universität Berlin, Fachbereich Raumfahrttechnik, 2010 (→ Seiten 19, 52, 70)
- [Mai06] Maimone, M. u. a., „Surface Navigation and Mobility Intelligence on the Mars Exploration Rovers“, *Intelligence for Space Robotics*, hrsg. von Howard, A. u. a., TSI Press, 2006 (→ Seiten 21, 22)
- [Mai07] Maimone, M. u. a., „Two years of Visual Odometry on the Mars Exploration Rovers“, *Journal of Field Robotics* 24(3), 2007, DOI: 10.1002/rob.20184 (→ Seite 20)
- [Mak02] Makarenko, A. A. u. a., „An experiment in integrated exploration“, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Sep. 2002, DOI: 10.1109/IRDS.2002.1041445 (→ Seite 15)

- [Mak03] Maki, J. N. u. a., „Mars Exploration Rover Engineering Cameras“, *Journal of Geophysical Research* 108(E12), 2003, DOI: 10.1029/2003JE002077 (→ Seite 16)
- [Mat97] Matthies, L. u. a., „Visual Localization Methods for Mars Rovers using Lander, Rover, and Descent Imagery“, *4th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Tokyo, Juli 1997, HDL: 2014/22227 (→ Seite 62)
- [McC08] McCrum, M. u. a., „Realistic Image Generation for Testing Vision-Based Autonomous Rendezvous“, *10th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA)*, Noordwijk, Nov. 2008 (→ Seite 37)
- [Mit03] Mitschick, A., „Interaktive Gesteinsmodellierung für computergenerierte Landschaften“, Diplomarbeit, Technische Universität Dresden, Fachgebiet Computergrafik, 2003 (→ Seite 45)
- [Ngu01] Nguyen, L. A. u. a., „Virtual Reality Interfaces for Visualization and Control of Remote Vehicles“, *Autonomous Robots* 11(1), 2001, DOI: 10.1023/A:1011208212722 (→ Seite 35)
- [Nor05] Norris, J. S. u. a., „Mars Exploration Rover Operations with the Science Activity Planner“, *IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Apr. 2005, DOI: 10.1109/ROBOT.2005.1570832 (→ Seite 35)
- [Ori04] Oriolo, G. u. a., „The SRT method: randomized strategies for exploration“, *IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, Apr. 2004, DOI: 10.1109/ROBOT.2004.1302457 (→ Seiten 52, 54)
- [Par04] Parkes, S. u. a., „Planet Surface Simulation with PANGU“, *Eighth International Conference on Space Operations (SpaceOps)*, Montreal, 2004 (→ Seiten 36, 45)
- [Par07] Parteli, E. J. R., „Sand Dunes on Mars and on Earth“, Dissertation, Institut für Computerphysik der Universität Stuttgart, 2007 (→ Seite 47)
- [Ped00] Pedersen, L., „Robotic Rock Classification and Autonomous Exploration“, Dissertation, Robotics Institute, Carnegie Mellon University, 2000 (→ Seite 25)
- [Pen09] Peniak, M. u. a., „Co-evolving controller and sensing abilities in a simulated Mars Rover explorer“, *IEEE Congress on Evolutionary Computation (CEC)*, Trondheim, Mai 2009, DOI: 10.1109/CEC.2009.4983290 (→ Seite 30)
- [Pey09] Peytavie, A. u. a., „Procedural Generation of Rock Piles using Aperiodic Tiling“, *Computer Graphics Forum* 28(7), 2009, DOI: 10.1111/j.1467-8659.2009.01557.x (→ Seite 45)
- [Pon02] Poncela, A. u. a., „Efficient integration of metric and topological maps for directed exploration of unknown environments“, *Robotics and Autonomous Systems* 41(1), 2002, DOI: 10.1016/S0921-8890(02)00272-5 (→ Seite 18)

- [Pou08] Poulakis, P. u. a., „3DROV: A Planetary Rover System Design, Simulation and Verification Tool“, *10th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA)*, Noordwijk, Nov. 2008 (→ Seite 37)
- [Sim00] Simmons, R. u. a., „Coordination for Multi-Robot Exploration and Mapping“, *National Conference on Artificial Intelligence (AAAI)*, Austin, Juli 2000 (→ Seite 24)
- [Sta09] Stachniss, C., *Robotic Mapping and Exploration*, Springer, 2009, DOI: 10.1007/978-3-642-01097-2 (→ Seite 14)
- [Ste94] Stentz, A., „Optimal and Efficient Path Planning for Partially-Known Environments“, *IEEE International Conference on Robotics and Automation (ICRA)*, San Diego, Mai 1994, DOI: 10.1109/ROBOT.1994.351061 (→ Seite 29)
- [Ste95] Steels, L., „When are robots intelligent autonomous agents?“, *Robotics and Autonomous Systems* 15(1-2), 1995, DOI: 10.1016/0921-8890(95)00011-4 (→ Seite 12)
- [Thr96] Thrun, S. u. a., „Integrating Grid-Based and Topological Maps for Mobile Robot Navigation“, *National Conference on Artificial Intelligence (AAAI)*, Portland, Aug. 1996 (→ Seite 18)
- [Tom01] Tommiska, M. u. a., *Dijkstra's shortest path routing algorithm in reconfigurable hardware*, Springer, 2001, DOI: 10.1007/3-540-44687-7_73 (→ Seite 73)
- [Tom05] Tompkins, P., „Mission-Directed Path Planning for Planetary Rover Exploration“, Dissertation, Carnegie Mellon University, The Robotics Institute, 2005 (→ Seite 23)
- [Ulr02] Ulrich, T., „Rendering Massive Terrains using Chunked Level of Detail Control“, *SIGGRAPH, 'Super-size it! Scaling up to Massive Virtual Worlds' course*, San Antonio, Juli 2002, URL: <http://tulrich.com/geekstuff/chunklod.html> (→ Seite 33)
- [Wan08] Wang, J., „Modeling and Matching of Landmarks for Automation of Mars Rover Localization“, Dissertation, Ohio State University, 2008 (→ Seite 18)
- [Was99] Washington, R. u. a., „Autonomous rovers for Mars exploration“, *IEEE Aerospace Conference*, Snowmass at Aspen, März 1999, DOI: 10.1109/AERO.1999.794236 (→ Seite 13)
- [Yam97] Yamauchi, B., „A frontier-based approach for autonomous exploration“, *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, Monterey, Juli 1997, DOI: 10.1109/CIRA.1997.613851 (→ Seite 24)
- [Ygu05] Yguel, M. u. a., „Wavelet Occupancy Grids: a Method for Compact Map Building“, *International Conference on Field and Service Robotics (FSR)*, Port Douglas, Juli 2005, DOI: 10.1007/978-3-540-33453-8_19 (→ Seite 52)
- [McE07] McEwen, A. S. u. a., „Mars Reconnaissance Orbiter's High Resolution Imaging Science Experiment (HiRISE)“, *Journal of Geophysical Research* 112(E5), 2007, DOI: 10.1029/2005JE002605 (→ Seite 21)

- [Tre01] Trebi-Ollenu, A. u. a., „Design and Analysis of a Sun Sensor for Planetary Rover Absolute Heading Detection“, *IEEE Transactions on Robotics and Automation* 17(6), 2001, DOI: 10.1109/70.976028 (→ Seite 18)

Online-Quellen

Zugriffsdatum aller Online-Quellen: 23. Mai 2011

- [1] ESA, Übereinkommen zur Gründung einer Europäischen Weltraumorganisation, ESA SP-1300 (DE), 2005, URL: <http://www.esa.int/esapub/sp/sp1300/sp1300DE1.pdf> (→ Seite 6)
- [2] JPL, Robotics Website, Image Gallery, URL: <http://www-robotics.jpl.nasa.gov/applications/applicationImage.cfm?App=11&Image=28> (→ Seite 35)
- [3] JPL, Robotics Website, Image Gallery, URL: <http://www-robotics.jpl.nasa.gov/applications/applicationImage.cfm?App=10&Image=33> (→ Seite 35)
- [4] JPL, Robotics Website, Image Gallery, URL: <http://www-robotics.jpl.nasa.gov/tasks/taskImage.cfm?TaskID=41&tdaID=2486&Image=336> (→ Seite 35)
- [5] JPL, Robotics Website, Video Gallery, SOOPS: Science Operations on Planetary Surfaces, URL: <http://www-robotics.jpl.nasa.gov/tasks/taskVideo.cfm?TaskID=41&tdaID=2486&Video=64> (→ Seite 4)
- [6] JPL, The Maestro Robot Interface Laboratory, URL: <http://www-robotics.jpl.nasa.gov/facilities/facility.cfm?Facility=9> (→ Seite 17)
- [7] ASU, Interactive Mars Map (MOLA Colorized Elevation), URL: http://jmars.mars.asu.edu/maps/?layer=MOLA_Color (→ Seite 8)
- [8] BULLET, Physics Library, URL: <http://bulletphysics.org> (→ Seite 73)
- [9] Blender, 3D Content Creation Suite, URL: <http://www.blender.org> (→ Seiten 31, 45)
- [10] DLR, Projekt BIRD, URL: http://www.dlr.de/os/desktopdefault.aspx/tabid-3493/5392_read-7874 (→ Seite 9)
- [11] DLR, Projekt RapidEye, URL: http://www.dlr.de/rd/desktopdefault.aspx/tabid-2440/3586_read-5336 (→ Seite 9)
- [12] Digital Spaces, 3D Realtime Multimedia and Simulation Engine, URL: <http://www.digitalspaces.net> (→ Seite 39)
- [13] DigitalSpace, Lunar Surface Access Module (LSAM), Image Gallery, URL: http://www.digitalspace.com/projects/index/251.html?vsig251_0=19 (→ Seite 39)
- [14] DigitalSpace, STS-125 Simulation, Image Gallery, URL: http://www.digitalspace.com/projects/index/257.html?vsig257_0=17 (→ Seite 39)

- [15] ESA, Cosmic Vision 2015-2025, URL: <http://sci.esa.int/science-e/www/area/index.cfm?fareaid=100> (→ Seite 7)
- [16] ESA, ExoMars Rover Image, URL: <http://exploration.esa.int/science-e/www/object/index.cfm?fobjectid=47068> (→ Seite 1)
- [17] ESA, ExoMars, Scientific objectives, URL: http://www.esa.int/SPECIALS/ExoMars/SEMOVIAMS7F_0.html (→ Seite 6)
- [18] ESA, New planetary simulation software puts ESA rover designs to the test, URL: http://www.esa.int/SPECIALS/Technology/SEM7TDE3GXF_0.html (→ Seite 39)
- [19] Eclipse Foundation, Rich Client Platform, URL: http://wiki.eclipse.org/index.php/Rich_Client_Platform (→ Seite 61)
- [20] ExoPlanet, Projektseite, URL: <http://redfibre.net/orbital/projects/exoplanet> (→ Seiten 47, 48, 74)
- [21] Google, 3D Warehouse, Mars Exploration Rover Spirit and Opportunity, URL: <http://sketchup.google.com/3dwarehouse/details?mid=9f1021ec701fd96dc82b8fee1057b30> (→ Seite 44)
- [22] Hugin, Panorama Photo Stitcher, URL: <http://hugin.sourceforge.net> (→ Seite 43)
- [23] JPL, Robotics Website, Systems, The Rocky 7 Rover, URL: <http://www-robotics.jpl.nasa.gov/systems/system.cfm?System=2> (→ Seite 10)
- [24] JPL, Multi-mission Instrument Processing Laboratory (MIPL), The VICAR Image Processing System, URL: <http://www-mipl.jpl.nasa.gov/vicar.html> (→ Seite 43)
- [25] Javaclient 3.0, Client for Player/Stage, URL: <http://java-player.sourceforge.net> (→ Seite 62)
- [26] MRPT, The Mobile Robot Programming Toolkit, URL: <http://www.mrpt.org> (→ Seite 74)
- [27] NASA, Mars Science Laboratory Mission, Rover Fast Facts, URL: <http://marsprogram.jpl.nasa.gov/msl/mission/rover> (→ Seite 9)
- [28] NASA, NASA Open Source Software, URL: <http://ti.arc.nasa.gov/opensource> (→ Seite 75)
- [29] NASA, Open Source Summit 2011, URL: <http://www.nasa.gov/open/source> (→ Seite 75)
- [30] NASA, The Mars Exploration Program's Science Theme, URL: <http://mars.jpl.nasa.gov/science> (→ Seite 6)
- [31] NASA, JPL, Mars Science Laboratory, Artist's Concept, URL: <http://marsprogram.jpl.nasa.gov/msl/multimedia/images/?ImageID=3451> (→ Seite 1)
- [32] NASA, JPL, Planetary Photojournal, PIA01466: Improved MPF 360-degree Color Panorama, URL: <http://photojournal.jpl.nasa.gov/catalog/PIA01466> (→ Seite 44)

- [33] NASA, JPL, Spirit: Detailed Traverse Maps, URL: <http://marsrover.nasa.gov/mission/tm-spirit/spirit-sol2526.html> (→ Seite 63)
- [34] NASA, JPL, Cornell University, Planetary Photojournal, PIA01907: 'McMurdo' Panorama from Spirit's 'Winter Haven', URL: <http://photojournal.jpl.nasa.gov/catalog/PIA01907> (→ Seite 42)
- [35] NASA, JPL, Cornell University, Planetary Photojournal, PIA04413: Artist's Concept of Rover on Mars, URL: <http://photojournal.jpl.nasa.gov/catalog/PIA04413> (→ Seite 1)
- [36] NASA, JPL, Cornell University, Planetary Photojournal, Super-Resolution View of Endeavour Rim, Sol 2239, URL: <http://photojournal.jpl.nasa.gov/catalog/PIA13197> (→ Seite 19)
- [37] NASA, JPL, Malin Space Science Systems, Planetary Photojournal, Opportunity Amid Mars Craters, URL: <http://photojournal.jpl.nasa.gov/catalog/PIA13196> (→ Seite 19)
- [38] NASA, JPL, University of Arizona, USGS, HiRISE Digital Terrain Models, Mars Exploration Rover Landing Site at Gusev Crater, URL: http://hirise.lpl.arizona.edu/dtm/dtm.php?ID=PSP_001513_1655 (→ Seiten 8, 31, 34, 41, 42, 47, 63)
- [39] ODE, Open Dynamics Engine, URL: <http://www.ode.org> (→ Seite 73)
- [40] OGRE 3D, Open Source 3D Graphics Engine, URL: <http://www.ogre3d.org> (→ Seite 39)
- [41] OpenCV, Open Source Computer Vision, URL: <http://opencv.willowgarage.com> (→ Seite 74)
- [42] PythonWare, Python Imaging Library (PIL), URL: <http://www.pythonware.com/products/pil/index.htm> (→ Seite 43)
- [43] SpatiaLite, A complete Spatial DBMS in a nutshell, URL: <http://www.gaia-gis.it/spatialite> (→ Seite 73)
- [44] TU Berlin, Fachbereich Raumfahrttechnik, Projekt Moon Rover, URL: http://www.raumfahrttechnik.tu-berlin.de/menue/studium_und_lehre/projekte (→ Seite 10)
- [45] The Player Project, Free Software tools for robot and sensor applications, URL: <http://playerstage.sourceforge.net> (→ Seite 59)
- [46] The Player Project, Interface specifications, URL: http://playerstage.sourceforge.net/doc/Player-2.1.0/player/group_interfaces.html (→ Seite 60)
- [47] University of Plymouth, Centre for Robotics and Neural Systems (CRNS), Evolution in Robotic Islands - Project Wiki, URL: <http://marsroversim.wikiki.com> (→ Seite 30)
- [48] Universitätsbibliothek der Humboldt-Universität zu Berlin, Fahrerloses Transportsystem (FTS), URL: http://info.ub.hu-berlin.de/bibliothek/zweigbibliotheken/nawi/vr/deuvr/vr_fts.html (→ Seite 11)

- [49] ogre-paged, Paged Geometry for Ogre3D, URL: <http://code.google.com/p/ogre-paged> (→ Seite 46)