Hazard Relative Navigation

Towards safe autonomous planetary landings in unknown hazardous terrain

Woicke, Svenja

DOI
10.4233/uuid:a638c550-0d30-41df-9d49-4f935890bd2b

Publication date
2019

Document Version
Final published version

Citation (APA)
Woicke, S. (2019). *Hazard Relative Navigation: Towards safe autonomous planetary landings in unknown hazardous terrain*. [Dissertation (TU Delft), Delft University of Technology].
https://doi.org/10.4233/uuid:a638c550-0d30-41df-9d49-4f935890bd2b

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

HAZARD RELATIVE NAVIGATION

TOWARDS SAFE AUTONOMOUS PLANETARY LANDINGS IN UNKNOWN HAZARDOUS TERRAIN

HAZARD RELATIVE NAVIGATION

TOWARDS SAFE AUTONOMOUS PLANETARY LANDINGS IN UNKNOWN HAZARDOUS TERRAIN

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. dr. ir. T. H. J. J. van der Hagen, voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 25 maart 2019 om 12:30 uur

door

Svenja WOICKE

Ingenieur in de Luchtvaart- en Ruimtevaarttechniek,
Technische Universiteit Delft, Delft, Nederland,
geboren te Minden, Duitsland.

Dit proefschrift is goedgekeurd door de

promotor: Prof. dr. ir. P. N. A. M. Visser
copromotor: Dr. ir. E. Mooij

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. P. N. A. M. Visser,	Technische Universiteit Delft
Dr. ir. E. Mooij,	Technische Universiteit Delft
Onafhankelijke leden:	
Prof. dr. J. de Lafontaine	Université de Sherbrooke, Canada
Prof. dr.-ing. E. Stoll	Technische Universität Braunschweig, Duitsland
Prof. dr. ir. A. Hanssen	Technische Universiteit Delft
Prof. dr. E. K. A. Gill	Technische Universiteit Delft
H. Krüger	Deutsches Zentrum für Luft- und Raumfahrt, Duitsland
Prof. dr. L. L. A. Vermeersen,	Technische Universiteit Delft, reservelid

Keywords: Hazard detection, hazard relative navigation, terrain relative navigation, planetary landing, Moon

Printed by: Ipskamp Printing

Front & Back: An artistic interpretation of a hazard map originating from this work inspired by the Dutch master Vincent van Gogh.

ISBN 978-94-028-1413-2

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

CONTENTS

List of Abbreviations

ALHAT	Autonomous Landing and Hazard Avoidance Technology
CSA	Canadian Space Agency
DEM	Digital Elevation Model
DIMES	Descent Image Motion Estimation System
DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Centre)
EKF	Extended Kalman Filter
ESA	European Space Agency
ESKF	Error-State Kalman Filter
FN	False Negative
FORSTERNAV	Flash Optical Sensor for Terrain Relative Robotic Navigation
FOV	Field of View
FP	False Positive
GLCM	Gray-Level-Co-Occurrence Matrix
GNC	Guidance, Navigation, and Control
GPS	Global Positioning System
HD	Hazard Detection
HDA	Hazard Detection and Avoidance
HILT	Hardware-in-the-Loop Testing
HiRISE	High Resolution Imaging Science Experiment
HRN	Hazard-Relative Navigation
IMU	Inertial Measurement Unit
IVN	Integrated Vision and Navigation
KF	Kalman Filter
LAPS	Lidar-based Planetary Landing System
LL	Lunar Lander
LRO	Lunar Reconnaissance Orbiter
MEKF	Multiplicative Extended Kalman Filter
MEPAT	Mars Exploration Advanced Technologies Program
MER	Mars Exploration Rovers
MRO	Mars Reconnaissance Orbiter
MRSM	Mars Rover Sample Return Mission
MSL	Mars Science Laboratory
NASA	National Aeronautics and Space Administration
NCC	Normalised Cross-Correlation
PANGU	Planet and Asteroid Natural scene Generation Utility
RMS	Root Mean Square
SAD	Sum of Absolute Differences
SEI	Space Exploration Initiative

SfM	Shape-from-Motion
SfS	Shape-from-Shading
SILT	Software-in-the-loop Testing
SLAM	Simultaneous Localisation and Mapping
SSD	Sum of Squared Differences
ST9	Space Technologies 9 Program
SURF	Speeded-up Robust Features
SV	Stereo Vision
TAN	Terrain-Absolute Navigation
TN	True Negative
TP	True Positive
TRL	Technology Readiness Level
TRN	Terrain-Relative Navigation
TRON	Testbed for Robotic Optical Navigation
VBHDA	Vision-based Hazard Detection and Avoidance
VBRNAV	Vision-based Relative Navigation Techniques

List of Symbols

Roman

a	Coefficient of a plane	[−]
a	Surface albedo	[−]
$\boldsymbol{a}$	Acceleration	[m/s^2]
b	Coefficient of a plane	[−]
b	Baseline of stereo set-up	[m]
$\boldsymbol{b}$	Bias	
$\boldsymbol{B}$	Control matrix	
c	Coefficient of a plane	[−]
C	Cost function	
d	Disparity	[pixel]
e	Sun elevation	[°]
f	Camera's focal length	[°]
$\boldsymbol{F}$	System matrix	
$\boldsymbol{g}$	Gravity	[m/s^2]
$\boldsymbol{G}$	System Noise Matrix	
$\boldsymbol{H}$	Measurement matrix	
i	Instance in time	[−]
I	Image intensity	[−]
$\boldsymbol{I}_x$	Identity matrix of size $x \times x$	
$\boldsymbol{K}$	Kalman filter gain	
$\boldsymbol{n}$	Gaussian white noise	
$\hat{\boldsymbol{n}}$	Surface normal (vector)	
p	Pixel coordinate (in left to right direction)	[pixel]
p	Slope in x-direction	[°]
$\boldsymbol{p}$	Vector part of a quaternion	[−]
$\boldsymbol{P}$	Covariance matrix	
q	Pixel coordinate (in up to down direction)	[pixel]
q	Slope in y-direction	[°]
$\boldsymbol{q}$	Quaternion	[−]
$\boldsymbol{Q}$	System covariance matrix	
r	radius	
$\boldsymbol{r}$	Position	[m]
R	Roughness	[m]
$\boldsymbol{R}$	Rotation matrix	
$\boldsymbol{R}$	Measurement covariance matrix	
s	image size	[pixel]
S	Slope	[°]

$\boldsymbol{S}$	Kalman filer innovation covariance	
t	time	[s]
t_{go}	Time-to-go	[s]
$\boldsymbol{T}$	Transformation matrix	
u	x-velocity	[m/s]
$\boldsymbol{u}$	(Control) inputs	
v	y-velocity	[m/s]
$\boldsymbol{v}$	Measurement noise	
$\boldsymbol{v}$	Velocity	[m/s]
$\boldsymbol{w}$	Noise	
x	x-coordinate	[m]
$\boldsymbol{x}$	State	
$\hat{\boldsymbol{x}}$	State estimate	
$\boldsymbol{X}$	Jacobian	
y	y-coordinate	[m]
z	z-coordinate	[m]
$\boldsymbol{z}$	Measurement equation	
Greek		
δd	Disparity step/minimum disparity	[pixel]
δz	Stereo depth resolution	[m]
$\delta\theta$	Error representation of orientation	
μ	Mean	
σ	Standard deviation	
σ^2	Variance	
τ	Sun direction in image	[°]
$\boldsymbol{\omega}$	Angular rate	[rad/s]

SUMMARY

Many successful landings have been performed on celestial bodies such as Mars, the Moon, Venus and others. All of these had in common that they were designed such that they had to land in regions, which were supposedly free of any hazards or that a certain level of risk was accepted. However, while rocks and other geological features are nightmares of any landing engineer they are the dream targets of scientists. Therefore, currently landing-site selection is a trade-off between the scientists' wishes and the engineers' fears.

To bring the engineering capabilities closer to what the scientists desire, landing capabilities need to be advanced. Therefore, this work tries to answer the research question:

> **Are autonomous safe landings in hazardous and potentially unknown environments possible?**

which lead to the following two sub-questions:

1. How can a landing vehicle autonomously assess the safety of a potentially unknown and unmapped landing site?

2. How can a landing vehicle ensure a safe touch down avoiding autonomously detected hazards?

To answer these question two methods are developed in this work: a hazard-detection algorithm, capable of autonomous assessment of the landing region, and a hazard-relative navigation algorithm, enabling precise touch-down relative to the detected hazards and selected safe landing site. Both methods were thoroughly tested both in a software, but also in a hardware-in-the-loop environment.

From a study of three feasible camera-based hazard-detection technologies, stereo-vision-based hazard-detection is found to be the most feasible candidate for on-board hazard detection and landing-site assessment. Therefore, a stereo method is implemented to reconstruct three-dimensional surface maps from a pair of descent input images. Based on these maps, the slope and roughness of the landing region is computed. In addition, the terrain texture and illumination is assessed. From this information a hazard map of the landing region can be constructed, enabling the autonomous selection of a safe landing site.

A thorough sensitivity study of this algorithm using software-in-the-loop tests showed that the algorithm can perform hazard assessment at altitudes of 200 m and lower at camera baselines of 2 m and less. Baselines in this order were found to be feasible for current lander designs (for example, the ESA Lunar Lander or the NASA Mars Science Laboratory). Enabling stereo-based hazard detections at altitudes of 200 m represents

an improvement of a factor of 2 with respect to the only known prior study conducted in this area.

Moreover, this research demonstrates that based on the resulting hazard maps selecting a safe landing site is possible. Out of the entire landing-region map, only 1% or less of all sites were wrongly identified as safe sites while actually being unsafe.

After extensive testing in a software environment using artificial images simulated by a software, PANGU, based on a lunar analogue surface model, the next step was to validate the performance using real input images. To this end the testbed for relative optical navigation (TRON) at DLR Bremen was used. This is a facility where real images of a lunar analogue surface model can be acquired, alongside ground-truth state measurements.

The hardware-in-the-loop tests of the hazard-detection method showed that successful selection of a safe landing site is still possible even when using real images with the associated problems, such as noise, problematic illumination and the challenges of camera calibration. However, the maximum percentage of undetected hazardous sites increased to only 2.5%.

As the successful development of a hazard-detection function was the prerequisite for the development of hazard-relative navigation methods, this step was taken next.

Linking the hazard detection and a relative-navigation method by using the computed hazard-detection surface maps as an input for the navigation filter is a novel approach. This idea enables a hazard (map) relative navigation without the addition of further errors from linking the hazard maps and the navigation output.

This approach was implemented by following the paradigm of simultaneous localisation and mapping (SLAM), frequently used to drive robots in unknown surroundings. Here, map measurements are used for updating a navigation filter and thus achieving more precise and accurate state knowledge.

Based on the robustness and computational efficiency, an error-state Kalman filter was used as a state observer. In a SLAM manner the hazard-map features are appended to the state and are thus also predicted and updated.

The developed filter was first tested during extensive software-in-the-loop testing. To date, the very final phase of the descent, as studied in this work, is flown on IMU-only propagation. This method is used as a benchmark. Final hazard-relative landing ellipses of 20×20 m were achieved opposed to 60×60 m of the current state-of-the-art benchmark method. Precisions of 10 m to 20 m are required for the successful implementation of hazard avoidance, thus hazard avoidance is possible using the proposed filter.

Moreover, it was proven that the filter removes 99% of all errors in the altitude measurements as compared to the benchmark, and is thus capable of very accurate and precise altitude estimation.

On a set of 500 runs less than 1% of outliers occurred, demonstrating that the method is not only accurate and precise, but also robust. An outlier is defined as any execution where the final error is higher than the final error achieved without the filter, *i.e.*, any situation where the filter performs worse than pure IMU propagation.

During an other validation campaign at TRON at DLR Bremen, it was found that the hazard relative navigation method was capable of performing even better using these real images, the hazard-relative landing ellipse size could be further reduced to 6×9 m,

which is an improvement of more than a factor of 2 as opposed to the software-in-the-loop tests. This further improvement is likely linked to the infinite resolution of the TRON terrain as opposed to the finite resolution of the terrains used during the software tests.

On the altitude component, the results were slightly less accurate than the software-int-the-loop results, which is in-line with the findings from the hazard-detection testing. Still, the altitude is estimated very well, with a error reduction of 97%. Also during the hardware test the method proved to be robust and no outliers were generated.

Concluding, the feasibility of hazard-relative navigation was demonstrated, the precisions achieved are clearly good enough for the successful avoidance of hazards detected in the landing site. Using the hazard-detection method it is possible to select a safe landing site autonomously on-board.

SAMENVATTING

Veel succesvolle landingen zijn uitgevoerd op hemellichamen, zoals Mars, de Maan, Venus en anderen. Al deze landingen hadden gemeen dat ze werden ontworpen voor gebieden waarvan werd aangenomen dat er geen gevaren waren, of dat een bepaald risico werd aanvaard. Maar terwijl stenen en andere geologische kenmerken een nachtmerrie zijn voor iedere landingsingenieur, zijn zij de droom van iedere wetenschapper. De selectie van een landingszone is dan ook een afweging tussen de wensen van de wetenschappers en de angsten van de ingenieurs.

Om de technische mogelijkheden beter te laten aansluiten bij de wensen van wetenschappers, moet vooral het vermogen om te landen worden verbeterd. Daarom is dit werk gericht op het beantwoorden van de onderzoeksvraag:

> **Is het mogelijk om autonoom te landen in gevaarlijke en mogelijk onbekende omgevingen?**

wat leidt tot de volgende twee subvragen:

1. Hoe kan een lander autonoom de veiligheid van een mogelijk onbekende en niet in kaart gebrachte landingsplaats inschatten?
2. Hoe kan een lander een gegarandeerd veilig landen, waarbij autonoom gedetecteerde gevaren worden vermeden?

Om deze vragen te beantwoorden zijn in dit werk twee methoden ontwikkeld: een algoritme om gevaren te detecteren en autonoom de landingsplaats te beoordelen, en een gevaren-relatief navigatie-algoritme, dat het mogelijk maakt om een precieze landing uit te voeren, relatief ten opzichte van de gedetecteerde gevaren en de geselecteerde veilige landingsplaats. Beide methoden zijn uitgebreid getest, niet alleen in een softwareomgeving, maar ook met “hardware in the loop”.

Na bestudering van drie mogelijke gevaren-detectie algoritmes die gebruik maken van camera’s, blijkt detectie op basis van stereovisie de meest haalbare kandidaat voor gevarendetectie aan boord, en voor beoordeling van de landingsplaats. Een stereovisiemethode is derhalve geïmplementeerd om driedimensionale kaarten van het oppervlak te construeren op basis van twee simultaan genomen foto’s tijdens de afdaling. Op basis van deze kaarten wordt de helling en ruigheid van de landingsplaats berekend. Bovendien wordt een inschatting gemaakt van de textuur en belichting van het terrein. Met deze informatie kunnen vervolgens de gevaren op de landingsplaats in kaart worden gebracht, waarmee autonome selectie van een veilige landingsplaats mogelijk wordt.

Een uitgebreide gevoeligheidsanalyse van dit algoritme op basis van software-in-the-loop tests liet zien dat het algoritme gevaren kan inschatten vanaf maximaal 200 m hoogte, met een onderlinge afstand tussen de camera’s van 2 m of minder. Een dergelijke onderlinge afstand bleek haalbaar te zijn voor huidige landerontwerpen (waaronder de ESA Lunar Lander en het NASA Mars Science Laboratory). De mogelijkheid om

op 200 m hoogte gevaren te detecteren met stereovisie betekent een verbetering met een factor twee ten opzichte van de enige andere bekende studie die eerder op dit gebied is gedaan.

Dit onderzoek laat bovendien zien dat het mogelijk is om een veilige landingsplaats te selecteren op basis van de resulterende gevarenkaarten. Gemeten over de hele kaart werd slechts maximaal 1% van alle locaties foutief aangewezen als veilige landingsplaats.

Na uitgebreid testen in een softwareomgeving met kunstmatige beelden, gegenereerd door de PANGU software op basis van een maanachtig oppervlaktemodel, werden de prestaties gevalideerd met echte foto's als input. Hiervoor werd het Testbed for Relative Optical Navigation (TRON) bij DLR Bremen gebruikt. In deze faciliteit kunnen echte foto's gemaakt worden van een maanachtig oppervlaktemodel, naast nulmetingen van de voertuigtoestand (positie en snelheid).

Uit de hardware-in-the-loop tests van het gevaren-detectie algoritme bleek dat het ook op basis van echte beelden mogelijk is om een veilige landingsplaats te selecteren, ondanks de problemen die foto's met zich meebrengen, zoals ruis, problematische belichting en problemen met de camera-calibratie. Ondanks alles steeg het maximale percentage van niet gedetecteerde gevaarlijke locaties tot slechts 2,5%.

De volgende stap was de ontwikkeling van gevaar-relatieve navigatiemethoden, die gebruik maken van het ontwikkelde gevaren-detectie algoritme.

Het koppelen van de gevarendetectie en de gevaar-relatieve navigatie, waarbij de berekende oppervlaktekaarten voor gevarendetectie als input dienen voor het navigatiefilter, is een nieuwe benadering van het probleem. Dit concept maakt het mogelijk om gevaar-relatieve navigatie te implementeren zonder de fouten die kunnen ontstaan bij het koppelen van de gevarenkaarten en de navigatie-uitvoer.

Deze methode werd geïmplementeerd volgens het Simultaneous Localization And Mapping (SLAM) paradigma. Dit paradigma wordt regelmatig gebruikt om robots op onbekend terrein te laten rijden. In dit geval worden gemeten kaarten gebruikt om het navigatiefilter te updaten, wat leidt tot een meer precieze en accurate kennis van de voertuig positie en snelheid.

Vanwege zijn betrouwbaarheid en beperkte rekentijd werd een error-state Kalman filter gebruikt als waarnemer van de positie en snelheid. De opvallende kenmerken van de gevarenkaart worden op een SLAM manier hieraan toegevoegd, en daarmee ook voorspeld en gecorrigeerd.

Allereerst werd het ontwikkelde filter uitvoerig getest in een software-in-the-loop setting. Daarmee werden uiteindelijk gevaar-relatieve landingsellipsen van 20×20 m verkregen, in tegenstelling tot de 60×60 m resulterend uit de bestaande state-of-the-art methode. Voor een succesvolle implementatie van gevaarontwijking is een precisie van 10 m to 20 m vereist, die dus geleverd kan worden door het voorgestelde filter.

Bovendien werd aangetoond dat vergeleken met de standaard methode het filter 99% van de fouten in de hoogtemetingen verwijdert, en daarmee in staat is de hoogte zeer precies en accuraat te schatten.

In een totaal van 500 simulaties kwam slechts 1% uitschieters voor, waarmee is aangetoond dat de methode niet alleen precies en accuraat is, maar ook robuust. Een uitschieter is hier gedefinieerd als een simulatie waarbij de uiteindelijke fout hoger is dan de fout die zonder filter zou worden gemaakt; met andere woorden, gevallen waarin het

filter minder goed presteert dan pure traagheidspropagatie.

Tijdens de validatie in TRON bij DLR Bremen presteerde de gevaar-relatieve navigatiemethode nog beter met de echte beelden, zodanig dat de gevaar-relatieve landingsellips verder verkleind kon worden tot $6\times 9\,\mathrm{m}$, een verbetering van meer dan een factor twee ten opzichte van de software-in-the-loop tests. Deze verbetering kan waarschijnlijk worden verklaard uit het feit dat het terreinmodel in TRON een oneindige resolutie heeft, in tegenstelling tot de digitale terreinen in de softwaretests.

Wat de hoogte betreft waren de resultaten iets minder nauwkeurig dan die van de software-in-the-loop testen, zoals te verwachten was na de bevindingen in de gevarendetectie testen. De schatting van de hoogte is echter nog altijd bijzonder goed, met een 97% reductie van de fout. Bovendien bleek de methode tijdens de hardware testen robuust en werden er geen uitschieters geproduceerd.

Al met al mag worden geconcludeerd dat de haalbaarheid van gevaar-relatieve navigatie is aangetoond, daar de behaalde precisie duidelijk volstaat voor het succesvol vermijden van gedetecteerde gevaren op de landingsplaats. De gevaren-detectie methode maakt het mogelijk dat een lander aan boord, autonoom een veilige landingsplaats selecteert.

ZUSAMMENFASSUNG

Seit dem Beginn des Raumfahrtzeitalters ist es das Bestreben von Wissenschaftlern der ganzen Welt, die Oberfläche von Himmelskörpern unseres Sonnensystems näher zu erkunden. Dieses Bestreben hat sich mittlerweile vielfach erfüllt, in dem spezielle Landegeräte auf dem Mond sowie den Planeten Mars und Venus landeten.

Bisher hatten alle Versuche Landgeräte auf der Oberfläche zu positionieren eines gemeinsam: sie sollten in solchen Regionen aufsetzen, die als möglichst sicher gelten und nur möglichst wenige Hindernissen aufweisen. Zwar reduziert dieser Ansatz das Risiko, dass das Landegerät durch gefährliches Terrain, wie zum Beispiel Felsbrocken, beschädigt oder sogar zerstört wird.

Allerdings sind gerade solche Territorien häufig geologisch divers und daher aus wissenschaftlicher Sicht am interessantesten. Es entsteht deshalb ein Konflikt zwischen dem wissenschaftlichen Interesse eine möglichst komplexe Landestelle auszuwählen und dem technologischen Interesse eine möglichst sichere Landestelle zu wählen. Ziel dieser Arbeit ist es, der Lösung dieses Konflikts einen Schritt näher zu kommen und folgende übergeordnete Frage zu beantworten:

> **Ist es mit Hilfe neuer Algorithmen möglich, Landegeräte sicher und autonom in hindernisreichen und unbekannten Regionen von Himmelskörpern in unserem Sonnensystem aufzusetzen?**

Diese Frage lässt sich in zwei weitere Teilfragen unterteilen:

1. Wie kann ein Landegerät eigenständig die Sicherheit einer unbekannten und unkartierten Landestelle bewerten?
2. Wie kann ein Landegerät eigenständig möglichen Hindernissen ausweichen, um so sicher zu landen?

Um diese Fragen zu beantworten, stellt diese Arbeit zwei neue Methoden vor. Erstens, einen Hindernis-Detektionsalgorithmus (HD Algorithmus) zur autonomen Bewertung der Sicherheit von Landestellen. Und zweitens, Algorithmus zur autonomen Relativnavigation im Bezug auf Hindernisse (Hindernis-Relativnavigation, HRN). In der Kombination ermöglichen diese Algorithmen eine sichere und präzise Landung relativ zu den detektierten Hindernissen. Beide Algorithmen wurden in einer Software-Simulationsumgebung sowie in einer Laborumgebung mit Hardware getestet.

In einer Vergleichsstudie wurden drei verschiedene Kamera-basierte HD Algorithmen gegeneinander abgewogen. Als beste Lösung wurde ein stereoskopischer Algorithmus ausgewählt. Mittels einer Reihe von zwei Stereo-Bildern, die während des Landeanflugs aufgenommen werden, erstellt dieser Algorithmus zunächst dreidimensionale Karten der Landestelle. Danach werden diese Karten weiterverarbeitet um Gefälle, Unebenheiten, Schatten und die Beleuchtungsbedingungen auf der Oberfläche zu detektieren. Zur Bewertung der Sicherheit einer Landung werden diese Informationen dann

zusammengeführt um eine Hinderniskarte des Landegebietes zu erstellen, auf deren Basis eine sichere Landestelle ausgewählt werden kann.

Diese Arbeit zeigt, dass es mithilfe der berechneten Hinderniskarten möglich ist, sichere Landestellen auszuwählen. In einen Referenzfall wurde nur 1% der abgedeckten Fläche als sicher eingestuft, obwohl diese zunächst als eher unsicher einzustufen ist.

Um den Anwendungsbereich sowie die Robustheit dieses Algorithmus zu bestimmen, wurde eine Sensitivitätsanalyse in einer Software-Simulationsumgebung durchgeführt. In dieser Simulationsumgebung wurde die Software "PANGU" zur Generierung der Kamerabilder verwendet. Diese Analyse zeigt, dass der Algorithmus ab einer Höhe von 200 m bei einer Stereobasis von 2 m sicher in der Lage ist Hindernisse zu erkennen. Dieses verbessert die mögliche Anfangshöhe für den Beginn der Hinderns-Detektion um das Doppelte anderer bisher veröffentlichter Studien. Eine Stereobasis von 2 m ist mit den Abmaßen heutiger Landegeräte kompatibel (z.B. ESA Lunar Lander, NASA Mars Science Laboratory).

Nach den umfangreichen Tests mithilfe von PANGU wurde der Algorithmus mit echten Kamerabildern in einer Laborumgebung getestet. Diese Tests wurden im Testbett für Robotische Optische Navigation (TRON) des Instituts für Raumfahrtsystem am Deutschen Zentrum für Luft- und Raumfahrt (DLR-RY) durchgeführt. TRON verfügt über ein physisches Terrainmodell dass der Mondoberfläche ähnelt. Für dieses Terrainmodell steht ein hoch genaues digitalisiertes 3D Modell zur Verfügung, welches als hochpräzise Referenz für das durch den Algorithmus erstellte stereoskopische Modell der Oberfläche dient. In dem Labor wurde der Landeanflug eines Landegeräts auf den Mond durch einen Industrieroboter, an dem zwei Kameras montiert waren, simuliert.

Die Tests mit TRON zeigen, dass der Algorithmus auch in einer realistischen Umgebung zuverlässig sichere Landestellen erkennen kann. Trotz Effekten - wie zum Beispiel dem Rauschen der Kamerasensoren, realistischer Beleuchtungsbedingungen oder der Kalibrierung der Kamera - bewertet der Algorithmus noch 97,5% der Oberfläche korrekt.

Im Anschluss daran wurde der HRN Algorithmus entwickelt, welcher eine Navigationslösung relativ zu den durch die HD-Funktion detektierten Hindernissen berechnet. Die Zustandsdefinition des Landgeräts in seiner räumlichen Orientierung und Position erfolgt auf Basis der generierten Karten, die mit den stereoskopischen Bildern erstellt werden. Dieser Navigationsansatz vermeidet Fehler, welche durch die funktionale Trennung einer optischen Navigationsfunktion und einer HD Funktion entstehen würden, da die Konstellation unmittelbar im Bezug auf die Hindernisse berechnet wird.

Der HRN-Algorithmus basiert auf dem in der Robotik weiterverbreiteten Prinzip der Simultanen Positionsbestimmung und Kartenerstellung (eng. Simultaneous Localisation and Mapping, SLAM). In SLAM-Algorithmen werden Karten als Input für einen Navigationsfilter verwendet, welcher fortlaufend durch stetig gesammelt Sensordaten aktualisiert wird. Dies ermöglicht eine genaue Navigation selbst in unbekannten Umgebungen. Für den HRN-Algorithmus wurde ein Zustandsfehler-bezogener Kalmanfilter (eng. Error-State Kalman Filter) als Zustandsbeobachter gewählt, welcher sich durch besondere Robustheit und Effizienz auszeichnet. Die in den HD-Karten detektierten Hindernisse werden in diesem Filter dem Zustandsvektor beigefügt, und damit auch propagiert.

Wie auch der HD-Algorithmus wurde der Filter in einer Software-Simulationsumgebung getestet. Während mit klassischen Navigationsmethoden, welche den Zustand

über eine inertiale Messeinheit propagieren, nur eine Genauigkeit von rund 60×60 m m für die Landeellipse im Bezug auf Hindernisse erzielbar ist, erreicht der hier entwickelte HRN-Algorithmus eine Genauigkeit von 20×20 m Um Hindernissen auszuweichen ist eine Genauigkeit von 10 m to 20 mnotwendig, so dass der HRN-Algorithmus zu diesem Zweck eingesetzt werden kann. Darüber hinaus ermöglicht der Algorithmus eine 99% genaue Bestimmung der Höhe des Landegeräts über der Oberfläche.

Die Robustheit der entwickelten Methode wurde durch eine Monte Carlo Simulation mit 500 Durchläufen verifiziert. In der gesamten Simulationskampagne traten nur 1% Ausreißer auf, wobei ein Ausreißer als Endzustand mit schlechterer Landegenauigkeit als ohne HRN definiert ist. Darüber hinaus wurde auch der HRN-Algorithmus im TRON Labor zur Probe gestellt. Hierbei stellte sich heraus, dass der Algorithmus rund doppelt so gut funktioniert wie zuvor in den rein Software-basierten Tests, da die HRN-Landellipse auf 6× 9 m reduziert werden konnte. Diese deutlich Verbesserung ist durch die unendliche Auflösung des Mondmodells im Labor zu erklären, während die Simulation mit PANGU nur eine begrenzte Auflösung als Input bietet. Obwohl sich die Landeellipse deutlich verkleinerte, war die Höhenauflösung etwas schlechter als in der Simulation und konnte nur auf 97% genau bestimmt werden. Während der Labortests konnten keine Ausreißer verzeichnet werden.

Zusammenfassend ist festzustellen, dass der in dieser Arbeit entwickelte HRN-Algorithmus erfolgreich demonstriert wurde. Die hier entwickelte Methode erzielt eindeutige Verbesserungen in der Navigationsgenauigkeit im Bezug auf Hindernisse, welche zum Ausweichen auf sichere Landestellen verwendet werden können. Die HD-Methode ermöglicht die Auswahl sicherer Landestellen autonom und während des Landeanflugs des Landegeräts von Höhen ab 200 m über der Oberfläche.

1

INTRODUCTION

SPACE and space flight have always fascinated humankind. Very early on humans dreamt of the stars and about having the ability of leaving Earth. Many old mythologies include stories where the protagonist(s), often of godly descent, leave the Earth and travel to space. With the advent of telescopes and their subsequent improvement, the planets turned from small dots into something that was observable in detail by humans. This again sparked human imagination. Even before space flight was in reach of the human race, the planets inspired writers for stories about alien civilisations on other planets (for example,"War of the world" by H.G. Wells written in 1897, but even 16th century texts already discussed the possibility of extraterrestrial life), but also about humans exploring these planets (for example, in Johannes Kepler's "Somnium" in 1608 and after that, and potentially also more popularly, in the book "From the Earth to the Moon" by Jules Verne).

After humans finally gained access to space by developing powerful launchers and only shortly after placing the first man-made object in Earth orbit in 1957, it was a logical conclusion to go to the Moon and planets next. This desire together with the political climate during those times led to the "Space race", resulting in multiple successful visits of NASA astronauts to the Earth's satellite. However, to date humans did not step on any celestial body other than the Earth and Earth's Moon. Humankind had to learn that space flight is difficult and expensive. Therefore, to date, robotic explorers are the only possibilities for in-situ exploration of our Solar System and try to answer the fundamental questions of the universe.

So far there have been 27, manned and unmanned, soft-landing attempts to land on the Moon, out of which 19 have been successful.[1] The manned landings were special in the sense that humans were able to actively steer the vehicle during its descent. The vehicle would have been able to land autonomously, but the astronauts on board eventually took over during all Apollo landings (Brady and Paschall, 2010).

[1] mission numbers are based on the list of all extraterrestrial landings published at https://en.wikipedia.org/wiki/List_of_landings_on_extraterrestrial_bodies, visited: 23.04.2018

1

Eleven successful landing attempts were made at Venus. Also Mars has been explored by multiple landers, out of which two are currently still roving the planet. Out of the total 13 landing attempts on Mars only seven were a success. In 2005 a lander was send to the surface of Titan, which successfully landed and conducted its mission. Next to that Hayabusa touched asteroid Itokawa in 2005 (Yoshikawa et al., 2006) and the Philae Lander (triple)-landed on comet 67P in late 2014 (Biele and Ulamec, 2008). This accounts to a total of 54 soft landing attempts, with a success rate of 40.

For the near future, more asteroid landings or touch-and-goes are planned, with two missions currently en-route. Also almost all big space agencies prepare for landings on Mars. The Chinese space agency just landed on the "dark side" of the Moon, and NASA is even investigating to land on Jupiter's moon Europa.

1.1. THE NEED FOR MORE ADVANCED LANDING SYSTEMS

Based on the introduction it seems like humankind has mastered the task of landing on other bodies. This, of course, leads to the question why there is any need to develop a more advanced landing system as proposed in this work. One has to understand that there is still a lot unexplored in our Solar System and there is still plenty of research, which needs to be conducted. Unfortunately, parts of it cannot be achieved using current-day landing technologies.

In this context, it is important to introduce the concept of an *inherently safe* landing site/region. Such a region is defined as an area, which is thought to be, based on orbital images, statistical models, and other observations, free or almost free of any hazards that could cause the lander to fail during touch down. The concept of landing hazards will be discussed in more detail in Chapter 2.

To date all landings where performed in regions that were selected to be inherently safe, *i.e.,* regions, which did not contain any hazards no matter where in the region the vehicle would touch down. However, it is not trivial to select regions that are actually hazard free, as this requires *a-priori* information on the landing site, most importantly high resolution surface images/maps. If 0.5 cm rocks would pose a risk to a lander, all these rocks should be identifiable in these maps. If they are not, one can try to derive the distribution of smaller scale surface features from larger scale surface features identifiable in low-resolution maps.

Unfortunately, this approach can go wrong. The Viking landers (1976, see Holmberg et al. (1980)) are a very good example of a (near) failure of this system. Based on statistics, the landing site of the Viking landers was thought to be inherently safe and free of boulders, while post-landing analysis of the landing region showed that this was not the case. A big boulder was found right next to the Viking 1 landing site. The boulder named "Big Joe" is 2 m wide and 1 m high, while the surface clearance of the Viking landers was 20 cm (Braun and Manning, 2007). Landing on a boulder of this size would have clearly caused landing failure. Since Viking did not feature a precision landing system, it was therefore sheer luck that the lander ended up next to and not on top of the boulder.[2] Post landing (in 1995) it was computed that the actual probability of a landing failure

[2] High resolution picture of Big Joe and discussion can be found at: https://www.lpi.usra.edu/publications/slidesets/winds/slide_28.html visited: 15.03.2018

was 20% (Johnson et al., 2002), while the landing sites were actually selected to be 99% "landable" (Ezell and Ezell, 1984). However, if the conclusions drawn from this incident are that high-resolution surface maps are required prior to any landing mission, this would mean that it is not possible to perform such a mission without sending an orbiter first. This does not only make missions even more costly, but also adds to the total time a mission will take.

Since the launch of the Mars Reconnaissance Orbiter with its powerful HiRise camera, high-resolution images of Mars do exist (Graf et al., 2005). However, to date the orbiter has not mapped the full surface of the planet. One might conclude that NASA has now solved the problem of identifying safe sites, but that is only part of the truth. Since landing missions on Europa, Venus or an asteroid are on the agenda of the big space agencies, the problem is not overcome yet. All of these bodies are currently not even mapped to the resolution Mars was prior to the Viking touchdowns. Venus is impossible to map with visual-light cameras due to its atmosphere, while the harsh radiation environment around Europa does not really make a long-term mapping mission feasible.

The findings reported by Brady and Paschall (2010) analysing the Apollo landings (1968-1972), conclude that all Apollo landings were at risk if the human pilots on board would not have intervened. For all six successful Apollo landings, each one faced at least two of the hazards identified in the work: dust, craters, slopes, and rocks. The authors' conclusion is that for the return to the Moon, new landing strategies need to be developed. Here, specifically hazard-detection and avoidance is named as one of the possible candidates to increase landing safety.

Moreover, there is proof (for example, in Viking surface images) that the Martian surface changes over time (*e.g.*, by erosion). To date, little is known about this phenomenon. This means that it might be risky to rely on "old" maps, while orbiters can only cover selected landing site at given intervals (often long intervals).

NASA's current approach on Mars is to land in areas that are scientifically more interesting, but thus also more challenging from an engineering standpoint. These sites usually contain hazards within the landing region and are not inherently safe [3]. This leads to two developments: decreasing the size of the landing ellipse, thus performing more precise landings, and development of hazard-detection and avoidance systems. Having these capabilities, larger portions of the Martian surface will actually be feasible as a landing region. In 2008 Huertas et al. (2008) predicted that inclusion of hazard detection will triple the area accessible for Mars and considerably decreases the risk of a landing failure by a factor of four.

To end with, it has to be stressed that avoiding hazards is not possible with current-day landing accuracies, as very accurate navigation of the lander with respect to the surface hazards is necessary. Therefore, both hazard detection and hazard-relative navigation techniques need to be developed to reach the goal of more advanced landings, in hazardous regions or on unmapped bodies, and by that paving the ground for next-generation missions aiming for more challenging landing sites.

[3] The proceedings of the 3rd Mars2020 landing site selection workshop can be found at https://marsnext.jpl.nasa.gov/workshops/wkshp_2017_02.cfm, visited 19.12.2017

1

1.2. RESEARCH AIM AND METHODOLOGY

In the foregoing discussion it was established that there is a need to enable more precise landings along with the ability to avoid (unknown) surface hazards during touch-down. This leads to the top-level research question:

> Are autonomous safe landings in hazardous and potentially unknown environments possible?

To answer this question two sub questions have to be addressed:

1. How can a landing vehicle autonomously assess the safety of a potentially unknown and unmapped landing site?

2. How can a landing vehicle ensure a safe touch down avoiding autonomously detected hazards?

This research aims to contribute to the current developments and answering the two previously stated questions, by attempting to advance hazard-detection technology and to enable more precise navigation to avoid these hazards. This leads to the following main goals:

1. Development of an autonomous, on-board hazard-detection function to assess the safety of a landing region.

2. Enabling precise landing within these landing region by precise navigation, relative to any detected hazards.

Since both navigation and hazard avoidance are very closely linked, this research tries to combine both systems by fusing the outputs of hazard detection with a hazard-relative navigation filter. The idea is that navigation will directly be performed relative to the detected hazards and will thus enable a safe and precise touchdown.

This work aims therefore to answer the question whether it is possible to use the hazard detection and avoidance (HDA) outputs as a navigation input and thus to achieve more precise and safer landings. It represents the first approach of closely linking these two capabilities and thus serves as a proof of concept, but does not aim to develop flight-ready software and/or hardware.

Hazard detection is for the largest part a map-building method. In the field of robotics, simultaneous localisation and mapping (SLAM) is a commonly used technique to navigate in an unknown environment, while building a map of this environment at the same time. This research follows a SLAM-like approach. However, due to the size of the hazard maps, as well as computational constraints, no full SLAM can be used in this scenario. Still, the general SLAM-idea of adding features to the state and using them for measurement purposes is still employed.

This work tries to establish whether such a SLAM-like approach is a feasible technique for a hazard-relative navigation filter.

In this context the two parts of this method need to be developed, implemented and tested: the hazard-detection (HD) method and the hazard-relative navigation (HRN) filter.

Testing of both parts is done first using a software simulator during software-in-the-loop testing (SILT). After the SILT of the HD method is successful, the hazard-relative navigation filter can be developed and also tested in the same software simulator.

The next and final step is to perform further testing of the algorithm using a more realistic set-up with real hardware outputs as an input, so-called hardware-in-the-loop testing (HILT). During this phase images are acquired in a scaled set-up at the Testbed for Robotic Optical Navigation (TRON) facility at DLR, Bremen, Germany.

The final aim is to answer the question whether vision-based hazard-detection and hazard-relative navigation are viable candidates for planetary landing missions. After performing both SILT and HILT it should be possible to reach a conclusion on the precision, the accuracy, as well as the robustness of the algorithm. In the process it will also be possible to investigate the robustness and accuracy of the hazard-detection function.

Summarising, the achievements with regards to the following sub-goals will be presented in the remainder of this work:

1. Development of a hazard-detection function;
2. Sensitivity study of the hazard-detection function;
3. Development of a hazard-relative navigation function;
4. Software-in-the-loop testing of the hazard-relative navigation function;
5. Set-up and execution of hardware-in-the-loop testing of both the hazard-detection function and the hazard-relative navigation function.

1.3. HISTORY AND STATE OF THE ART

To provide the context of this work, place it into the greater picture and create a common starting point, this section presents the current state of the art, as well as the (development) history of both hazard-detection and avoidance methods and terrain-relative navigation techniques, being the super-class of hazard-relative navigation methods.

1.3.1. HAZARD DETECTION AND AVOIDANCE

In 1989 the American President Bush announced the return to the Moon, as well as landing on Mars, with his Space Exploration Initiative (SEI)(Bush, 1989). Within this scope a Mars Rover Sample Return Mission (MRSM) was announced. As discussed earlier in this chapter the “big shock” during the Viking mission was that the “thought to be hazard free” landing site ended up being extremely unsafe, with the landers only surviving by luck. Clearly, these findings from the Viking missions, led to the conclusion that future landers should be able to avoid hazards and thereby decrease the risk of a landing failure.

One year prior, in 1988 Martin Marietta (now part of Lockheed Martin) reported for the first time on the development of an HDA system, set-up during an internally funded research and development project. During this project multiple vision-based approaches were investigated, based on both optical flow and more basic methods, such as intensity segmentation and edge detection. The optical-flow algorithm, however, does not generate dense maps, but sparse maps are used to average the elevation levels for

the remaining points. After the Bush announcement in 1989, Martin Marietta proposed this algorithm for the MRSM (Cuseo and Dallas, 1988).

After the publication of the results of Martin Marietta and the announcement of Bush, more research in this area was published. At first these methods were mainly vision based, however, in 1991 a sensor-trade study was performed on request of NASA (Tchoryk Jr et al., 1991). During this study lidar [4], and a combination of passive sensors and lidar, were found to be the most feasible options for HDA.

However, other research groups still continued to research vision-based systems for the SEI. Part of this research was conducted by a team of researchers at the Charles Stark Draper Laboratory (Pien, 1991a). They investigated a pure intensity, edge-based method, which tries to detect ellipses in the detected edges to find round features such as boulders, rocks and craters. Also they proposed to use these intensity images to preselect sites, which would be screened by a range sensor. Next to this, they also proposed a pure range-based detection. Their final algorithm made use of intensity images during the early descent phase and a laser range-image during the final descent phase (Pien, 1991b).

When in 1992 the "faster, better, cheaper" policy of NASA administrator Daniel Goldin was put into place, less focus was put on developing and using HDA strategies (Gross, 2001). Because of the low technology readiness level (TRL) of HDA it was not available for a fast implementation.

The focus on the new policy, and thus the design of missions that would be less complex and more robust, led to a pause in the development of HDA systems.

In 2001 Halbrook et al. (2001) correctly concluded that landers will always be either very robust, *i.e.*, by using airbags, or can aim to perform very precise landings. Hazard-tolerant landers cannot perform precise landings, *e.g.*, due to bouncing of the airbags. Therefore, if precise landings are desired, it is likely that HDA will be necessary, as these landers will be less robust to hazards. The term “precise landing” also encloses landing next to known hazards, for example, next to a crater rim.

With the start of the Mars Science Laboratory (MSL)-project (Grotzinger et al., 2012) in the 2000s, the NASA Mars Exploration Advanced Technologies Program (MEPAT), and the New Millennium Programs Space Technology 9 Project (ST9), HDA was put more into focus again, mainly camera-based systems were designed (Huertas et al., 2006; Huertas et al., 2007; Huertas et al., 2008; Huertas et al., 2010). These studies investigated multiple methods of camera-based HDA, for example, stereo methods and pure rock detection from texture. In 2006, the Autonomous Landing and Hazard Avoidance Technology (ALHAT) program was started, which lead to the development of a lidar-based HDA system. The ALHAT objective is to develop technologies necessary for a Lunar and planetary pinpoint landing. To this end three different lidar sensors were developed including a flash lidar Epp and Smith, 2007; Striepe et al., 2010, which is also used in the most recent version of the system. NASA implemented the ALHAT system on the Morpheus vertical takeoff and vertical landing test vehicle. In spring 2014 the full ALHAT capability was sucessfully demonstrated and tested on-board Morpheus (Epp et al., 2014).

Like NASA, also the European Space Agency (ESA), started to investigate HDA methods. This was first investigated under the Integrated Vision and Navigation (IVN) con-

[4] In this specific study they refer to lidar as “laser radar”

tract in 1999 (Strandmoe et al., 1999), and the Vision Based Relative Navigation Techniques Framework (VBRNAV) (Câmara et al., 2005). Various vision-based methods were analysed, Shape-from-Shading (SfS) at Deimos (Rogata et al., 2007; Câmara et al., 2005), and SfS and Stereo-from-Motion (SfM) techniques at Astrium (now Airbus Defence and Space) (Devouassoux et al., 2008). For the ESA Lunar Lander originally scheduled for launch in 2018, but unfortunately put on hold during the ESA Council Meeting at Ministerial level in 2012 and later cancelled, a lidar-based system was selected. Lidar was chosen over a camera, because the selected landing site would have been in the polar region, which is, due to its illumination conditions, a difficult region for an imaging sensor (De Rosa et al., 2011). From 2011 to 2014 a European consortium investigated and developed a more advanced flash lidar system, Flash Optical Sensor for Terrain Relative Robotic Navigation (FOSTERNAV), which is superior to the scanning lidar systems used during previous studies. A flash lidar can generate an instantaneous lidar scan in the same way a camera takes a picture (and is therefore sometimes called imaging lidar), contrary to a scanning lidar has to scan the area before a complete digital elevation model (DEM) is obtained (Pollini, 2012). Even though more developments were initiated after FORSTERNAV, there is still no qualified space flash lidar available on the European market, meaning that ESA missions would so far be constrained to the use of scanning lidars.[5] For the PILOT (Precise and Intelligent Landing using Onboard Technologies) Airbus Defence and Space, NGC Aerospace and Neptec UK are currently developing hazard detection and avoidance technologies for ESA. The developed system is scheduled to fly as a ESA contribution on the Roscosmos mission Lunar-Resource scheduled for 2020[6].

On 14 December 2013, the Chinese ChangE'3 lander successfully touched down on the Lunar surface (Ip et al., 2014). During its final descent it went into hover at approximately 100 m above the surface to perform HDA (Liu et al., 2014; Sun et al., 2013; Lakdawalla, 2014). According to Sun et al. (2013), the spacecraft made use of 3-D maps in combination with camera images. From the work of Xiong et al. (2013) on lidar exhaust-plume interaction it may be concluded that the 3-D maps were obtained using lidar. Still, five years after the successful landing, there is no dedicated publication describing the HDA system used. Nevertheless, it was the first Lunar, and extraterrestrial, lander to date performing HDA.

The previous paragraphs gave an overview of the main HDA projects in the past 25 years. Still, there are also smaller research projects conducted in the more recent years. Some of these will be mentioned in the following to give an impression of how many different approaches are chosen and projects are conducted. However, this section is not intended to be a complete list of every research ever conducted, but is more intended to show the greater picture.

For example, Crane and Rock (2012) investigated the possibility of predicting rock maps using an extended Kalman filter (EKF). Moreover, they showed that small changes in the descent trajectory can improve the HDA performance. However, their proposed HD algorithm is a shadow-based rock detection. Such an algorithm cannot detect slopes

[5] However, with the fast developing research field of autonomous driving, more and more COTS flash lidars are developed for the automotive industry. This development might eventually be a game changer for hazard detection as cheaper sensors are developed, also on the European market.

[6] http://www.airbus.com/content/dam/corporate-topics/publications/press-release/news-release-pilot_phaseb-en.pdf, visited: 20.05.2018

and will also not deliver a dense hazard map.

In the field of vision-based HDA, many algorithms have been developed. Most of these are very much comparable, and mainly serve the goal of various research organisations and companies to develop their own algorithm. These developments can roughly be divided into pure rock detection, crater detection, optical flow (also called stereo-from-motion) (Zhu et al., 2012), texture/intensity-based approaches (Howard et al., 2011; Yan et al., 2013), pure shadow-based (Cohanim et al., 2012), stereo vision (Huertas et al., 2007), and combinations of these. However, most of these developments are at a very conceptual level, testing the algorithm just for specific cases and on few datasets (or even just a single one). Many times, the development ended after a single study and no thorough follow on work was conducted.

The Lidar-Based Autonomous Planetary Landing System (LAPS) is a guidance, navigation and control (GNC) system developed by a consortium of Canadian space companies for the Canadian Space Agency (CSA). It is comparable to the system developed for ALHAT, although, it focuses on HDA (de Lafontaine et al., 2008; Langley et al., 2007). It was tested in a 1:1 scale test on-board a helicopter in 2010, where successful performance was demonstrated (Neveu et al., 2011a).

As mentioned previously, the HDA system developed for the ESA lunar lander is based on a scanning lidar, however, it also includes a camera-based system as a back-up. The camera-based back-up makes use of a SfS algorithm. The system is developed by Deimos (Parreira et al., 2013). The back-up camera system is based on the previous research at Deimos on Vision-based HDA (VBHDA) as reported by Rogata et al. (2007).

Like the ChangeE'3 lander, most other research projects made use of lidar-based systems. Performing HDA using lidar has different challenges compared to camera-based systems. Therefore, this research has a very different focus than these previous projects. Some research also focused on cameras as a hazard detection sensor. Here, either stereo vision or shape-from-shading is used for reconstructing the terrain. A project conducted under ST9 and MEPAT is the only reported project making use of a stereo-vision algorithm (Huertas et al., 2007). They found that including a hazard-detection algorithm would have decreased the risk of a landing failure by a factor of four, while the area accessible could have been increased by a factor of three for an MSL-like scenario. Their algorithm was designed for use at 100 m and below.

However, all previous research focused on the development of the respective algorithms and a proof of concept of these. In contrast, the current research does not only focus on a proof of concept, but also on determining the limitations of stereo vision to perform hazard detection on current and future planetary landers. To this end a thorough sensitivity analysis is performed to establish the limits of this kind of system. This approach is necessary to show that stereo-vision-based hazard detection can be used on future landers. Moreover, altitudes of 100 m and lower as reported by Huertas et al. (2007) are very low and it would be very desirable to increase the operational envelope to at least 200 m. Mapping at higher altitudes will be very beneficial for the system, because it increases the time the GNC system has to react.

The algorithm presented here makes use of the same basic principles as the ST9 HDA project, but as mentioned before the main focus of this research is put on an in-depth sensitivity analysis to determine the limitations of stereo vision for planetary landings

and to demonstrate that it is feasible at altitudes above 100 m as well. Moreover, ST9 used a single reference scene of rocks attached to a concrete wall, while this research makes use of both SILT on larger data-sets and HILT. In addition, clear trade-offs and motivations are presented for all algorithmic choices made. This highlights that the most optimal choices are made, but also increases the conceivability and reproducibility of the results.

1.3.2. TERRAIN-RELATIVE NAVIGATION

As in the field of terrain relative navigation(TRN) different terminologies are used for the same concepts, it is important to first define the terminology used in this work. Often, all techniques that use images or DEMs as inputs for localising the lander are referred to as TRN systems. However, there should be a clear distinction made between systems that use images to locate the lander with respect to an *a-priori* map (like Mars2020), and those that only do localisation of the lander with respect to what is currently observable from the spacecraft (the image used). The former is thus able to resolve the inertial, *absolute* position (and maybe orientation) of the lander, whereas the latter can only provide localisation *relative* to an image (but do limit the accumulation of additional relative error accumulated by the inertial measurement unit (IMU) over time). The former method will be referred to as terrain absolute navigation (TAN) in this work, while the latter method is called terrain relative navigation (TRN) as these naming conventions clearly stressed the main difference between these two methods[7]. It should be noted that there are important differences between TAN and TRN. TAN can be used to guide a spacecraft towards a predefined landing region, whereas TRN can be used to avoid hazards, which are identified in a hazard map on-board of a lander or even simply precision landing without the need of also avoiding hazards. The accuracy of TAN methods does depend on the resolution of the reference maps or catalogues used for matching, while TRN is not limited by any *a-priori* data. If very high resolution images of a surface exist, which are of sufficient resolution to perform hazard detection before the landing, TAN can be used to perform a safe precise landing, without requiring a TRN system. As such terrain maps are not available for most bodies, currently the only exception being Mars due to the very high resolution DEMs recorded by the Mars Reconnaissance Orbiter, a TAN system is not sufficient for safe landing, on all other bodies when unsafe landing regions may be present. As this work developed a TRN method, more precisely a hazard-relative navigation method, the full range of possibilities for TAN will not be discussed in this introduction.

The first mentioning of TRN was in the early 90s. Pien (1991b) summarised what was considered to be necessary for the autonomous exploration of Mars. In this context he did not only mention HDA, but also autonomous navigation and precision-landing strategies. Based on what Pien describes, this is synonymous to TAN/TRN. Also a study from 1991 exists outlining simple TAN/TRN systems (Vaughan et al., 1991). Prior to its appearance in the context of planetary-landing application terrain-aided navigation was already studied and implemented for missile systems.

[7] Note that within the context of the ESA Lunar Lander the two systems were sometimes referred to as terrain relative relative navigation (still abbreviated as TRN) and terrain absolute relative navigation (also abbreviated as TRN)

Like HDA, also TRN has already been studied for quite a while. So far, only simple implementations of a TRN system have already been flown on an actual mission. The Descent Image Motion Estimation System (DIMES) flew on the two Mars Exploration Rovers mission in 2003 (Crisp et al., 2003). DIMES can be considered the first, yet simple, TRN system to fly on a mission and is to date the only such algorithm that was ever used on a spacecraft. Contrary to the TRN method presented in this work, DIMES did not operate on the full spacecraft state, but was only able to measure the horizontal velocity based on two camera images. This arose from the need of knowing the horizontal speed to counteract for potential steady-state winds, which were known to be a potential hazard to the airbag landing-system. The full story of DIMES, its development and its performance during the Mars Exploration Rovers (MER) landings is discussed by Cheng et al. (2004).

Both the aforementioned NASA ALHAT project and the ESA Lunar Lander developed TAN/TRN solutions next to their HDA efforts. On board of the Morpheus test vehicle, the ALHAT team was able to demonstrate that their HDA in combination with their TRN solution is able to perform a safe landing in the presence of landing hazards (Epp and Smith, 2007). The ALHAT project made use of a lidar as a TRN sensor.

Also Neveu et al. (2011b) presents a lidar-based TRN solution which makes use of a similar matching principle that star-trackers use. A lidar-based TRN solution might become necessary when lighting conditions prohibit the use of camera-based techniques. However, the necessary additional overhead due to motion compensation problems linked to the fact that a lidar-scan (to date) still takes significant longer to acquire than a camera image, should not be under estimated.

For the ESA Lunar Lander project not only a TRN, but also a TAN system was developed. Testing of these algorithms unfortunately never left the laboratory hardware-in-the-loop test stage. Some of the result obtained from these project can be found in (Hamel et al., 2006; Simard Bilodeau et al., 2012), which describe the developments with respect to TAN based on crater matching. However, the PILOT system developed by NGC Aerospace and Neptec UK, is currently undergoing elaborated testing. As mentioned before this system is large building upon on the results of the LAPS project (Neveu et al., 2011a). Unfortunately there were little dedicated publications on this topic to date, intermediate results of the testing of the HDA functionality are presented in (Hamel et al., 2018).

The Mars2020 lander, an MSL successor, will most likely be the next mission to fly a TRN system. Mars2020 will be built using MSL heritage, but will differ from MSL by landing in an non-inherently safe landing region. Since high resolution surface images of Mars exist today, hazards will be classified pre-mission and the TRN will navigate based on these *a-priori* maps. Currently a lot of research is done in the context of Mars2020 and its TRN approach, but this introduction does not aim at giving a summary of this work. However, it is important to realise that the Mars2020 approach is fundamentally different from the method proposed in this work, since this algorithm is capable of aiding precision landing in unknown terrain, which the Mars2020 approach is not. In the previously employed terminology the approach followed by NASA is a TAN method.

The different possibilities for TRN/TAN studied in detail until 2008 are discussed in the paper by Johnson and Montgomery (2008). This paper is recommended as an entry

point into the field of TRN, note that due to its publication date, more than ten years ago, it is not up-to-date any more and does not cover the full range of algorithms, which exist today.

After 2008 and next to the larger projects already mentioned in this section, there were some other developments in the field of TRN, however, most of these developments focus on TAN-based methods, for example, with landmarks-to-map matching (*e.g.*, (Delaune et al., 2016)) or crater matching (*e.g.*, (Maass et al., 2011)). Since this work focuses on feature-to-feature matching and a SLAM-like implementation of this, a thorough analysis of the conducted TAN work is skipped. However, Mourikis et al. (2009) present a combined TAN and TRN approach, where feature-to-feature matching is performed in case feature-to-map matching is not possible, for example, because of large differences in resolution. They were able to show that as a backup the TRN method was able to limit the error growth by estimating the velocities from the extracted features.

Also Bilodeau et al. (2014) combine a TRN and TAN method into an full landing navigation system, to achieve final landing accuracies of less than 100 m. They use image-to-image feature tracking to measure the angular rates as well as the velocity direction during the descent, while the absolute navigation matches features to a database.

However, to date no system combining stereo-vision based hazard-detection and terrain-relative navigation was presented for a planetary lander. In this work an attempt to exploit the close link of these two systems is presented, resulting in a hazard-relative navigation algorithm. Most developments to date aim at absolute localisation and not limiting the error growth with matching real-time extracted features. In the absence of an *a-priori* surface map, map-to-feature mapping will not be possible. Moreover, SLAM-based techniques were not implemented either.

Bilodeau et al. (2012) present a rover navigation system which makes use of a similar to the approach, using a stereo-based mapping system in combination with feature extraction and tracking. They were able to achieve less than 1% position estimation error at the end of a more than 200 m traverse. These findings demonstrated that stereo measurements can be a feasible data source for relative navigation. It should be noted that performing stereo measurements on a rover has very different challenges than on a landing vehicle since the cameras are a lot closer to the surface.

Figure 1.1 shows a landing scenario involving all aforementioned systems: hazard detection, hazard-relative navigation, terrain-relative and terrain-absolute navigation. It can be seen that TAN is performed at high altitudes, while TRN is only used at lower altitudes. Once the landing region is in the lander's field-of-view HDA and HRN are switched on, enabling a precise hazard relative landing.

1.4. CONTRIBUTION OF THIS WORK

Based on the foregoing discussion, it was shown that this work is one amongst the few attempts to perform hazard detection based on stereo images. It is the first to perform a detailed software-in-the-loop, as well as a hardware-in-the-loop analysis. The outcome is a method that is thoroughly tested and may be a viable candidate for future landing missions. Moreover, the operational envelope is expanded by a factor of two as compared to previous work.

Almost all image-based navigation approaches to date try to match features extracted

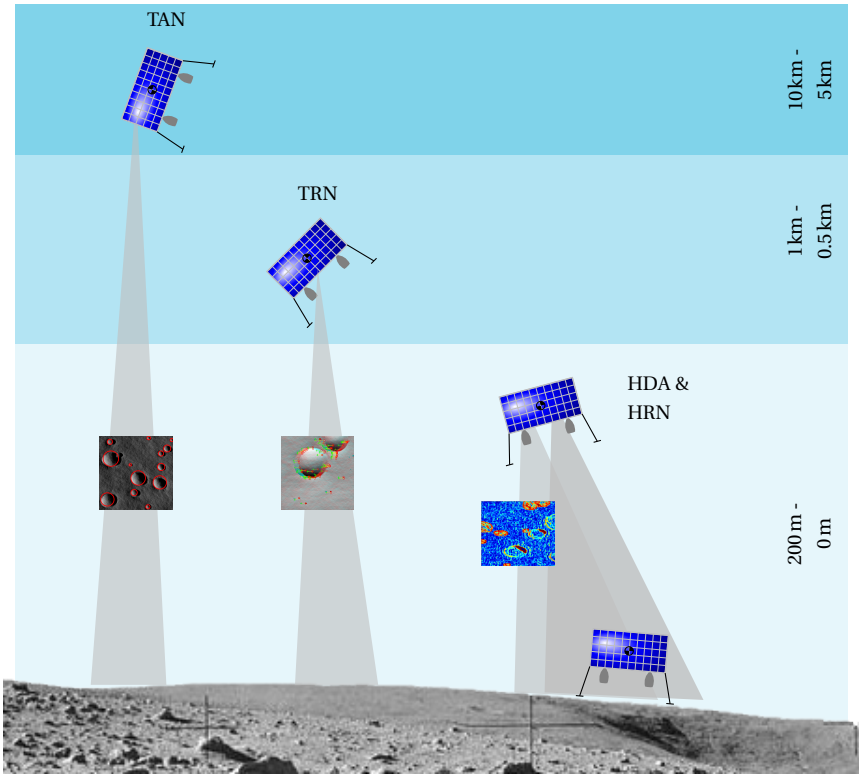


Figure 1.1: Landing scenario with advanced GNC system.

from the images to an *a-priori* map, which enables absolute localisation of the vehicle. However, with the hazard map being measured from the vehicle, it is not necessary to have precise, absolute (inertial) state knowledge to successfully avoid the hazards. Even more, linking the hazard maps to some inertial frame via absolute state information obtained from TAN will potentially only introduce new errors. Avoiding this intermediate step by directly navigating with respect to the hazard-detection maps will overcome these issues.

This work will link the hazard-detection outputs to the navigation filter in such a way that by default localisation will be performed relative to the detected hazards. This avoids problems of linking the hazard maps to the navigation outcome.

In terrain-relative navigation, it is usually common to derive a measurement from the extracted features, while including the features in the state is usually not done. Here, the idea is to establish whether a SLAM-based approach is feasible for hazard-relative navigation.

In conclusion, this work will be able to advise on the applicability of the developed approach for future methods, as well as giving an outlook on further research to extend and improve the proposed system.

1.5. THESIS OUTLINE

This work has three main parts: 1) the development and testing of the hazard-detection algorithm, 2) the development of the hazard-relative navigation algorithm, and 3) the performance analysis of the hazard-relative navigation solution in combination with the hazard detection in a hardware-in-the-loop environment.

Chapter 2 starts by introducing the concept of hazards and hazard detection in Sec. 2.1. In Sec. 2.2 the development framework for the hazard-detection method is described. Section 2.3 presents the method to construct hazard maps from surface DEMs. Following this, an assessment of three different camera-based hazard-detection techniques is presented in Sec. 2.4. Stereo vision, the selected method, is summarised in Sec 2.5, as well as a reference scenario and performance requirements are presented in the same section. The chapter continues in Sec. 2.6 with a thorough sensitivity study of the designed method, which is then used to determine the application envelope of the stereo-vision hazard-detection function. The next step necessary for including hazard detection into future missions and remaining areas of work are presented in 2.7. A brief summary of the findings of Chapter 2 is given in the final section, Sec. 2.8.

The next chapter, Chapter 3, starts by introducing the principle of state estimation and Kalman filters in Sec. 3.1 and hazard-relative navigation in Sec. 3.2. After this a brief introduction to the principles of simultaneous localisation and mapping is given in Sec. 3.3. Section 3.4 outlines the filter set-up, as well as the image and surface DEM-based measurements used. Before the filter can be tested, it is necessary to tune the filter (Sec. 3.5). The next step is to perform testing of the proposed method, and to do so the reference scenario used for this testing is presented in Sec. 3.6. The results and set-up of the software-in-the-loop testing is presented in Sec. 3.7. The chapter closes with a brief summary of the findings in Sec. 3.8.

Chapter 4 outlines the hardware-in-the loop testing of both the HD function and the HRN method. First, the TRON facility is described in Sec. 4.1. The set-up of the HILT is presented next in Sec. 4.2. After that, the HILT results of the HD function and the HR are presented in Secs. 4.3 and 4.4, respectively. A summary of the HILT test is given in Sec. 4.5.

This thesis closes with the final conclusions, recommendations and an outlook for future work in Chapter 5.

1

2

HAZARD DETECTION

ENABLING a vehicle to land autonomously on a planetary body or moon that might contain substantial landing hazards on its surface, will be a game changer for future exploration missions. Not only will this reduce the risk of landing missions, it will also greatly increase the number of potential targets, both on currently unvisited bodies, but also on the Moon and on Mars. The ability to autonomously detect surface hazards on-board and in (near) real time is a necessary asset for this endeavour. This chapter presents the development of such a hazard-detection method.

After introducing the basic definitions and concepts of landing hazards and hazard detection in Sec. 2.1, as well as the development framework of such methods in Sec. 2.2, the method to constuct hazard maps from surface DEMs is presented in Sec. 2.3. Three different camera-based hazard-detection methods are introduced and compared in Sec. 2.4, namely stereo vision, stereo-from-motion, and shape-from-shading. From this assessment, stereo vision was found to be the most suitable candidate method for camera-based hazard-detection. The resulting algorithm is presented and evaluated in Sec. 2.5. Section 2.6 presents a thorough analysis of the developed method, proving its robustness and determining its performance envelope. The chapter continues with some general remarks on current hazard-detection developments (Sec. 2.7), which also aims to serve as a bridge to the next chapter. A brief summary of the findings presented in this chapter is given in Sec. 2.8.

2.1. BASIC PRINCIPLES OF HAZARD DETECTION

Before diving into the development of a hazard-detection approach, it is useful to discuss the basic principles of hazard detection and avoidance. To do so, first the concept of landing hazards should be established. There are surface features and conditions that can lead to a failure, so-called landing hazards. The main hazards are:

Parts of this chapter have been published in Acta Astronautica (2016) (Woicke and Mooij, 2016a) and in the Proceedings of the AIAA Guidance, Navigation and Control Conference 2016 (Woicke and Mooij, 2016b).

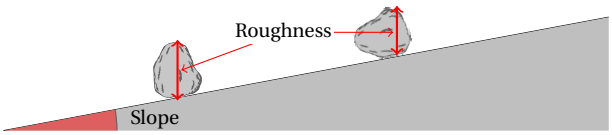
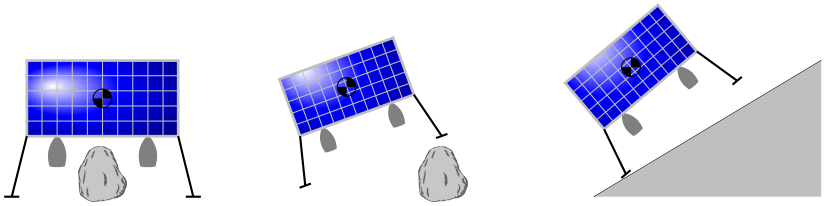


Figure 2.1: Roughness as opposed to slope.



a “Sitting” on rock b Tipping c Sliding

Figure 2.2: Hazard failure modes.

- Too large roughness
- Too large slope
- Too large craters
- Insufficient illumination

In the following, these different hazard are discussed in more detail.

Roughness Roughness describes the features that deviate from the mean plane. Simply speaking, roughness is created by small features lying on the surface, such as rocks. The difference between slope and roughness is depicted in Figure 2.1. Larger rocks are often referred to as boulders. Therefore, roughness is mainly a measure for rocks and boulders lying on the surface. How much roughness a lander can handle, depends on its design. The ground clearance, a measure that indicates how much space there is in between the vehicle’s base plate and the ground, determines the maximum allowable size for roughness features. The associated failure modes are shown in Figure 2.2a and b.

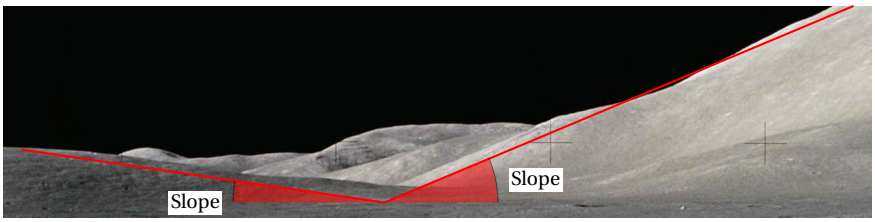


Figure 2.3: Slope, note: this scene contains more slopes than those two indicated.

Slope The slope between two points describes how steep the terrain is in between these two points. Likely, the slope is not constant over an entire landing region, but can vary. Usually, the slope is computed by fitting a mean-plane through multiple neighbouring points, as this will exclude smaller outliers, such as rocks, from the slope evaluation. Figure 2.3 shows an image of the Lunar surface. Multiple different slopes are present in this scene, two of which are indicated. If the local slope is too large it can pose a threat to the lander, as the lander can either slip away or, even worse, tip over, as shown in Figure 2.2b and c. If and how extreme slipping might occur, depends on the composition of the ground, as well as the mass and velocity of the lander. The stability with respect to tipping over is based on the location of the lander's centre of mass.

Craters Craters do not necessarily pose a hazard. Whether or not this is the case, strongly depends on the mission itself, but also on the age and size of the crater. On one hand, for a rover mission it might be very unfavourable to land inside a (smaller) crater, as the rover would be confined to the inside of this crater. On the other hand, younger craters might be attractive from a geological point of view and could be a very interesting landing site. Moreover, crater rims are often very rough and also ejecta from the impact that form the crater can cause a lot of roughness hazards to be present in the vicinity of craters.

Illumination As for craters, the hazardousness of certain illumination conditions depends on the mission itself. If a mission will receive its entire energy from solar power, it is of utmost importance that the landing region is illuminated. The same goes for landing systems that rely on visual imagery for certain descent systems, such as terrain-relative navigation, which will be discussed later. Last but not least, it is also important to consider that public outreach is important. If the landing site is in total darkness, no descent images could be broadcasted. However, (continuously) shaded areas might be very interesting in the search for life as they are extremely cold and might therefore have preserved organic material.

How much slope poses a hazard or how large a boulder can be depends on the lander design and differs for every mission. Slope and roughness hazards both depend on the centre of mass location, while roughness hazards are also linked to the ground clearance of a lander. For a nuclear-powered lander with a landing system that does not require visual images, landing in total darkness might be allowable. Several reference values of the slope and roughness tolerances of past missions are given in Table 2.1. From this table it can be concluded that there is no clear trend concerning roughness hazards nor slope. Both values do not simply decrease or increase over time, rather these values are linked to the robustness of the landing system. The MER rovers, for example, landed inside an airbag system, which strongly increased the robustness of the landing.

Currently, it is common practice to select landing sites that are (thought to be) free of any hazardous slopes, roughness, craters and unfavourable illumination conditions (*i.e.*, shadows). These landing sites are selected prior to a landing, off-line by human analysts using DEMs and images obtained by orbiters and previous missions. These landing sites are called "inherently safe" landing sites. To select these landing sites, either very high

Table 2.1: Roughness and slope requirements of past missions.

Mission	Date	Max slope	Max roughness	Source
Viking (NASA)	1976	≤ 15°	20 cm	(Braun and Manning, 2006)
MER (NASA)	2004	≤ 30°	50 cm	(Braun and Manning, 2006)
MSL (NASA)	2012	≤ 30°	55 cm	(Golombek et al., 2012)
ChangE'3 (CNSA)	2013	≤ 8°	20 cm	(Jiang et al., 2016)
Philae (ESA)	2014	≤ 15°	-	(Ulamec and Biele, 2010)
ESA LL (ESA)	cancelled	≤ 15°	50 cm	(De Rosa et al., 2012)

resolution orbital images are required or assumptions on rock and boulder distributions have to be made, based on low-resolution images. However, requiring *a-priori* orbiter data makes landing missions expensive and delays the schedule. But, there are bodies where orbiting is difficult *e.g.*, due to high radiation. Moreover, selecting only safe landing regions as candidate landing sites constrains the choice of candidate landing sites. Since there are already plenty of other constraints that limit the choice, such as visibility of a communication relay (to transfer science data to Earth) and terrain elevation of the site (to achieve sufficient deceleration before touch-down), to name only a few, it is greatly desired to avoid adding even more constraints due to hazards. In retrospect, this also means that using active hazard detection and avoidance, a larger portion of a body becomes feasible to land in (Huertas et al., 2008).

To date, it is only possible to obtain images of sufficiently high resolution to select inherently safe landing sites for the Moon and Mars, but even for these bodies no full high resolution DEMs exist for the entire body. Images might be taken on request, however, it depends on the satellites' orbits how fast and when these images are available.

Many lander missions have been performed in the last decades, with plenty being successful. This raises the question why there should be a need for a system capable of detecting and avoiding surface hazards, if until now landing was possible without? Here, one has to consider how landing sites were selected for past missions: they were selected such that landing ellipses would not contain any known hazards. This requires two things: 1) that digital elevation maps (DEMs) of the landing region are available during the mission planning, and 2) that these maps are of sufficient resolution to detect all surface features that pose possible landing hazards. If either one is missing, one has to make assumptions and use mathematical models.

Apart from Mars and the Moon, no other body fulfils the first requirement. Thus, for bodies such as Venus, Mercury, or moons such as Europa, it is impossible to preselect a safe landing region based on pre-mission DEM analysis. Therefore, lander missions to each of these bodies might require HDA to ensure a safe landing. But how is the situation for the Moon and Mars? As DEMs are available it is possible to select a landing site, which is free of hazards, so-called "inherently safe". To analyse whether the DEMs of Mars and the Moon are of sufficient resolution, the best available DEMs for these two bodies have to be analysed.

The most recent complete Lunar DEM was generated based on Lunar Reconnais-

sance Orbiter (LRO) data. This mission provided the highest resolution global lidar DEM currently available. It mapped the entire Lunar surface at a resolution of approximately 30 m/pixel (Smith et al., 2010). Compared to previous mapping missions, such as Clementine with 8 km/pixel to 30 km/pixel (Smith et al., 1997) and the Kaguya Laser Altimeter with approximately 2 km/pixel (Araki et al., 2009), this is a significant improvement. Next to the lidar, LRO also carries a camera on-board, the Lunar Reconnaissance Orbiter Camera (LROC), capable of providing images of resolutions of approximately 100 m/pixel with global coverage. Both the lidar and the camera can deliver better resolutions for the polar regions. In these regions, this enables the detection of boulders and rocks of 1 m horizontal scale with heights larger than or equal to 0.5 m. The minimum size for detecting a crater would be 2.5 m (Robinson et al., 2010).

The Mars Reconnaissance Orbiter used the High Resolution Imaging Science Experiment (HiRISE) instrument to reconstruct the DEM using a stereo approach. This instrument has a vertical resolution of 0.3 m/pixel (Kirk et al., 2008). This results in a DEM of 1 m in horizontal scale. It delivers a vertical precision of approximately 0.25 m (McEwen et al., 2007). However, this means that only rocks higher than 0.5 m can be detected. Therefore rocks and craters of less than 2 m diameter and 0.5 m vertical size cannot be detected using MRO DEMs.

Concluding, both for Mars and the Moon, all features smaller than 0.5 m in vertical size as well as 1 m to 2.5 m horizontal size will remain undetected. These features can still pose substantial landing hazards. Entire fields of small rocks, craters, and boulders might be missed. Landing with one leg inside a crater, which is smaller than 2 m might still cause the lander to tip over. The same applies to landing on a rock, which is only 1 m in diameter, but also 1 m high. However, both hazards would not be detectable from recent high-resolution orbiter DEMs. Therefore, it is insufficient to fully rely on orbital DEMs to determine the absence of any hazard in a landing region. This also shows that regions that appear to be entirely flat from orbit may actually contain very small hazards, which could still lead to landing failures. Still, it can be argued that if no large hazards are found in an area, it is likely that there are also no very small hazards. But, this will not provide a 100 percent reliability. Moreover, Viking proved that this is a dangerous assumption. Especially those Lunar regions, which were not mapped to the high resolutions as for the poles, may contain plenty of hazards, unidentifiable from the LRO DEMs.

Investigating two current lander/rover designs one can find the following: the current design of the ESA Lunar Lander (LL) has a footprint diameter of 5.6 m and a body diameter of 2.56 m (Carpenter et al., 2012). The NASA MSL rover is reported to have body dimensions of 3 m × 2.7 m × 2.2 m (Way et al., 2007). A 2.5 m large crater or boulder would almost be equal to the size of MSL and be half the size of the LL body. Clearly, this kind of hazards will pose a substantial risk for the success of a mission. Therefore, orbiter maps are insufficient for guaranteeing the safety of a mission. This underlines the importance of developing HDA systems.

For bodies mapped at insufficient resolutions, it is possible to estimate boulder distribution based on models. However, this is not necessarily a safe approach: Figure 2.4 shows the boulder “Big Joe”, a 2 m large boulder found right next to the Viking-I lander. Back then, no high-resolution maps were available of the Martian surface and based on

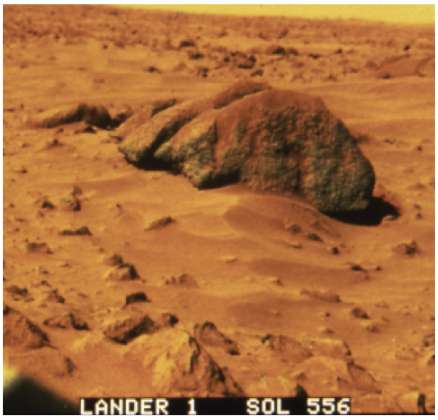


Figure 2.4: Boulder "Big Joe". A 2 m boulder right next to the Viking lander. Source: NASA

coarse images and models it was estimated that the Viking landing site would be free of hazards. Inspecting Figure 2.4 carefully, one can conclude, that "Big Joe" was not even the only boulder present, but the lander landed inside a boulder field. Post mission it was therefore calculated that the actual risk of a landing failure was a lot higher than designed for.

It is important to note that this kind of mistakes will not happen any more for Mars landing sites, as the resolution of the available images and DEMs greatly improved over the last 40 years, since the Viking landings. However, the scientific community is turning towards the exploration of bodies, which are today as insufficiently mapped as Mars in the late 1970s. To name a few examples, one could think of Jupiter's moons, *e.g.*, Europa, Saturn's moon Titan, comets and asteroids, but also bodies with an optically impenetrable atmosphere, like Venus. In 2015 a NASA Venus study team called hazard detection and avoidance one of the necessary technologies for the future exploration of Venus (Group, 2014). Also the NASA Dragonfly concept proposed for the New Frontiers Program in 2017 will require an HDA system for landing (and flying) on Titan (McGee et al., 2018).

Lastly, even for bodies where sufficiently accurate surface models exist, there is never a guarantee that the surface does stay the same in between mapping and landing. Surface change might be rare on other planets, but they do happen. Figure 2.5 shows a boulder (red circle) that traversed a large distance on Mars, shortly before the image was acquired. The boulder would have been large enough to pose a possible landing hazard.

During the Apollo-era, hazard detection and avoidance was performed by the pilots. Brady and Paschall (2010) explain how nevertheless every mission almost failed. Based on this it is questionable whether humans are a reliable sensor for real-time hazard detection. This implies that future human missions to the Moon or even Mars might better be equipped with computers for hazard detection, rather than giving this tasks to the human crew.

Concluding, it can be said that there are various reasons why on-board real-time hazard identification and avoidance is necessary. Especially the next generation of missions,

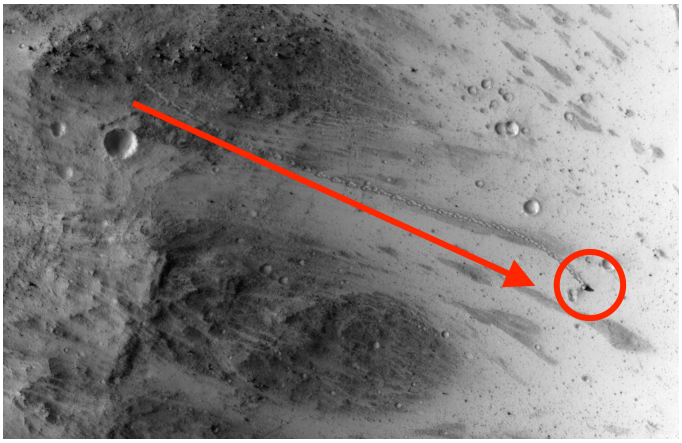


Figure 2.5: Surface change on Mars, an image taken by MRO. Source: NASA/JPL-Caltech/Univ. of Arizona

targeting less explored bodies or more complex terrain to yield more scientific output, will require the capabilities to create and evaluate hazard maps on-line.

HDA can be split into two main tasks, hazard detection and hazard avoidance. HD covers the task of obtaining the map of the landing region and processing it into a so-called hazard map, indicating safe and unsafe regions. HA has the task of assessing theses maps in combination with, for example, propellant availability and reachability information to decide if a diversion from the nominal landing site is necessary or even possible, and, if so, to select a new landing site. This thesis focuses on the task of hazard detection, as well as on the task of more precise localisation (Chapter 3), which is a prerequisite for performing hazard avoidance. However, hazard avoidance in itself, with a piloting function and guidance laws capable of re-targeting, will not be addressed.

As was mentioned before, hazards can be identified using camera images or DEMs. This is no different for a real-time system as compared to the way landing sites are selected off-line. When using images directly, patterns and image textures can be used to identify specific features, such as boulders or craters, or roughness in general. For a more robust method it is always preferable to detect hazards, not only based on images, but also based on DEMs. Dense DEMs can be obtained either directly from active sensors, *i.e.*, lidar, or reconstructed from passive sensor measurements, *i.e.*, images. Due to the budgetary advantages of cameras over (space-qualified) lidars, in terms of mass, power and cost[1], cameras are a very interesting HD sensor for low-cost and small missions, but also as dissimilar redundancy for a lidar-based system. As mentioned previously, DEMs are not directly "measured", but rather computed when using passive sensors. Therefore, passive HD poses a different challenge compared to lidar-based HD. Passive sensors will not work in the absence of a light source, therefore landing in (constantly) shaded areas

[1] For example: DragonEye 3D flash lidar space camera™ Advanced Scientific Concepts, Inc.: dimensions 11.2 cm × 13.2 cm × 11.9 cm, mass 3 kg, and a power consumption 35 W http://www.advancedscientificconcepts.com/products/dragoneye.html, visited 24.04.2018. The EACM space imaging system of Malin Space Science Systems: dimensions 7.8 cm × 5.8 cm × 4.4 cm, mass 256 g, power consumption 2.5 Watt http://www.msss.com/brochures/ecam.pdf, visited 24.04.2018

Table 2.2: Sensor options overview.

Sensor	Methods	Output	Reference
Scanning lidar		Full DEM	(Jiang et al., 2016)
		Full DEM	(de Lafontaine et al., 2008)
Flash lidar		Full DEM	(Epp and Smith, 2007)
Radar			
Single camera	Shadow detection	Shadow map	(Cohanim et al., 2012)
	Texture detection	Texture map	(Howard et al., 2011)
		Texture map	(Yan et al., 2013)
	Shape-from-shading	Full DEM	(Rogata et al., 2007)
		Full DEM	(Câmara et al., 2005)
		Full DEM	(Parreira et al., 2013)
		Full DEM	(Devouassoux et al., 2008)
	Shape from motion	Full DEM	(Devouassoux et al., 2008)
	Rock detection	Rock map	(Huertas et al., 2006)
Multiple cameras	Stereo vision	Full DEM	(Woicke and Mooij, 2014)
		Full DEM	(Huertas et al., 2008)

would require an active sensor. Especially at the Lunar south pole where regions, which are always in darkness, exist. Such a region was chosen as the landing target for the ESA Lunar Lander (Carpenter et al., 2012). It is frequently wrongly assumed that lidar DEMs are obtained faster than camera-based DEMs, as these need to be reconstructed, while lidar DEMs are measured. This is, however, not the case as the sampling process is slower with a lidar, because it has to scan the ground and cannot take an instantaneous image, and the raw data needs post processing before an actual DEM is constructed. Therefore, lidar-based HDA is not by default faster than passive HDA methods. Table 2.2 lists different sensor options, and algorithm options for camera-based methods. This is supposed to show that different methods and sensor options are studied and tested. It leads to the conclusion that so far no method is ruled out and neither was a “best” solution found.

A choice for a certain system will always depend on the selected mission scenario, mission requirements and constraints. Camera-based systems may present a very interesting low-budget candidate and may therefore also serve as backup option to active HDA technologies. Due to these appealing characteristics of camera HDA, the remainder of this chapter, and this thesis in general, will deal with camera-based hazard-detection techniques.

Derived from the previous discussion of surface hazards and the requirements formulated for related research, *i.e.*, (Epp and Smith, 2007), (Langley et al., 2007) and (De Rosa et al., 2011), the following requirements were set up for designing a hazard mapping algorithm:

1. Slopes greater than 15° shall be detected;

2. Roughness, *i.e.*, perturbations from the mean surface, larger than 0.5 m shall be detected;

3. The percentage of wrong detections, *i.e.*., undetected hazards, shall be less than 1%,;
4. The algorithm shall be executable in less than 2 s on a 2.3 GHz Intel Core i7 ivy bridge processor with 16 GB RAM, to ensure it is fast enough on a flight processor.

These requirements clearly only define the maximum number of undetected hazards, but do not define the number of safe sites labelled as hazardous (false alarm). However, to use the map for hazard avoidance, it will still be suitable, even if only a single safe site would be detected. Therefore, it is not necessary to pose a requirement on the number of false alarms. Of course, this number should be minimised, if possible. Therefore, the number of total correct detections, all correctly identified safe and hazardous sites, is analysed.

2.2. HAZARD DETECTION DEVELOPMENT FRAMEWORK

Before elaborating on the selection of a suitable camera-based hazard-detection algorithm and then detailing this method, first the framework used for this trade-off and the further development is presented.

To accurately test how well the algorithms reconstruct the DEMs, it is necessary to know the true DEM of the scene imaged by the stereo set-up. Based on this DEM, it is further possible to compute the ground-truth slope and roughness, and from the combination of these, the true hazard map.

The ground-truth DEM can be obtained in multiple manners. It is possible to use a hardware set-up of stereo cameras that image a scene, which was measured using lidar or an equivalent device, so that the ground-truth DEM is known. This option has the drawback that it is laborious to link image pixels to DEM coordinates. Moreover, if alterations of the terrains are required this will involve manual labour. Also, realistically modelling a reference scene is a challenge. Recalling that a sensitivity analysis has to be conducted, which will call for different terrains, this may be impractical.

A second option is to simulate an artificial camera, which takes images of a known DEM using a software simulation. It seems to be practical for testing purposes, if not only real DEMs, but also artificial DEMs can be used as an input for generating these images. Here, it is very important that the models for terrain, craters, and boulders are realistic. For this, the PANGU planet surface simulation software is a suitable tool (Parkes et al., 2004). It can simulate camera measurements of pre-loaded and artificial DEMs. The artificial DEMs are based on a very extensive model, which can create surfaces equal to those encountered on Mars and the Moon. Especially the possibility of adding boulders, to both real and artificial DEMs, is very beneficial for this research. It should be noted, that currently there are no Lunar or Martian DEMs available that have sufficient resolution for generating input for this research. Simply upsampling these DEMs to the desired resolution is not possible, as this will lead to problems when analysing the algorithm's performance.

The trade-off, as well as the prototype development is performed using artificial Lunar-analogue images generated in PANGU. Using real imagery and thus tackling the additional challenges that come with the use of real, noisy and erroneous data is not suitable for such an early stage of a development process. If the input images would contain

either noise or distortion, this would require additional preprocessing before reconstruction of the DEM, *i.e.*, Gaussian smoothing or undistortion of the input images. Moreover, these image errors would make it more complex to assess the algorithm's performance since errors in the resulting maps may result from algorithmic errors, but could also be caused by image errors. Therefore, real images will only be used during HILT as presented in Chapter 4.

2.3. CONSTRUCTION OF HAZARD MAPS

To assess the safety of a landing region hazard maps are derived from the DEMs computed by the hazard detection function. Computing these hazard maps from the DEMs is independent of the source of the DEMs. Therefore, hazard map construction will be discussed first, and only after that the different methods to generate the input DEMs will be introduced and analysed. A hazard map is assembled from slope, roughness, texture and illumination maps. The following sections discuss how to generate these individual maps and how to merge them into a final hazard map.

2.3.1. SLOPE ESTIMATION

Using a DEM as an input, it is possible to compute the slope. Most methods to compute the slope make use of a plane fit, but also other methods are available, such as steepest-descent and finite-difference methods. An analysis investigating execution time and slope-reconstruction accuracy was performed for six slope-computation methods: steepest descent, third-order finite-differences, quadratic surfaces, partial-quadratic equation, linear regression and intelligent plane-fitting, *e.g.*, (Zhang et al., 1999) and (Black and Sapiro, 1999). For intelligent plane-fitting, two different methods were implemented, one making use of a median plane, while the other uses a mean plane. In general, median planes are more robust with respect to outliers than a mean plane. However, this can also cause a loss of detail.

The analysis was performed for seven different DEMs of known slopes, some of which contained either random noise or boulders. It was then investigated how fast and how precise the different algorithms are able to compute the slope. It was found that finite differences, partial-quadratic equation, and steepest descent produce very large slope errors, overestimating the slope by more than 200 %, in the presence of noise or boulders opposed to noise-free DEMs. Therefore these three methods are not feasible for the proposed algorithms, because the input DEMs will contain boulders and noise. Out of the remaining methods, linear regression and the median-based intelligent mean-planes gave the lowest errors. The error of the median-mean-plane is approximately 10 % less than for linear regression, resulting in an absolute difference of 1°. However, linear regression can be executed four times as fast as the "intelligent" algorithm. The quadratic-surfaces algorithm consistently results in slightly larger errors than linear regression and is twice as slow.

Based on this analysis a linear-regression mean-plane was selected to be the best option for the proposed algorithm. The intelligent mean-plane algorithm estimates the magnitude of the slope slightly better, but the execution time is considerably larger. The difference between an intelligent mean-plane and a "normal" mean plane (thus the

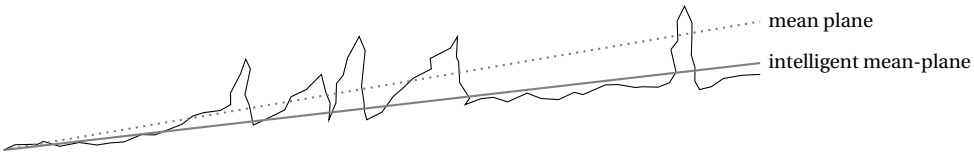
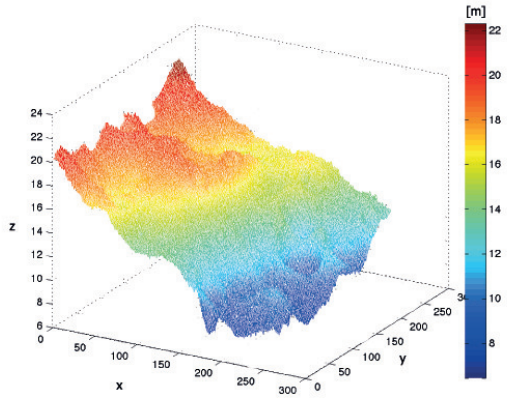
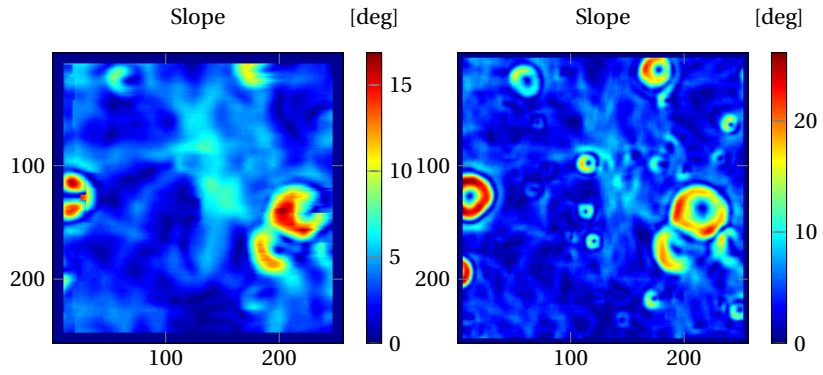


Figure 2.6: Plane fitting and roubust/intelligent plane-fitting.



(a) DEM.



(b) Intelligent mean plane (c) Mean plane

Figure 2.7: Intelligent mean-plane and mean plane compared.

linear-regression mean-plane) is the following: the former tries to filter out roughness features and errors before computing the slope. This is done by performing two plane fits. Based on the first one outliers are eliminated, which are excluded for the second, final fit. This difference in the resulting mean planes is shown in Fig. 2.6. Figure 2.7

shows a comparison between the normal mean plane and the intelligent mean plane, computed for the DEM shown in Fig. 2.7a. The DEM contains a slope from the left to the right, with some smaller boulders and craters on the terrain. The maximum altitude is in the range of 20 m to 24 m and the minimum altitude is 6 m to 12 m. The input DEM was distorted with Gaussian white noise. Performing a trade-off between these two methods, it can be seen that the intelligent mean plane leads to slightly lower slopes, while it filters out most of the smaller craters and sections of the larger craters as well. Therefore, it is concluded that the intelligent mean plane removes too many smaller details. Moreover, the 1° gain in slope accuracy does not justify an increase in the execution time by a factor of four.

It should be mentioned that related projects did frequently make use of an intelligent mean plane, *e.g.*, (Langley et al., 2007). However, this analysis clearly shows that the resulting slope and roughness maps are hardly improved. It is stressed, that for lidar-based hazard-detection systems an intelligent mean-plane is necessary to remove measurement errors from the lidar DEM before computing the slope.

It was concluded that linear regression should be used to compute the slope of the terrain. To do this a set of points from the DEM will be used to fit a plane trough them. A square window is used to determine the sub set of pixels used for this computation. The window size is determined by the footprint of the lander. In this research a 3 m footprint was assumed. Since the resolution increases with decreasing altitude, the pixel window size will increase as well, since the pixel resolution will get smaller at lower altitudes. The slope of the resulting plane is then the slope at the central pixel of the window. To assigne a slope value to the next pixel, the window is shifted to the next pixel, thus windows are overlapping. This plane can be defined by:

$$0 = aX + bY + cZ + d \tag{2.1}$$

which can be simplified to

$$Z = a^* X + b^* Y + c^* \tag{2.2}$$

where Z is the "height" of a point on the plane and X and Y are the x- and y- coordinates of the plane in the frame defined by the local gravity vector. a^*, b^*, c^* are constant for a plane and fully define it. Equation (2.2) can be expressed in matrix form as

$$\underbrace{\begin{bmatrix} X_1 & Y_1 & 1 \\ X_2 & Y_2 & 1 \\ \vdots & & \\ X_n & Y_n & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} a^* \\ b^* \\ c^* \end{bmatrix}}_{\boldsymbol{x}'} = \underbrace{\begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix}}_{\boldsymbol{b}} \tag{2.3}$$

The least-squares solution for $\hat{\boldsymbol{x}}$ is can be computed by singular-value decomposition. The slope, S, of this patch can then be computed from:

$$S = \arccos(\hat{\boldsymbol{n}} Z) \tag{2.4}$$

where $\hat{\boldsymbol{n}}$ is the surface normal of the plane given by $|(a, b, -1)|^T$.

2.3.2. ROUGHNESS ESTIMATION AND TEXTURE DETECTION

Roughness is defined by features that differ from the local mean plane (as shown in Fig. 2.2). Therefore, the roughness can be computed as the deviation from the mean plane. Here, the mean plane described in Eq. (2.2) is used. Using the root-mean-square (RMS), the roughness, R, can be computed by:

$$R = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(z_i - z_{\text{mean}})^2}. \tag{2.5}$$

where z_{mean} is given by the mean plane and N is the number of pixels in the window defined by the lander footprint size. This method is capable of computing the height of a roughness feature. It is therefore a quantitative assessment of the roughness in the landing region. The quality of this roughness strongly depends on the accuracy of the slope (and thus the mean plane) computed.

However, not only the mean plane can be used to determine the local roughness, also the texture of the input images can give information on the roughness of the terrain. To this end, Haralick micro-texture indicators (Haralick et al., 1973) and histogram-based methods (Dekker, 2003) were investigated. Texture detection has the capability of detecting deviations of the local texture from the global texture of an image. If such a deviation is found, this means that the global texture is disturbed. This could be a rock lying on the surface, but also the rim of a crater.

The simplest way to evaluate texture is to evaluate the pixel intensities over a given window, so-called histogram-based texture detection. This can be done using different texture measures. In this research two of these measures were investigated, namely the mean and the variance.

$$\text{mean} = \mu = \frac{1}{N}\sum_{i,j} I_{i,j} \tag{2.6}$$

$$\text{variance} = \sigma^2 = \frac{1}{N-1}\sum_{i,j}\left(I_{i,j} - \mu\right)^2 \tag{2.7}$$

where $I_{i,j}$ is the intensity value of the pixel at the location (i, j) and N is the total number of pixels of the window used (Dekker, 2003).

Haralick micro-texture indicators make use of the Gray-Level Co-Occurrence Matrix (GLCM), $p(i, j)$, rather then using the pixel values directly. The GLCM is defined as

$$p(i,j) = \sum_{p=1}^{n}\sum_{q=1}^{m}\begin{cases}1, & \text{if } I(p,q) = i \text{ and } I(p+\Delta x, q+\Delta y) = j.\\ 0, & \text{otherwise.}\end{cases} \tag{2.8}$$

with Δx and Δy being offsets from the centre pixel, and i and j are image intensity val-

ues. The following three descriptors were used in this research (Haralick et al., 1973).

$$\text{energy} = \sum_{i,j} p(i,j)^2 \tag{2.9}$$

$$\text{contrast} = \sum_{i,j} (i-j)^2 p(i,j) \tag{2.10}$$

$$\text{homogeneity} = \sum_{i,j} \frac{p(i,j)}{1+|i-j|} \tag{2.11}$$

Figures 2.8 and 2.9 show the texture maps for the five evaluated algorithms, as well as the roughness map generated using the deviation from the mean plane, no colour bars are given since all of the presented method are a qualitative assessment of the scene. The first scene contains one crater (Fig. 2.8a). One can expect that the rim of this crater is identified as roughness. Analysing the results for the first scene in Fig. 2.8, it can be concluded that only the results of Haralick energy (Fig. 2.8c) and histogram-based mean (Fig. 2.8e) did not distinctly detect the rim and the slopes. Moreover, both methods seem to have problems with shadowed regions, as they both detect the shadow inside the crater as high texture. Haralick homogeneity (Fig. 2.8d) does not detect large differences in the image. This will make it difficult to tune the threshold. Overall, the histogram-based variance (Fig. 2.8f) seems to detect the rim in the most complete way. Furthermore, it seems that combining this method with a mean-plane algorithm (Fig. 2.8g) will lead to the best results. Here, it is important to note that the quality of the results of the mean-plane algorithms strongly depends on the quality of the DEM.

The second scene contains rocks. It is expected that for a suitable algorithm all rocks are distinguishable from the rest of the scene. If craters or crater rims are identified this is favourable. Furthermore, false alarms should be minimised. Since the total number of rocks in the scene is known, it is possible to evaluate how many of these rocks are correctly detected. Also, it can be investigated how many rocks are detected in locations where no rocks were present. The percentages of correct detections, detection errors and false-alarm rates are presented in Table 2.3. Investigating Figs. 2.9a to 2.9g in combination with Table 2.3, it can be concluded that apart from Haralick energy (Fig. 2.9c), histogram-based mean (Fig. 2.9e), and the mean-plane algorithm (Fig. 2.9g), all methods detect the rocks. Haralick energy does not produce any useful result, while histogram-based mean detects some of the rocks, but primarily detects shadow as roughness. Again, Haralick homogeneity (Fig. 2.9d) results in a map with very little variation in colour, and thus texture level. This will make it difficult to distinguish rocks from noise, which can be seen in the high false-alarm rates. The mean-plane algorithm (Fig. 2.9g), on one hand, seems to overestimate the size of the rocks and does not detect small rocks as well as other methods and produces a lot of false alarms. On the other hand, it also detects the small craters in the scene. However, the histogram-based variance (Fig. 2.9f) detects the rims of these craters and the rocks very well. Furthermore, its false-alarm rate is very low. Haralick contrast (Fig. 2.9b) performs equally well in terms of correct detections, but has a higher false-alarm rate. It is thus found that histogram-based variance is the best choice for roughness detection.

Therefore, histogram-based variance and the deviation from the mean plane will be

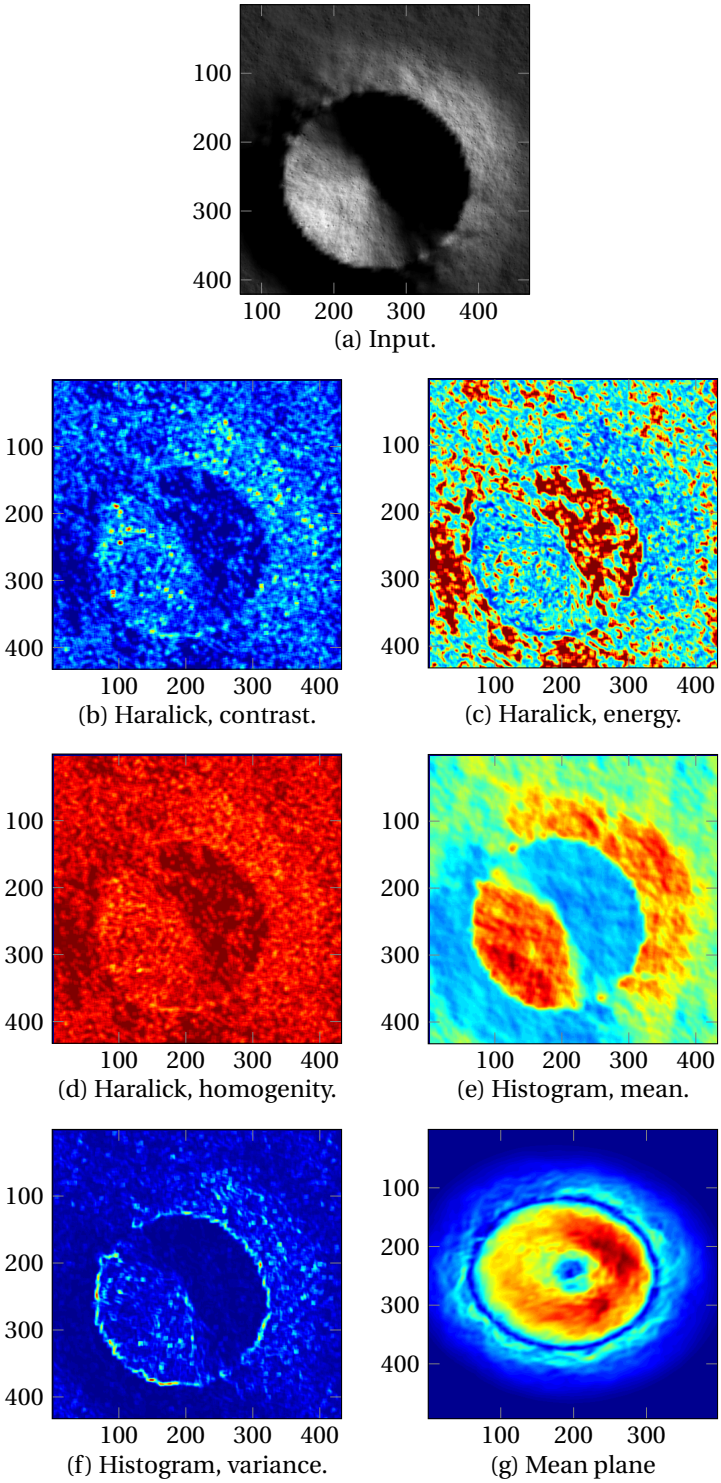
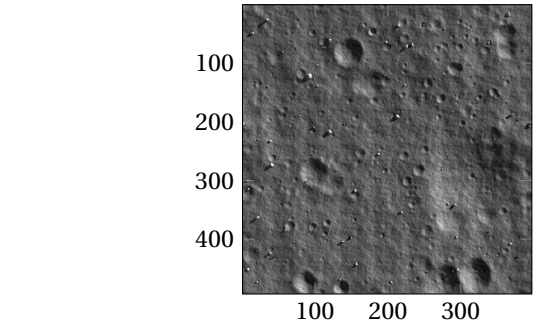
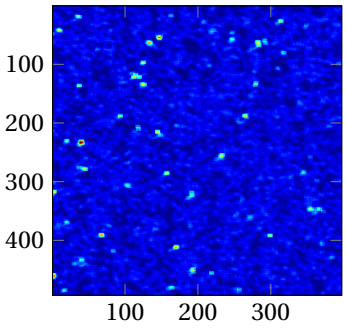


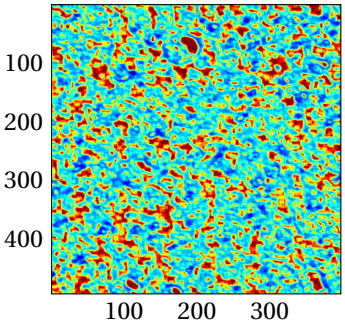
Figure 2.8: Scene 1, roughness estimation and texture-detection trade-off.



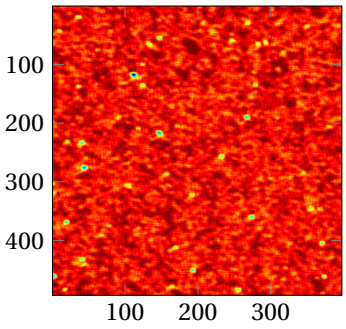
(a) Input.



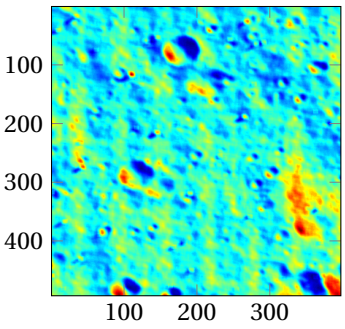
(b) Haralick, contrast.



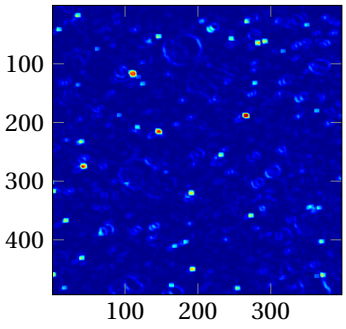
(c) Haralick, energy.



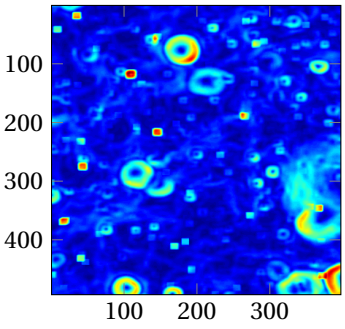
(d) Haralick, homogenity.



(e) Histogram, mean.



(f) Histogramm, variance.



(g) Mean plane.

Figure 2.9: Scene 2, roughness estimation and texture-detection trade-off.

Table 2.3: Rock counts for results in Fig. 2.9, Scene 2 (60 rocks).

Method	Rocks detected	Rocks undetected	False alarms
Haralick, contrast	44 (73.3 %)	16 (26.7 %)	10 (18.5%)
Haralick, energy	0 (0.0 %)	60 (100%)	0 (0.0 %)
Haralick, homogeneity	43 (71.6 %)	17 (28.4%)	20 (31.7%)[1]
Histogram, mean	6 (10.0 %)	54 (90 %)	11 (64.7 %)
Histogram, variance	45 (75 %)	15 (25%)	3 (6.3%)
Mean plane	19 (31.6 %)	41 (68.4%)	7 (26.9%)

[1] This depends on the way of counting, either one gets large number for detected hazards and large numbers for false alarm or both numbers are low.

combined to detect roughness. The histogram-based variance can be found from:

$$\text{variance} = \sigma^2 = \frac{\sum_{i,j} \left(I_{i,j} - \mu\right)^2}{N-1} \tag{2.12}$$

Roughness is detected if this variance is larger than a certain threshold, which has to be tuned. This threshold depends on the quality of the image, but also on the terrain encountered.

Contrary to the deviation from the mean-plane algorithm, this method is not capable of computing the height of a roughness feature. It is therefore a pure qualitative assessment and cannot distinguish between "safe boulders", *i.e.*., small enough, and "unsafe boulders".

2.3.3. HAZARD MAPPING

After the slope map, roughness map and texture map have been created, it is possible to combine them into the final hazard map. Here, an input image is required as well to also include shadow hazards. Shadowed regions are those, which are too dark in the input image. To detect these, all image pixel intensities, which are below a certain image intensity value, are marked hazardous. This threshold may simply be fixed, but could also be variable based on the overall image intensities.

The final hazard map will be represented on a scale of zero to one, where a value of one indicates regions that are hazardous. The hazardousness of regions are defined by thresholds, which can be changed according to the mission requirements. As stated earlier these thresholds are 15 deg for slope and 0.5 m for roughness. The thresholds for shadow and texture are tuning parameters. These values do not correspond to a specific roughness, but are tuning parameters that have to be tuned based on *a-priori* knowledge of the landing region. For example, when landing in a very dark region the shadow threshold should be lower than when landing in a well-illuminated region. For the analysed scene intensities below 20 and a histogram-based variance (texture) above 300 were determined as thresholds. These were selected based on analysis of the input images. For a real mission, the sensitivity and robustness of these values should be studied based on the *a-prior* knowledge of the target body, landing region and illumination conditions.

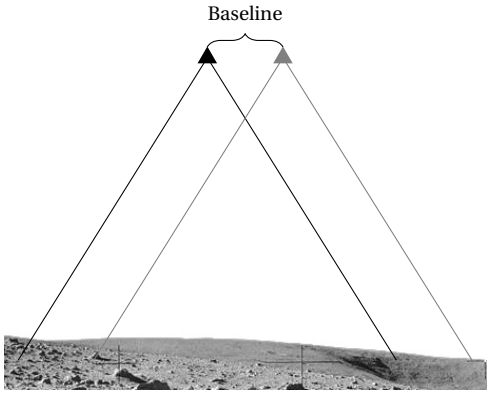


Figure 2.10: Stereo vision.

2.4. CAMERA-BASED HAZARD DETECTION

To compute dense DEMs from camera images, three different methods can be used: stereo vision (SV), stereo from motion (SfM), and shape from shading (SfS), see Table 2.2. The latter two require one camera, while the former one requires two cameras mounted on the lander with a fixed baseline. To date no other vision-based dense HDA method has been applied and used for space applications. Other camera-based mapping techniques might exist but are currently infeasible for implementation due to computational constraints, the extreme robustness required or limitations of the set-up.

2.4.1. COMPARISON OF DIFFERENT METHODS

In the following a brief introduction to these three methods is given. A more detailed elaboration on stereo vision will be presented later in this chapter. For now, the discussion will focus on the basic working principles of the individual methods.

STEREO VISION

Stereo vision is a computer vision technique miming the human eyes. Two cameras are attached rigidly with a fixed, known, baseline. This set up is shown in Fig 2.10.

The generation of the DEM is the main challenge for a stereo-based hazard-detection algorithm, opposed to active sensors, where the focus is mainly on the correct reconstruction of slope and roughness, because the DEM is measured and not reconstructed. Also, the quality of the generated DEM directly influences the quality of the slope and roughness maps. This is especially because roughness and noise behave quite similarly, as roughness is in principal a noise on the surface normal. It will thus always be difficult to distinguish between noise and roughness. Therefore, DEMs should contain as little noise as possible.

To reconstruct the scene from a stereo-image pair it is necessary to know the baseline of the set up and the focal length of the cameras. These variables are constant for the entire mapping phase. In addition, the disparity has to be known. The disparity (d) is different for every pixel in an image. From this information the distance from the camera

to a point in the scene can be computed from (e.g. (Hartley and Zisserman, 2004))

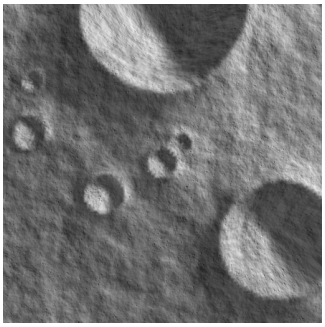
$$Z(d) = \frac{fb}{d} \tag{2.13}$$

where $Z(d)$ is the distance to the surface as a function of the disparity. Both the focal length, f, and the baseline, b, are fixed by design. The disparity is the distance that the projection of a real life point jumps between the left and the right stereo image, as shown in Fig. 2.11. In this figure, the left and right stereo images are shown, as well as the overlay of the two images. In the overlay, Fig. 2.11c, it is distinct that the features move in the second image relative to the first. If this movement is measured for a specific pixel, the disparity is found. It can be computed by searching for a pixel from the left image in the right image. This searching process is called "matching". Here, one makes use of the epipolar constraint. This constraint ensures that the same surface point lies on the same line in corresponding images, thus one only needs to search along a line to find a correspondence. However, it only holds for perfectly aligned cameras. If the cameras are not perfectly aligned, images need to be rectified before the epipolar constraint can be exploited. During this development of the stereo HD method, images are generated such that perfect alignment is given. During a real mission, thorough (re-)calibration will be required to obtain accurate knowledge of the camera alignment.

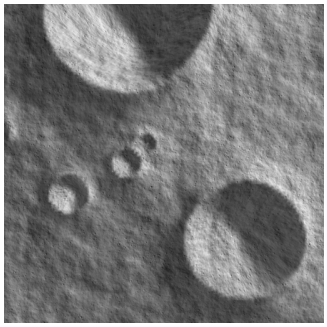
The chosen matching technique greatly influences the quality of the resulting DEM but also the required execution time. From a thorough literature study based on (Brown et al., 2003) and (Tombari et al., 2008), it was concluded that dynamic programming and block matching are suitable candidates. Block matching is a very simple technique comparing a pixel and its neighbours to each potential matching candidate pixel, while dynamic programming does the same however tries to optimise certain constraints along a scanline, *e.g.,* smoothness. During this study it was found that most more advanced matching algorithms were not applicable for the algorithm due to the very short runtime requirement, but also since they were designed to handle occlusion problems, which are a big problem in robotics applications, but not for hazard detection. In hazard detection objects occluded will also always be in the shadow of the occluding object and thus an illumination hazard. Both matching functions were implemented and tested on images of planetary surfaces generated by PANGU. It was found that dynamic programming was at least twice as slow as the block-matching algorithm, while the results did not differ significantly. Moreover, dynamic programming led to wrong results at the right map boundaries. Therefore, it was chosen to use the block-matching algorithm using the sum-of-squared differences (SSD). The SSD was chosen over the sum-of-absolute differences (SAD) and normalised-cross-correlation (NCC) (Patil et al., 2013), as it was found to deliver better results. The SSD matching cost is computed from:

$$\text{SSD} = \sum_{u,v} (I_1(u,v) - I_2(u+d,v))^2 \tag{2.14}$$

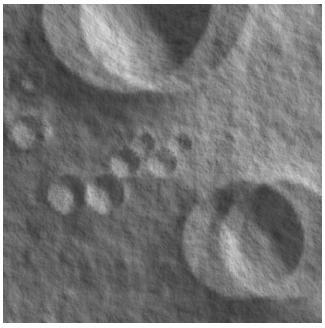
where $I(u,v)$ defines the intensity value of the images at the pixel positions given by (u,v). Furthermore, it was found that the optimum size of the matching window has to be 7×7 pixels, by increasing the window size from 1×1 to 21×21 pixel. Starting from 7×7 pixel, the performance did not improve any further. Smaller windows will lead to



(a) Left image.



(b) Right image.



(c) Combined.

Figure 2.11: Matching of stereo-vision images.

worse DEMs, while larger windows will mainly increase the execution time, but will not improve the quality of the DEM.

Since the matching is performed for terrains at a considerable distance from the camera, it is important to implement some measures to improve the depth resolution, which is the resolution in z-direction, and thus the most important factor to resolve boulder height. The maximum depth resolution, δz, that is achievable using stereo vision can be computed from

$$\delta z = \frac{Z^2}{bf}\delta d \tag{2.15}$$

As discussed before, b and f are fixed for a given design. z, the altitude above the surface, and δd, the disparity step, are the design parameters that can be altered to increase the quality of the results. z is the mapping altitude and is constrained by the mission design. Thus, decreasing δd is the best option to obtain a better depth resolution.

If matching is performed without any additional functions then $\delta d = 1$ pixel. This will result in a depth resolution, which is clearly infeasible for mapping as boulders of 0.5 m height need to be resolved. One solution to decrease δd below 1 is by upsampling the

image and thereby reducing δd. If, using linear interpolation based on the neighbouring intensities, one pixel is inserted in between every existing pixel, this would reduce the minimum δd to 0.5. An example of the resulting matching scores and the minimum disparity is shown in Fig. 2.12a. This approach was used in an earlier version of this algorithm (Woicke and Mooij, 2014). For that version it was found that only mappings at an altitude of 100 m and below are feasible. Further decreasing δd proved to be infeasible due to the requirement on the execution time. Moreover, the results met the requirement, but still problems due to the still finite depth-resolutions were clearly visible. The resulting depth resolutions, and thus terrain elevation values, were all grouped in bins linked to the finite minimum disparity (0.5 pixel). Therefore, all pixel in the resulting DEMs were grouped in these bins. No smooth transitions between different terrain elevations could be achieved. Namely, the stepping effect, which was causing "false-alarm-lines", as discussed in (Woicke and Mooij, 2014).

To overcome the limitation of the limited δd, fitting a quadratic function through the SSD scores around the minimum (integer) disparity is used in this updated version of the algorithm. It has to be noted, that all cost not linked to the minimum disparities and its two neighbouring values do not contain any information which would be helpful to retrieve the minimum disparity. Therefore, only these three points should be used for the computation of the minimum disparity. In Fig. 2.12a, the SSD matching scores as a function of the disparity is shown. In the case of pure upsampling, only distinct disparity steps can be resolved. In this case the algorithm would return 12 as the minimum disparity. In Fig. 2.12b the quadratic fit is shown, again using the same matching scores, however without the added values from the upsampled pixel. Again, 12 would be returned as the minimum disparity. However by fitting a curve trough the scores, ≈ 12.2 is found as the minimum disparity. This gives even better results and enables mapping at higher altitudes. The quadratic fit can be computed using the matching scores for the minimum disparity as well as for its left and right neighbours, $d_{\min}-1$ and $d_{\min}+1$. Only the x-coordinate of the origin of the fitted parabola has to be known, as this is the desired disparity. The updated minimum disparity can be computed from:

$$\hat{d}_{\min} = d_{\min} - \frac{C(d_{\min}-1) - C(d_{\min}+1)}{4C(d_{\min}) - 2C(d_{\min}-1) - 2C(d_{\min}+1)} \tag{2.16}$$

where C is the cost function used, in this case the SSD. However it is important to note that applying this method will not allow for computing a theoretical value for the minimal depth resolution, as the minimum disparity is not limited any more.

Furthermore, the DEM will be filtered after it is computed to remove outliers that might be present in the resulting map. This is very important for the quality of the slope and roughness maps. It was found that a combination of a linear prediction and a median filter are most beneficial. The linear prediction filter can be described by:

$$\bar{z}_{i,j} = \begin{cases} z_{i,j} & \text{if } \left| \dfrac{z_{i,j}}{\left(z_{i-1,j} + \frac{z_{i+N,j} - z_{i-N,j}}{2N} \right)} \right| < k \\ z_{i-1,j} + \dfrac{z_{i+N,j} - z_{i-N,j}}{2N} & \text{else} \end{cases} \tag{2.17}$$

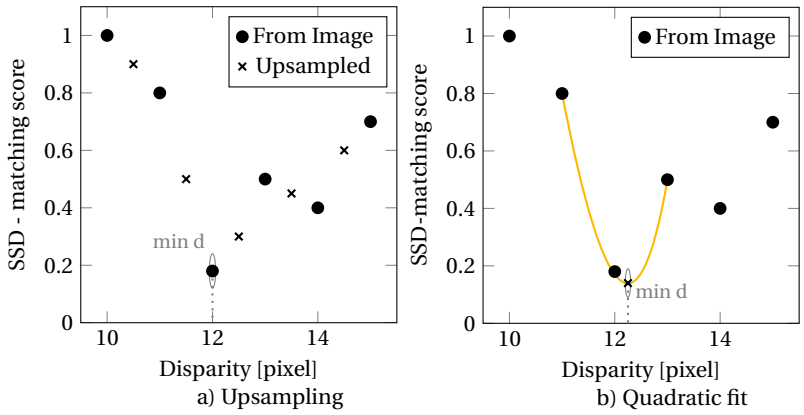


Figure 2.12: Upsampling compared to quadratic fit.

This logic describes the following steps: based on the neighbouring pixels, located N pixels to the left and to the right of pixel (i, j), the local one-dimensional slope is computed. Based on this slope the terrain elevation of pixel (i, j) is predicted. If terrain elevations at (i, j) differ more than a factor k from the prediction, the terrain elevation is discarded and replaced by the prediction. In the proposed algorithm $k = 1$ was chosen as higher higher values would smooth the DEM too much and lower values would not remove the sufficient outliers. The median filter is given by

$$\bar{z}_{i,j} = \text{median}\{z(u,v) \mid u = i - N/2, i - N/2 + 1, \ldots, i - N/2 + N, \\ v = j - N/2, j - N/2 + 1, \ldots, j - N/2 + N\} \tag{2.18}$$

which represents the median of all pixels in an $N \times N$ window around the centre pixel located at (i, j). The presented algorithm uses $N = 3$, for which analysis of values from $N = 1$ to $N = 10$ showed it to be the best choice, as it does not smooth the resulting DEM too much while still removing outliers. Again it must be stressed that removing noise is always a trade-off between removing algorithmic errors and removing roughness features, since it is not easy to distinguish between these two.

Since the reconstruction of the DEM is done scanline by scanline, and no inter-scanline constraints are used, individual scanlines can be computed independent from each other, for example, using parallel processing. This might be desirable to decrease the execution time, if necessary.

STEREO FROM MOTION

Like stereo vision, stereo from motion requires two images. However, the two necessary input images are not taken by two different cameras at the same instant in time, but are rather obtained from the same camera at two different instances in time as shown in Fig. 2.13. The approach presented here is based on (Xiong et al., 2001).

Because the two input images are taken at different altitudes during the descent, they will have different resolutions. Furthermore, the orientation and location of the camera might have changed in between capturing the two images. Therefore, it is necessary to

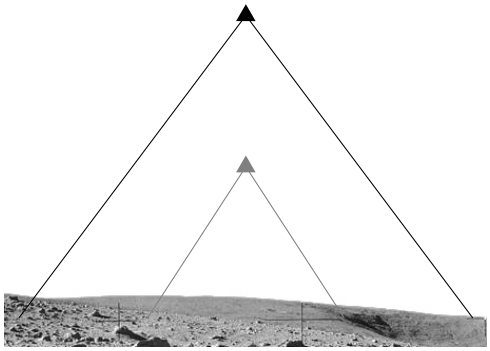


Figure 2.13: Stereo from motion.

perform both image downsampling and image warping to make the two input images compatible before using them as input images for the SfM algorithm. Both downsampling and warping will require knowledge of the camera position and orientation, while capturing the image. This can either be done by the SfM algorithm itself, by including a feature-tracking function or could be an output of the navigation system, for example, from terrain relative navigation. Obviously, the camera state should be known as accurately as possible, because inaccuracies will lead to errors in the results. For this trade-off, it is assumed that the camera positions are known from the navigation system. Moreover, a purely vertical descent is analysed. Since errors in the state estimation will cause the algorithm to perform worse than in the error free case, the results presented for this algorithm should be seen as the best possible results. If this method should appear as a feasible candidate, the sensitivity of the method to error in the state estimation should be analysed. Moreover, it should be noted that the method constrains the potential trajectory flown. The pure vertical descent case, as shown in Fig. 2.13, maximises the overlap of both descent images, but also presents the limiting case with the epipole, the point where no reconstruction is possible, located in the image centre. Horizontal, well known, motion in between image acquisition, does not cause any problems to the algorithm, it will only move the epipole location. Moreover, it is a reasonable assumption that the trajectory flown during the last metres of the descent is vertical.

In general, the approach for SfM is the same as for SV, however, this time the algorithm is not searching for the minimum cost for all possible disparities for a certain pixel and then derive the terrain elevation from it, but rather an terrain elevation is assumed, the cost is computed and the minimum cost is assumed to occur at the “correct” terrain elevation. The image warping takes care of linking the correct pixels. As for SV, the cost function used is the SSD, see Eq. (2.14). Also, like for SV, a parabolic fit is used to obtain terrain elevation values that are not fixed to the selected “step” as represented by Eq. (2.16). Furthermore, the algorithm needs a certain terrain elevation range and terrain elevation step as an input. Therefore, some *a-priori* information on the terrain images is necessary. If the cost is computed for every pixel in the image at every possible terrain elevation and the minimum cost terrain elevation is found per pixel, the full DEM is generated. Obviously, the quality of the result strongly depends on the terrain elevation

step. However, very small step sizes will also lead to a higher computational load.

SHAPE FROM SHADING

Shape from shading, Fig. 2.14, exploits the fact that the intensity of an image is uniquely linked to the surface orientation of the image at that given point. Knowing this, one can conclude that there should be a method to derive the surface orientation from an intensity image, I. Therefore, there should be an intensity function that is a function of surface orientation, *i.e.*, the slope (Horn, 1977). Opposed to SV and SfM, SfS computes the DEM based on only one input image. In addition to the input image, the Sun elevation, e, and azimuth, α, of the incoming Sunlight have to be known. Furthermore, a good first guess for the maximum and minimum surface altitude should be present.

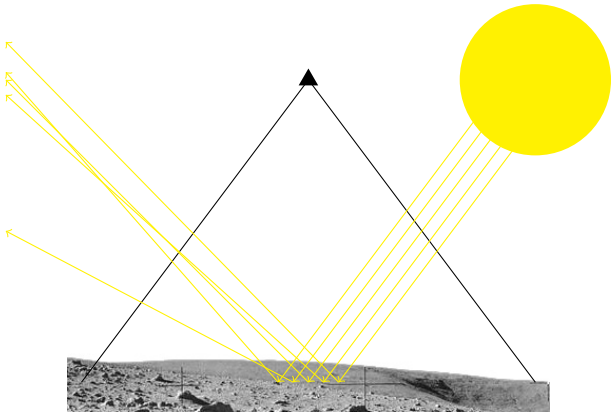


Figure 2.14: Shape from Shading princple.

For this SfS implementation a very basic SfS principle is used. Some important assumptions are made. 1) The image is rotated in the direction of incoming sunlight, as shown in Fig. 2.15. This means that the angle between the x-axis and the projection of the Sun into the image, τ, is zero (see Fig. 2.22 (b)). 2) The surface is a Lambertian (*i.e.*, perfect) reflector 3) The albedo of the surface is uniform. 4) The Sun is the main light source and all shadows are caused by this source.

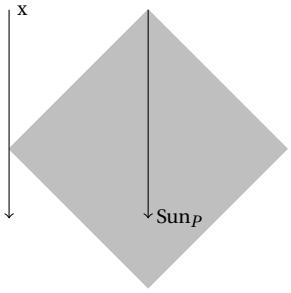
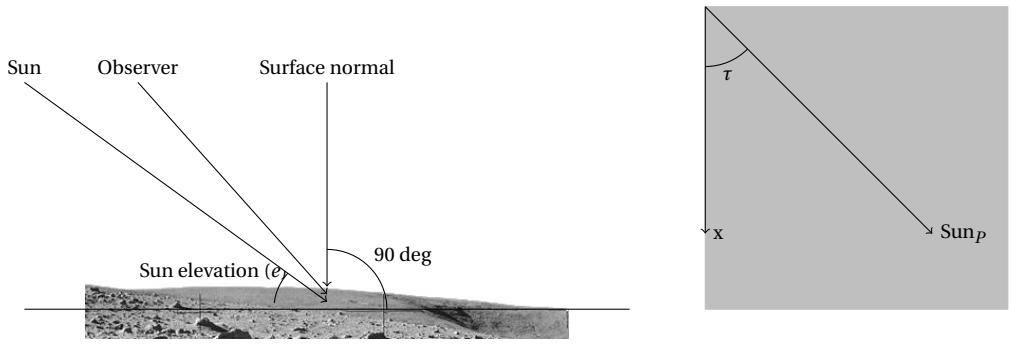


Figure 2.15: Shape from shading rotating the image.



(a) (b)

Figure 2.16: Shape from shading, angles. Sun_P is the sun projected in the image plane.

In Pentland (1990) it is shown that the image intensity for a Lambertian reflector can be approximated by a linear function, given that the angle between the viewer and the incoming Sun light is large. This should be ensured by mission design and camera layout. The geometry of the problem and necessary variables are indicated in Fig. 2.16. This linear approximation is given by (Pentland, 1990)

$$I(x,y) \approx a(\cos e + p\cos\tau\sin e + q\sin\tau\sin e) \tag{2.19}$$

where p is the slope in x-direction, q is the slope in y-direction directly, and a is the surface albedo (Pentland, 1990). Since it was assumed that $\tau = 0$ the equation simplifies to

$$I(x,y) \approx a(\cos e + p\sin e) \tag{2.20}$$

As p is the slope in x-direction, the terrain elevation of the scene, z, can be reconstructed along a line in x-direction from

$$z_{x,y} = z_{x-1,y} + p \tag{2.21}$$

where x is measured in the unit of pixels. Solving Eq. (2.19) for p, the terrain elevation becomes

$$z_{x,y} = z_{x-1,y} + \frac{I(x,y) - a\cos e}{a\sin e} \tag{2.22}$$

The albedo can either be considered as known or computed from the expectation of the intensity as

$$a \approx \frac{E(I)}{\cos e} \tag{2.23}$$

Using Eq. (2.22) the terrain elevation of every point in the scene can be computed per point along an x-line. Because this SfS method leads to line artefacts in the output images two measures are taken to reduce these effects. 1) The resulting DEM is filtered in

y-direction using an averaging filter, to level the effect of the lines. 2) The integration is performed once from the bottom up and once top down after which the results are averaged. The resulting DEM is the final DEM as generated using SfS.

A comparable algorithm can be found in (Neveu et al., 2015). It describes an SfS HD algorithm based on the same method as given above. Furthermore, it uses the same method to deal with the line artefacts in the resulting DEMs.

2.4.2. RESULTS AND PERFORMANCE OF THE ALGORITHMS

In this section results of all three algorithms are presented. Here, different scenes were selected for the different algorithms. This is done, because the different algorithms have different regimes they perform well in, as will be discussed in Sec. 2.4.3. Obviously, this section does not constitute a full performance analysis, but is rather meant to give the reader a general idea of the performance of the algorithms.

All algorithms are complemented with slope, roughness, and full hazard maps, computed as described in Sec. 2.5. Both slope and roughness are computed by fitting a mean plane through the terrain elevation point. Furthermore, a texture-detection method is added for better detection of roughness, as discussed later in this chapter in Sec. 2.5. Combining all these maps a hazard map can be constructed and compared to the ground-truth hazard map.

All algorithms are analysed for their mean errors and standard deviations. This mean error is defined by the absolute difference between the computed terrain elevation and the true terrain elevation. This way the mean error is representative of the ability to detect roughness features, whereas the standard deviation is indicative of the precision of the results. Overall, the problem is that few extreme outliers can strongly influence these statistics for which reason the results should always be discussed in context with the actual DEMs and more weight is given to the overall hazard detection performance.

This hazard detection performance is evaluated by comparing it to the true hazard map computed from the ground-truth maps. The results of the comparison can be grouped in the following four different classes: 1) False negative (FN) a hazard, undetected by the algorithm (wrong detection); 2) False positive (FP) defines a safe site, which is erroneously labelled as unsafe (false alarm); 3) True negative (TN) is a correctly detected hazard; and 4) True positive (TP) is a correctly detected safe site.

STEREO VISION

In Woicke and Mooij (2014) it was found that SV has acceptable performance for HDA at altitudes below 150 m for baselines of a minimum of 2 m. Therefore, a scene imaged at 100 m altitude with a camera baseline of 2 m is presented in the following. The input image is shown in Fig. 2.17. From the results presented in Table 2.4 and Figs. 2.18 to 2.21, it can be concluded that the stereo-vision hazard detection algorithm presented in this work reconstructs the scene very well, with only a few outlier pixels as an exception. Furthermore, the resulting hazard map is very close to the actual solution, as can be seen in Fig. 2.21. This leads to the conclusion that SV is a suitable hazard-detection algorithm for comparable scenarios. In the table the mean errors, μ, and standard deviation, σ, for DEM, slope, and roughness are presented. These values seem rather high, however when investigating the slope and roughness maps presented in Fig. 2.20 and Fig. 2.19

Table 2.4: Performance stereo vision

	FN	FP	TN	TP	
	1.75%	20.07%	37.89%	40.29%	
μ_{DEM}	σ_{DEM}	μ_{slope}	σ_{slope}	$\mu_{\text{roughness}}$	$\sigma_{\text{roughness}}$
1.52 m	11.42 m	5.88°	9.88°	1.37 m	8.64 m

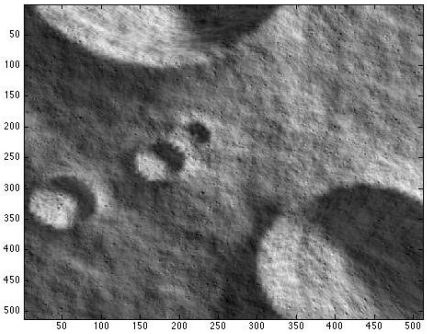


Figure 2.17: SV input image.

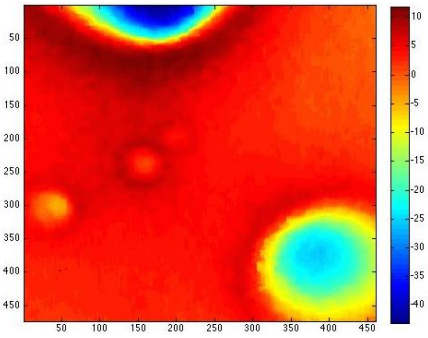


Figure 2.18: SV DEM [m].

it can be seen that there are some local extreme outlier regions, like on the crater rim. These strongly influence the error statics, while the remaining parts of the slope and roughness maps are estimated a rather well. For a better idea of the performance, the percentages for undetected hazards (FN), safe sites classified as hazards (FP), correctly identified hazards (TN) and correctly identified safe sites (TP) are given. It can be concluded that all mean errors are low, however, the standard deviations show that some outliers are present. Very few hazardous sites are not identified, and the total number of correct detections (TN + TP) is very high.

As mentioned above, Woicke and Mooij (2014) showed that the present SV algorithm performs well at altitudes below 150 m for baselines of 2 m and more. Contrary to the algorithm presented in there, the algorithm used for this research was supplemented

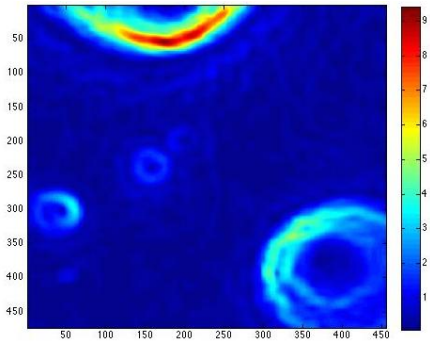


Figure 2.19: SV roughness [m].

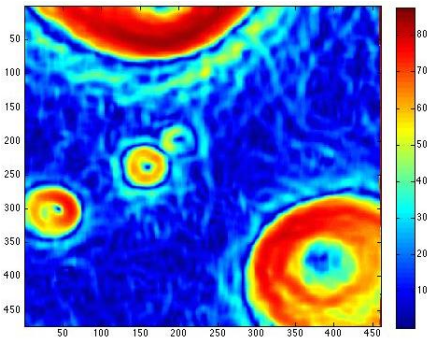


Figure 2.20: SV slope [°].

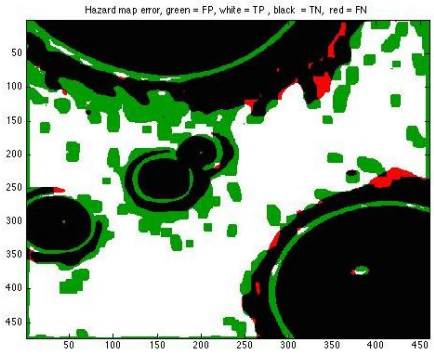


Figure 2.21: SV hazard map error.

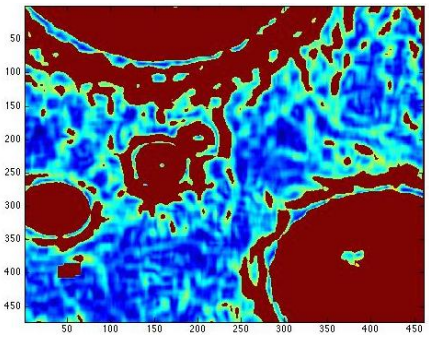


Figure 2.22: SV Hazard map.

with a parabolic fit for computing non-integer disparity values. With this alteration the algorithm is able to perform HD up to altitudes of 200 m. This seems to be low, but is sufficient for one local hazard-detection survey, mainly searching for features that cannot be resolved at higher (orbital) altitudes, *i.e.*, boulders. As shown by Woicke and Mooij (2014), the SV algorithm always managed to detect all boulders for the analysed scenes. Therefore, it can be concluded that SV is a suitable candidate for HD at low altitudes. Moreover, it shows good performance with respect to boulders, *i.e.*, roughness, detection.

STEREO FROM MOTION

As compared to SV higher baselines can be achieved using SfM. Therefore, it is able to deliver good results at higher altitudes than SV. To underline this assumption, the results in these scene are computed for a scene imaged at 400 m, Fig. 2.23, and 300 m, Fig. 2.24. Before analysing the results it should be noted that the epipole of the image cannot be reconstructed. This is the same for SV, however, the epipole is outside of the image for SV. For the SfM setup used in this research the epipole is located at the image centre. This will always be the case for DEMs constructed during the final phase of a descent, since the motion will be very close to a vertical descent. Slight offsets from a lateral motion will slightly shift the epipole location, but it will still fall inside the DEM. Therefore, all resulting maps will be inaccurate at the image centre, this can be seen clearly in Fig. 2.25, *i.e.*, towards the image centre the DEM seems to be random. This fact is also represented in the very large mean errors and standard deviation, since a large portion of the scene is simply reconstructed incorrectly. Also, SfM can only reconstruct a fraction of the scene as imaged by the first camera, as obviously the second image will only show a part of the scene. It is furthermore important to note that usually the nominal landing site will be located in, or close to, the image centre. This means that hazard assessment on the nominal landing site is not possible with SfM. This is clearly a shortcoming of the algorithm.

The results are presented in Table 2.5 and Figs. 2.25 to 2.29. The table gives mean errors and standard deviation for DEM, slope, and roughness as well as the FN, FP, TN and TP percentages. The number of FN is considerably higher than for the SV algorithm, however, in total the number of correct detections is in the same order. For this algorithm

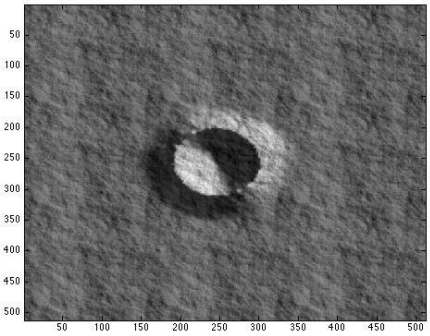


Figure 2.23: SfM input image, high altitude.

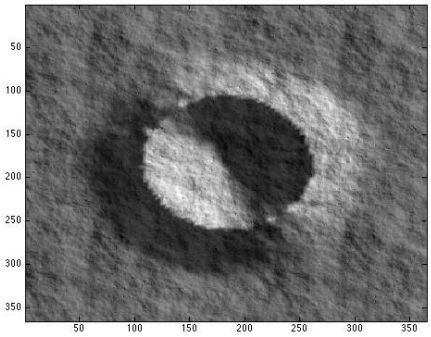


Figure 2.24: SfM input image, low altitude.

the standard deviations are a lot smaller as compared to SV, which indicates that the resulting DEM is more uniform.

Table 2.5: Performance shape from shading.

	FN	FP	TN	TP	
	5.17%	22.21%	23.60%	49.02%	
μ_{DEM}	σ_{DEM}	μ_{slope}	σ_{slope}	$\mu_{\text{roughness}}$	$\sigma_{\text{roughness}}$
2.15 m	2.32 m	17.61°	16.99°	0.82 m	1.03 m

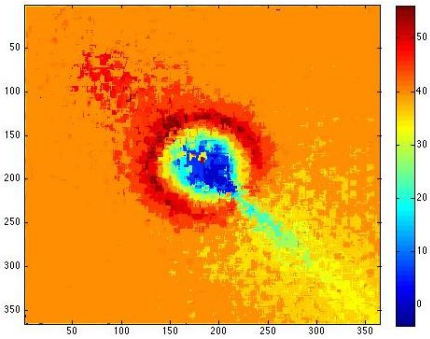


Figure 2.25: SfM DEM [m].

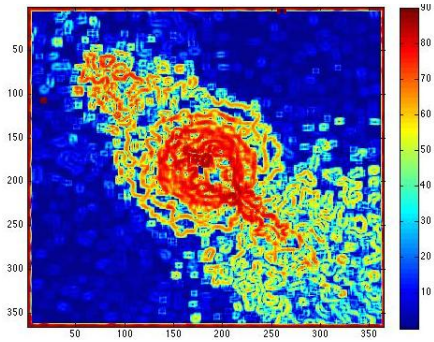


Figure 2.26: SfM slope [°].

Based on a larger number of SfM executions for different scenes, the conclusion that it should perform comparable to SV with the only difference of being able to survey the terrain already at higher altitudes was confirmed. Since the baseline can be freely chosen as opposed to the fixed, narrow baseline for SV, successful mapping completely independently of altitude should be possible. Eventually, the size of the resulting DEM will get too small, due to the projection of the second image onto the first one. As for SV, SfM is able to detect roughness features well. However, the method cannot reconstruct the scene in and around the epipole, which renders the image centre useless for hazard

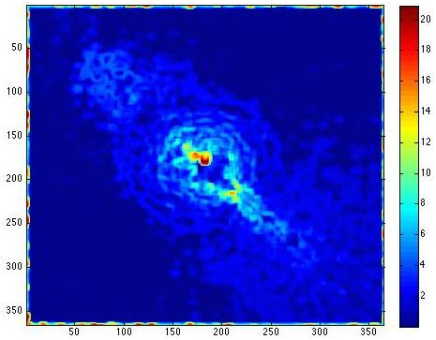


Figure 2.27: SfM roughness [m].

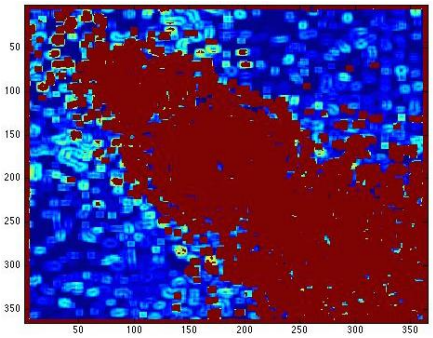


Figure 2.28: SfM hazard map.

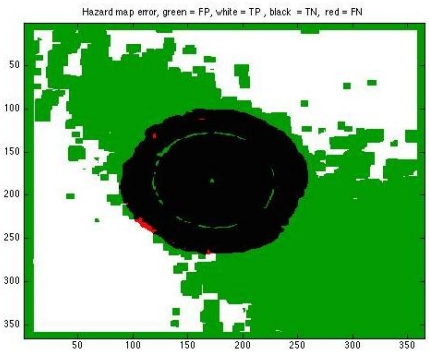


Figure 2.29: SfM hazard-map error.

detection purposes.

SHAPE FROM SHADING

Shape from shading does not depend on the altitude while imaging, since it only computes gradients. The presented results are created for a scene imaged at 1 km altitude. The input image is shown in Fig. 2.30. It can be seen in Fig. 2.31 that it seems like the DEM is not reconstructed well. However, when investigating the roughness (Fig. 2.32), slope (Fig. 2.33) and hazard map (Fig. 2.35), one can conclude that the algorithm works well. However, some of the line artefacts are still visible.

The error percentages, means and standard deviations are given in Table 2.6. The table underlines the previous findings that the DEM is not reconstructed well, while slope and roughness maps are of better quality. Again very large local errors are present which strongly influence the error statistics. Looking at the images one can see that especially compared to SV and SfM, slope and roughness are reconstructed well even though the error statistics might indicate otherwise. This is, because opposed to SfM and SV, SfS does compute the slope and then derives the DEM from the slope, thus the DEM is only a secondary product. This means that a mediocre DEM does not necessarily relate to a bad overall performance of the algorithm. One should note, however, that most rough-

Table 2.6: Performace shape from shading.

	FN	FP	TN	TP	
	0.45%	37.92%	14.52%	47.11%	
μ_{DEM}	σ_{DEM}	μ_{slope}	σ_{slope}	$\mu_{\text{roughness}}$	$\sigma_{\text{roughness}}$
13.36 m	10.05 m	5.48°	5.82°	0.16 m	0.33 m

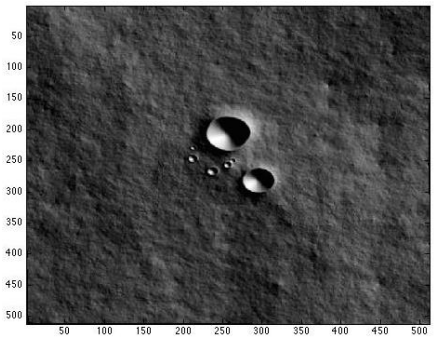


Figure 2.30: SfS input image.

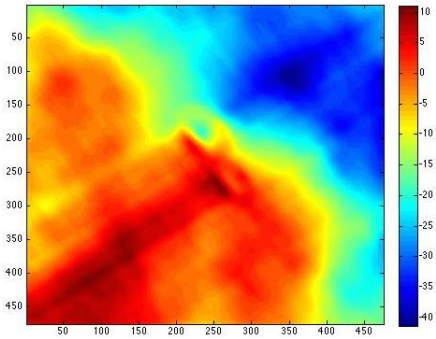


Figure 2.31: SfS DEM [m].

ness features cannot be resolved at these high altitudes. From the hazard-error map presented in Fig. 2.34 it can be concluded that SfS can be used to safely select a landing site for the given scenario. Next to this specific scene, also multiple others were analysed to draw conclusions concerning the performance of the SfS algorithm. It was found that the algorithm is very well capable of detecting the slope of the scene from various altitudes. Within the analysed altitude range (up to 2 km), SfS did not show any limitations (Woicke and Mooij, 2015), Table 2.7 summarizes these findings. During that study only the mean errors were investigated, since these already led to the clear conclusion that for the desired application stereo vision is the only feasible option. A detailed study was thus only conducted on the stereo results, which was presented in Sec. 2.4.1.

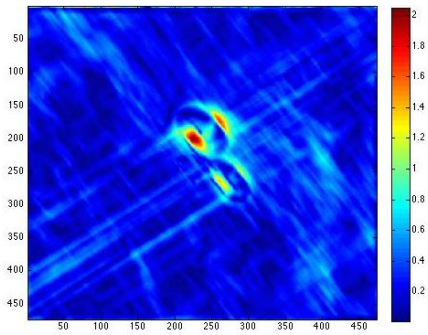


Figure 2.32: SfS roughness [m].

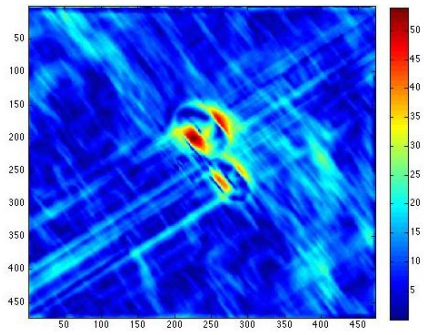


Figure 2.33: SfS slope [°].

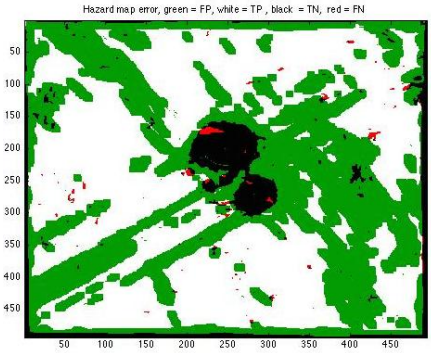


Figure 2.34: SfS hazard mapping error.

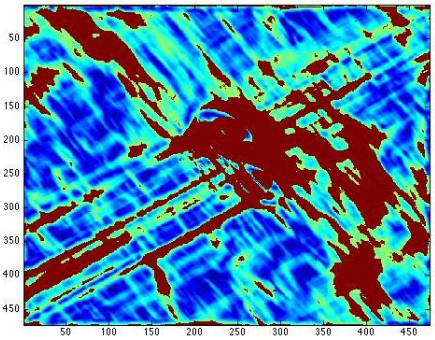


Figure 2.35: SfS hazard map.

Table 2.7: Mean errors for DEM, slope and roughness as presented in (Woicke and Mooij, 2015)

	SV			SfM			SfS		
Altitude [m]	μ_{DEM} [m]	μ_{slope} [°]	μ_{rgh} [m]	μ_{DEM} [m]	μ_{slope} [°]	μ_{rgh} [m]	μ_{DEM} [m]	μ_{slope} [°]	μ_{rgh} [m]
1000	9.60	52.19	1.15	9.97	4.48	0.06	7.86	2.41	0.03
500	7.03	27.73	0.61	2.68	10.22	0.18	2.40	1.40	0.02
150	0.24	2.63	0.05	0.61	7.76	0.22	3.35	6.89	0.10

2.4.3. SYNTHESIS OF DIFFERENT CAMERA BASED HAZARD-DETECTION OPTIONS

In the previous section some basic comparisons were done. The selection of the three different scenes already indicated that the three algorithms all have different strengths and weaknesses. One of the findings presented in Sec. 2.6 in more detail is that stereo vision does not perform well at higher altitudes. This can be deduced from looking at the minimum achievable depth resolution, δz, when using SV (Kytö et al., 2011):

$$\delta z = \frac{z^2}{bf}\delta d \tag{2.24}$$

This equation shows that the resolution scales with the square of the distance from the camera to the surface, z and improves if the minimum depth resolution, δd, is reduced. It underlines the finding that SV only performs well at lower altitudes and that its performance quickly degrades with increasing altitudes. Therefore, SV delivers good results at low altitudes, as will be shown in Sec. 2.6, but it also means that it is limited to low altitudes only.

Opposed to SV, the depth resolution of SfM is not directly linked to the distance to the surface. Still the depth resolution is linked to the pixel size, the warping and also the execution time. Opposed to SV, however, SfM requires accurate knowledge of the landing-vehicle's location, because otherwise both warping and downsampling will be done based on erroneous values. This will lead to errors in the final DEM, or even to a situation where reconstruction is impossible. Therefore, for implementation on a real

mission, SV will be easier to be included than an SfM algorithm. Moreover, SfM would require additional (potentially killer) constraints on the descent trajectory to move the epipole outside of the footprint; or at least away from the nominal landing site in the image centre. Both SV and SfM deliver good DEMs at the altitudes that are low enough to resolve boulders on the surface, with a size that could pose a potential danger for the lander. Without any modification to the trajectory or the method to overcome the problems with the epipole location, SfM cannot be used for hazard detection, since the epipole is located in the image centre, thus potentially covering the nominal landing site, as well as other landings sites, which could be reached with the least effort.

Lastly, SfS seems to have issues to actually compute the DEM, however, the resulting slope and roughness maps are very well usable for creating hazard maps. Shape from shading has the great benefit of being altitude independent. It can therefore be used very early in the descent as opposed to SfM and SV. Since it does recover the slopes very well it is a potential candidate for investigating hazardousness of landing sites based on slope only during early descent stages and switching to an algorithm more suitable for boulder detection, *i.e.*, SV, later in the descent. However, it does not perform well at detecting roughness and has therefore only limited applicability for the detection of boulders. Moreover Table 2.7 showed that SV outperforms SfS at lower altitudes.

As all three algorithms use the same kind of sensors, a combination of different algorithms for one mission would be possible. For example, one camera of a stereo pair could be used for SfS at high altitudes to detect slopes and as an SV system at low altitudes to detect boulders and slopes.

Concluding, it can be said that none of the presented passive hazard-detection algorithms clearly outperforms any of the other for all scenarios. However, SfM in its presents form is not a feasible candidate since both the constraints on the localisation accuracy and the epipole problem render it useless for the application at hand. Shape from shading can be used at high altitudes, while both SV and SfM cannot. On the other hand, both SV and SfM (apart from in the epipole region) can reconstruct the DEM more precisely than SfS, which helps with the detection of boulders. Therefore, it is recommended to base the selection of a passive hazard-detection algorithm strongly on the mission scenario. For a mission to an entirely unmapped body it would be advisable to use SfS to perform a first hazard mapping based on slopes only. At lower altitudes a second hazard mapping should be executed to detect boulders. This should be done with an SV algorithm. Since this work focuses on an application at low altitudes stereo vision is chosen for the final hazard-detection algorithm.

2.5. STEREO-VISION BASED HAZARD DETECTION

Based on the previous findings it was concluded that for the application at hand, mapping at very low altitudes with the desire to create high-quality high-resolution DEMs, stereo vision is the best candidate to perform this task. The stereo-vision algorithm was already presented in the previous section. This section will briefly summarise the finding from the previous sections and after that present the reference scenario that will be used for the elaborate sensitivity study of the hazard detection algorithm.

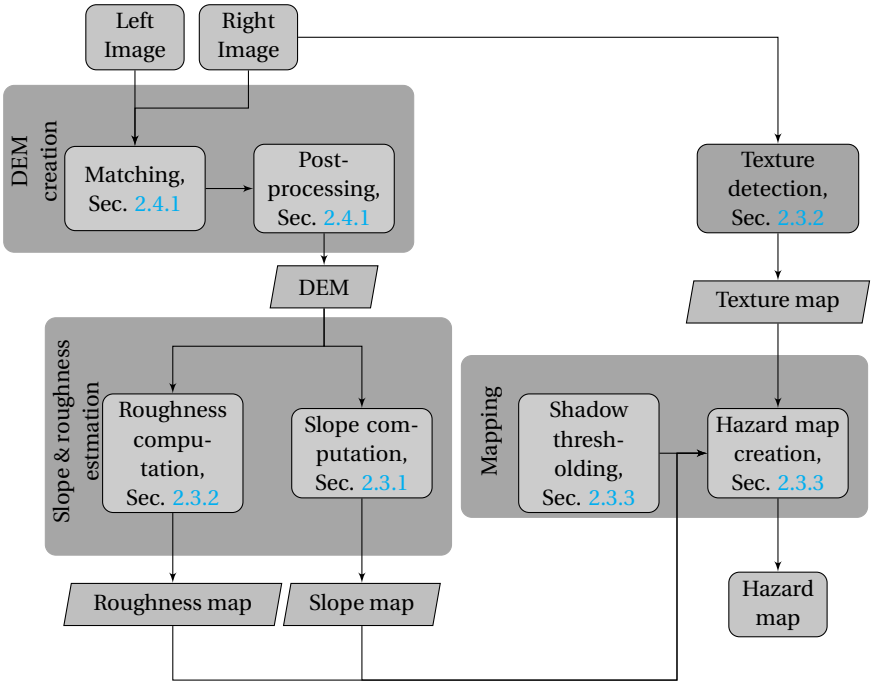


Figure 2.36: HDA software architecture.

2.5.1. SUMMARY OF THE DEVELOPED METHOD

To summarise, the hazard-detection algorithm used in the next sections and chapters consist of stereo-vision methods that use the SSD over a 7×7 square window for image matching. A parabolic fit is used to achieve non-integer disparities. Once the DEM has been created it is filtered using both a linear-regression and a median filter. Slope and roughness are computed based on the resulting DEM using a least-squares plane-fit. The size of the plane is determined by the lander footprint. Next to the mean-plane roughness, the algorithm also makes use of texture detection for detecting roughness, this is done using histogram-based variance. Additionally, image intensity is assessed to identify shadowed regions. Figure 2.36 presents the flow diagram of the algorithm. The block post-processing summarises the filtering of the DEM as described in Sec. 2.5.

Stereo vision is a well-understood concept, which is frequently used in robotics and computer vision (*e.g.*, (Hartley and Zisserman, 2004) and (Maimone et al., 2006)). Due to this very extensive heritage in other fields it is an interesting algorithm to use, as it is a well documented and understood. Nevertheless, the performance and feasibility of the method for the use in a hazard detection systems is not proven, yet. This proof is part of this thesis and will be presented in the following sections.

Using an image pair obtained from a stereo set-up as input, it is possible to reconstruct all necessary maps to create a full hazard map. First, the DEM is reconstructed from the stereo input images. Second, the slope and roughness maps are computed based on the resulting DEM. Third, from the input images a texture map is created,

which serves as an additional roughness measure. Fourth, based on the requirements stated above and the computed maps, a combined hazard map is composed. Two versions of the algorithms were developed, a MATLAB prototype and a C++ prototype version. All results presented in the following were obtained using the prototype version, both versions were verified to produce identical outputs and only differ in terms of execution time.

2.5.2. REFERENCE MISSION

The reference mission was defined as performing one hazard survey at an altitude of 150 m with a stereo baseline of 2 m. For comparison, the design of the ESA Lunar Lander has a footprint diameter of 5.6 m and a body diameter of 2.6 m (Carpenter et al., 2012), see also Fig. 2.37. The NASA MSL rover is reported to have body dimensions of 3 m × 2.7 m × 2.2 m (Way et al., 2007). Figure 2.38 shows the rover next to a human. Considering that it is most desirable to attach a camera setup to stiff parts of the body, the feasible maximum baseline seems to be in the order of 2.5 m to 3 m. Even though the footprint diameter of the ESA Lunar Lander is almost double of this, attaching cameras to the landing legs is not a wise choice as vibrations and movements in the legs will influence camera orientation and position and thus negatively influence the results. Even with the current setup, *i.e.*, cameras attached to the body, vibrations might still cause problems. Concluding, a baseline of 2 m can be achieved on a lander comparable to MSL or the ESA Lunar Lander.

The imaging altitude of 150 m was chosen, as this value falls well within the operational envelope of the algorithm (≤ 200 m, see Sec. 2.6 for a detailed discussion) and is representative for the imaging altitudes of other mission studies, which usually intend to perform HDA at altitudes of 100 m to 200 m, see, *e.g.*, (Xiong et al., 2013) and (Huertas et al., 2007). Encountering a scene consisting of 15 % to 20 % hazards was identified as a realistic landing scenario. Due to the method of creating the scenes (as described in Sec. 2.2), it is not possible to create a certain percentage of hazards. Therefore the scene was created by trial and error, leading to 18.3 % hazards. Variations of the initial parameters, *i.e.*, baseline, altitude, and hazard percentage, and their influence on the results are discussed in the sensitivity analysis presented in the next section. An overview of the reference-mission inputs is given in Table 2.8. It should be noted that a rather small camera was chosen, as the computational effort scales with the image size. Therefore, smaller images result in faster execution times. The current setup was found to be a good compromise between execution time, resolution and ground-patch size.

The stereo-image pair of the terrain surveyed is shown in Fig. 2.39. The image contains multiple smaller craters, as well as part of a larger crater in the bottom half of the image. Furthermore, three boulders of different sizes and shapes are distributed over the scene. The altitude of the scene varies from −13.5 m to 0.0 m.

It is possible to evaluate the quality of the final hazard map by comparing it to the true hazard map computed from the ground-truth maps. The results of the comparison can be grouped in the following four different classes: 1) False negative (FN) a hazard, undetected by the algorithm (wrong detection); 2) False positive (FP) defines a safe site, which is erroneously labelled as unsafe (false alarm); 3) True negative (TN) is a correctly detected hazard; and 4) True positive (TP) is a correctly detected safe site.

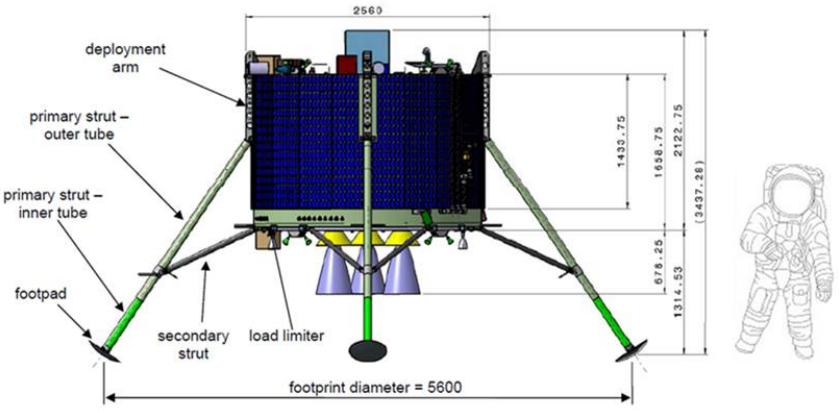


Figure 2.37: ESA Lunar Lander configuration (Carpenter et al., 2012).

Figure 2.38: MSL configuration with human for scale (NASA).

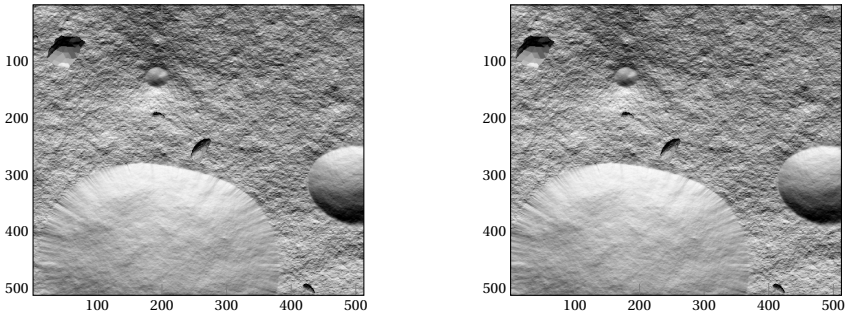


Figure 2.39: Reference-mission image pair.

Table 2.8: Reference-mission inputs.

Parameter	Value
Altitude, m	150
FOV, deg	30
Baseline, m	2
Max altitude, m	0.0
Min altitude, m	−13.5
Image size, pixels	512 × 512
Resolution x, y, m	0.16
Size of ground patch, m×m, m^2	82 × 82 = 6724
Hazards in scene, %	18.3

Table 2.9: DEM, slope, and roughness mean errors and standard deviations, altitude 150 m, reference mission.

FN, %	0.70
FP, %	29.42
TN, %	17.62
TP, %	52.26
$\mu_{DEM,I}$, m [1]	0.49
$\sigma_{DEM,I}$, m [1]	0.46
$\mu_{DEM,II}$, m [2]	0.47
$\sigma_{DEM,II}$, m [2]	0.40
μ_{Slp}, deg	4.87
σ_{Slp}, deg	5.00
μ_{Rgh}, m	0.13
σ_{Rgh}, m	0.20

[1] without DEM filtering.
[2] with DEM filtering.

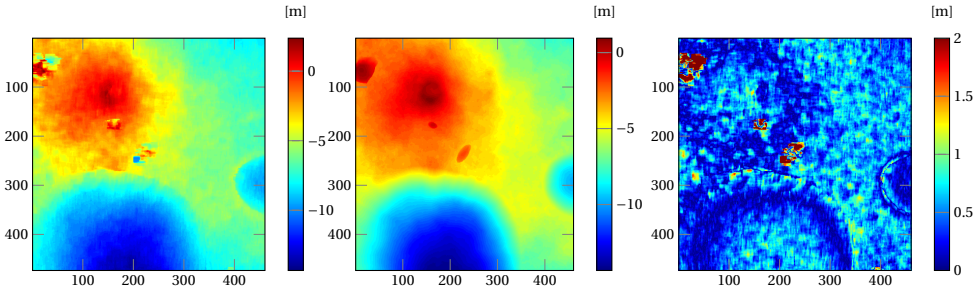
The full set of DEM, slope, and roughness errors, as well as the detection probabilities, FP, FN, TN, and TP, are given in Table 2.9. Most importantly, one should note that the FN percentage is below the allowable maximum of 1%. Furthermore, more than 70% of the scene is reconstructed correctly, which is assumed to be sufficient for selecting a safe landing site. It can be seen that both the mean and the standard deviation for slope and roughness are less than half of the values that would indicate a hazard (15 deg and 0.5 m, respectively). This clearly shows that the detection of hazards is possible. Also, it can be seen that filtering the DEM reduces the DEM error as expected (see Table 2.9). It also improves all other results. The FN and FP percentages would be 0.75% and 29.9%, respectively, without filtering. Also μ_{Slp}, σ_{Slp}, μ_{Rgh}, and σ_{Rgh} are higher without filtering namely 5.03 deg, 6.17 deg, 0.15 m, and 0.96 m, respectively. Analysing the number of correctly detected hazards, it was found that out of all hazards, 96.2% are correctly detected. There was no requirement posed on this number, however, this result clearly shows that indeed the algorithm performs well.

Figures 2.40a to 2.43 show the computed DEM, slope, and roughness map, as well as the ground-truth and the resulting error maps. Figure 2.40a shows the DEM. Two main observations can be made in the figures. First, the terrain elevation gradients seem to be a bit noisy. This is due to the parabolic fit, as it will introduce small disparity errors. This has been described and discussed by Shimizu and Okutomi (2005). Second, there

are some errors in the rather flat plane in the bottom region of the DEM. These errors are due to the parabolic fit for estimating sub-pixel disparities, as in few cases the filtering will amplify smaller errors caused by the fitting. Apart from these small errors, the maps are reconstructed well. The craters in the scene are all detected and so are the boulders. The terrain elevations are neither under- nor overestimated.

Investigating the slope and roughness map in Figs. 2.41a to 2.42c, respectively, it can be seen that these errors from the DEM are propagated to the slope and roughness maps. However, no new errors are introduced. Furthermore, all slopes and roughness in the craters are reconstructed. Also the boulders are detected. However, some of the large slopes and roughness regions are slightly overestimated by the algorithm. In general, overestimation is preferable to underestimation, because it can be seen as a safety factor. Figure 2.43 shows the texture map. As desired, the crater rims and the boulders in the scene are detected.

Figure 2.44 shows the hazard-error map. It indicates the locations of FP, FN, TP and TN errors. All FN errors (wrong detections, dark gray) are always surrounded by or adjacent to TP regions (correctly detected hazards, black). The largest number of FP errors (false alarms) is linked to the overestimation of slopes.



(a) Computed. (b) True. (c) Error.

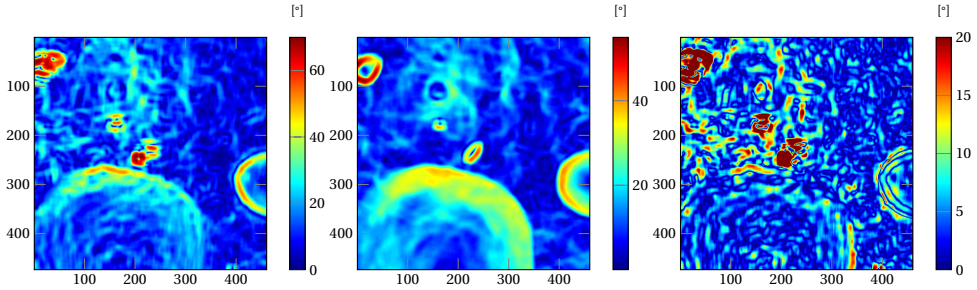
Figure 2.40: DEM (reference mission)

Based on the requirements stated previously, the execution time shall be below 2 s. The hazard detection for this scene took 1.1 s, when executed on a 2.3 GHz Intel i7 ivy bridge processor. Therefore, the algorithm satisfies this requirement.

2.6. SENSITIVITY ANALYSIS OF STEREO-VISION HAZARD-DETECTION METHOD

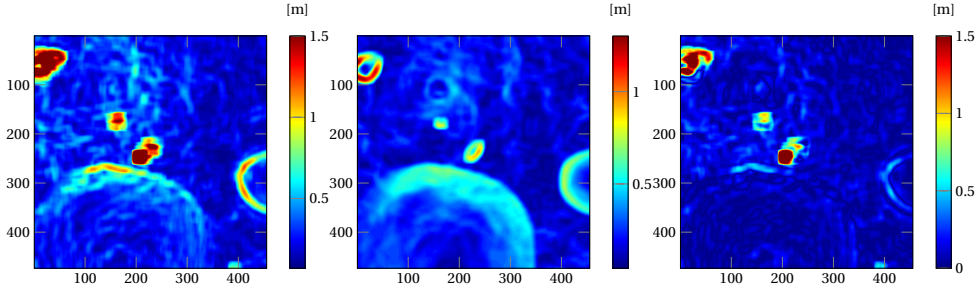
The purpose of this sensitivity analysis is to determine the limitations of the developed algorithm, as well as to determine the optimal combination of altitude and baseline to be used for hazard detection. To judge the algorithms' capabilities concerning site complexity, five different scenes are used.

Scene 1 contains two larger craters on an otherwise rather flat terrain. In addition, multiple boulders of different sizes and shapes are present in this scene. It is less com-



(a) Computed. (b) True. (c) Error.

Figure 2.41: Slope (reference mission).



(a) Computed. (b) True. (c) Error.

Figure 2.42: Roughness (reference mission).

plex when viewed from higher altitudes, but gets more complex the lower the spacecraft goes as it will move towards the centre of the crater and because more boulders appear with increasing resolution. At higher altitudes, approximately 10% of the scene is hazardous. At lower altitudes this increases up to a maximum of 68% at an altitude of 50 m. Figure 2.46a shows this terrain viewed from an altitude of 400 m.

Scene 2 contains multiple craters of different sizes, but no boulders on a mostly level terrain. This scene has a steady rate of ≈ 10% hazards in the scene. In Fig. 2.46b an image of the surface taken at an altitude of 400 m is presented.

Scene 3 contains a large number of boulders of various sizes, shapes and terrain elevations. The maximum terrain elevation difference is only in the range of 4 m, therefore this scene is safe in terms of slope hazards. The hazardousness of this landing region is around 10% for all altitudes. In Fig. 2.46c an image of the surface taken at an altitude of 400 m is given.

Scene 4 contains multiple craters and boulders on a more sloped and more complex terrain. The maximum terrain elevation difference is approximately 20 m. The scene is 10% hazardous at high altitudes, which increases to 25% at lower altitudes. Figure 2.46d

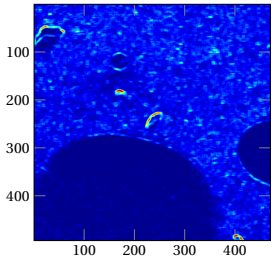


Figure 2.43: Texture map (reference mission).

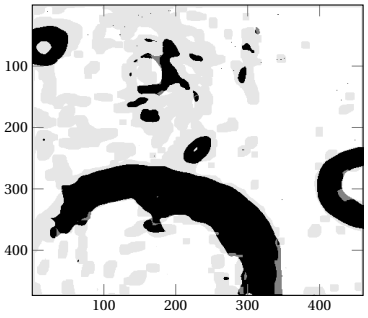


Figure 2.44: Hazard mapping errors, darkgray = FN, white = TP, lightgray = FP, black = TN.

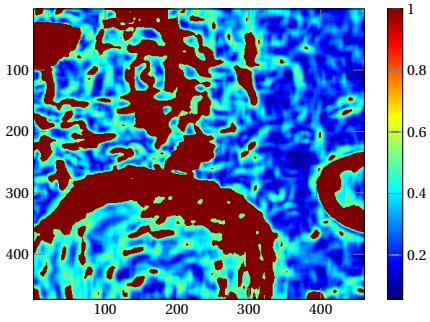


Figure 2.45: Scaled hazard map (reference mission).

gives an image of the surface taken at an altitude of 400 m.

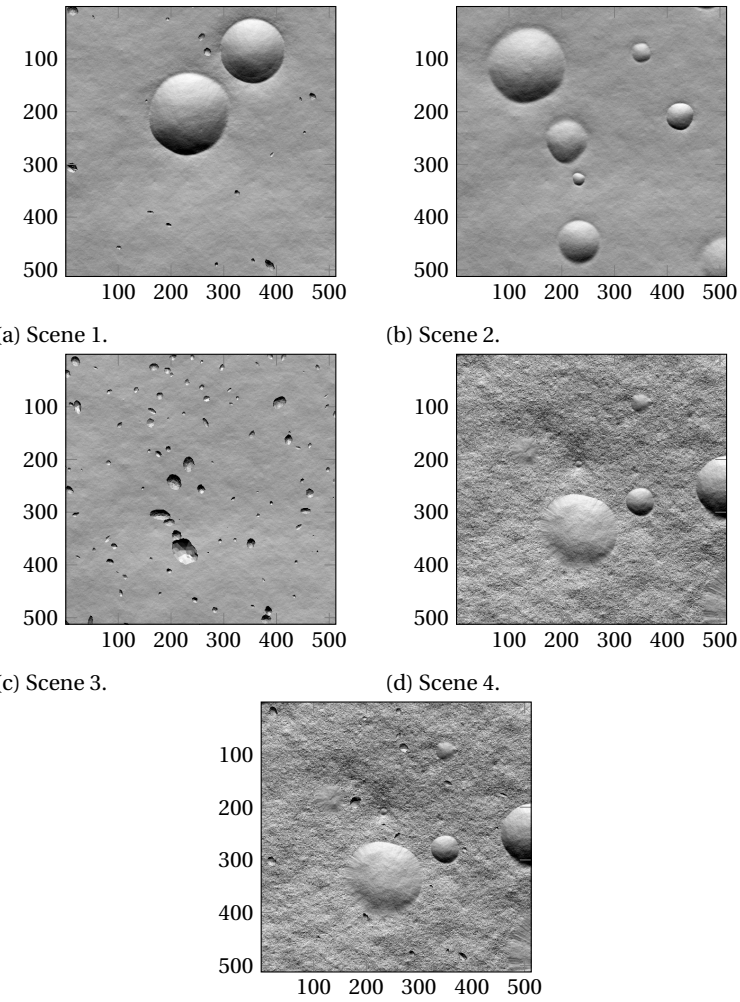
Scene 5 is a copy of scene 4; however, with the addition of multiple boulders. Due to the added boulders the maximum hazardousness increases to 30%. In Fig. 2.46e an image of the surface taken at an altitude of 400 m is presented.

If the imaging altitude is lowered, the cameras are moved towards the centre of the image, thus the x- and y-coordinates of the camera are kept fixed.

BASELINE AND ALTITUDE

This section discusses the results of the sensitivity analysis based on the scenes as presented earlier. To limit the length of this section the resulting graphs of the sensitivity analysis for two of these scenes are presented. Scene 2 and Scene 5 are selected, representing the simplest and most complex scene used for the analysis, respectively. The results for the remaining scenes are still discussed, but their results are not presented in graphical form.

To assess the performance of the algorithm four main aspects are analysed. These are: 1) the percentage of undetected hazards (FN) out of all possible landing sites as discussed in Section 2.5; 2) the percentage of detected hazards out of all hazards TN / (TN + FN). 3) the total percentage of correct detections (TN+TP); being the sum of all correctly



(a) Scene 1.

(b) Scene 2.

(c) Scene 3.

(d) Scene 4.

(e) Scene 5.

Figure 2.46: Scenes used during sensitivity analysis.

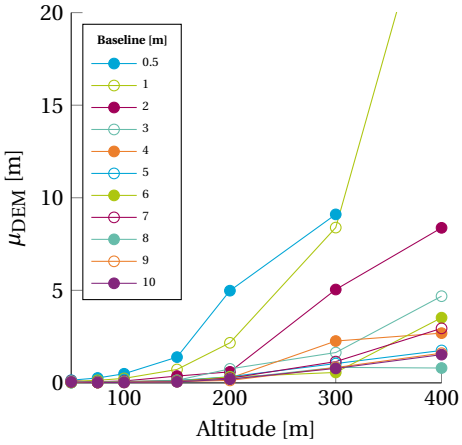


Figure 2.47: DEM error Scene 2.

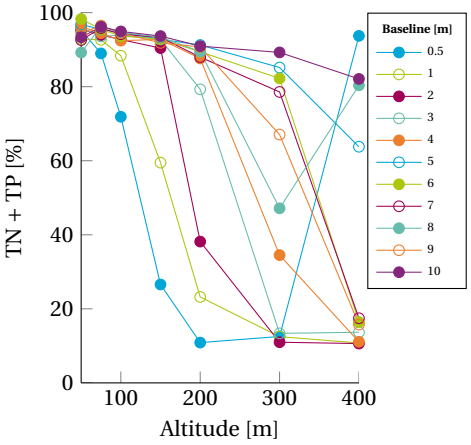


Figure 2.48: Correct detections (TP + TN), Scene 2.

detected hazards and correctly detected safe sites; 4) the mean DEM error. The second percentage under investigation, *i.e.*, the number of correctly detected hazards out of all hazards, is investigated to account for the fact that the different scenes have different hazard-levels. By comparing TN/(TN+FN), one can compare algorithm performance independent of these.

The algorithm is executed on these five different scenes, for eleven different baselines (0.5 m, 1 m, 2 m, 3 m, 4 m, 5 m, 6 m, 7 m, 8 m, 9 m and 10 m), and seven different altitudes above the ground (50 m, 75 m, 100 m, 150 m, 200 m, 300 m and 400 m), which results in a total of 385 cases.

In Fig. 2.47 the mean DEM error for Scene 2 is presented. Here, it can be seen that for baselines greater than or equal to 2 m the DEM error stays below 1 m up to 200 m altitude. At lower altitudes, the error decreases even further for all baselines. At 100 m altitude, the error drops below 0.5 m for all baselines larger than 0.5 m.

Fig. 2.48 shows the total percentage of correct detections (TN +TP). Obviously, this number should be maximised. For baselines larger than 1 m the correct detections are above 90% up to altitudes of 150 m. For all baselines but 2 m it stays at this high level for 200 m as well, however, for a 2 m baseline the number of correct detections drops. Still, it reaches 40%, which would be sufficient to select a landing site. This drop can be explained by a rather large number of false alarms, which is caused by the increasing noisiness of the DEMs. Based on this plot it can be concluded that at altitudes of 150 m and lower even baselines of 1 m might be feasible. In the next figure, Fig. 2.49, the percentage of undetected hazards (FN) is given. Up to 200 m altitude, all results stay below the required 1%. Since FN is a measure of how many pixels represent undetected hazards out of all pixels in the scene, it should also be investigated how many hazards out of all hazards are correctly detected. The results of this analysis are given in Fig. 2.50. From this graph it can be concluded that on average, above 90% of all hazards are correctly identified. Moreover, investigating the resulting hazard maps, it can be seen that the undetected hazards are always adjacent to a detected hazard or even surrounded by

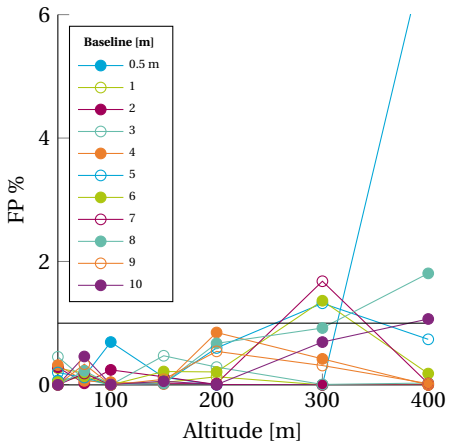


Figure 2.49: FN scene 2.

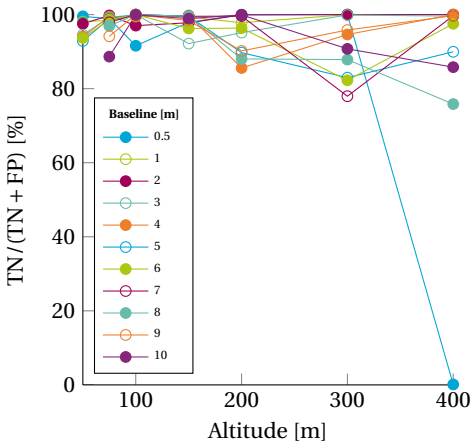


Figure 2.50: Correctly detected hazards, Scene 2.

detected hazards. Concluding, based on the analysis of this scene it seems that the algorithm can perform as desired for a baseline of 2 m, which is feasible for state-of-the-art landers, at altitudes of 200 m and lower.

The same graphs are also generated for Scene 5. Figure 2.51 plots the DEM mean error. Comparing this graph to the graph of Scene 2 (Fig. 2.47), it is apparent that the algorithm performs worse on this scene than on Scene 2. This is due to the increased complexity of this scene. Still, for altitudes of 200 m and lower, the DEM error stays in the order of 1 m and lower. Also, in the total number of correct detections (Fig. 2.52), the increased complexity of the scene reappears. Yet, for baselines of 2 m and more and altitudes of 200 m and less, the correct detections stay well above 40% and will therefore always allow for the selection of a safe landing site. Again, the decrease in correct detections is caused by an increase in false alarms and not by undetected hazards, as can be concluded from analysing the following two figures. Figure 2.53 shows the percentage of undetected hazards for Scene 5. Comparing this figure to Fig. 2.49, the same graph for the less complex scene, it can be concluded that the algorithm has more problems recovering the hazards for complex scenes than for easy terrain. Nevertheless, the FN percentage stays below 1% up to 150 m altitude. Even though, at 200 m altitude this error is very close to 1%, it is not below 1% for all baselines. However, as discussed for Scene 2, the FN errors are always directly next to or inside a correctly detected hazardous region. Lastly, Fig. 2.54 presents the total number of correctly detected hazards out of all present hazards. Here, the algorithm detects above 90% of all hazards correctly up to altitudes of 200 m. From the graphs discussed, it can be concluded that added complexity is reflected in the resulting hazard maps. Yet, hazard mapping is possible up to altitudes of 200 m with baselines of 2 m and above.

From the previous analysis it can be concluded that hazard mapping is possible with a 2 m baseline. Therefore, the DEM mean error and the FN percentage for all five scenes at baselines of 2 m are presented in Figs. 2.55 and 2.56, respectively. Here it can be seen, that apart from the previously discussed Scene 5, all other scenes have FN percentages

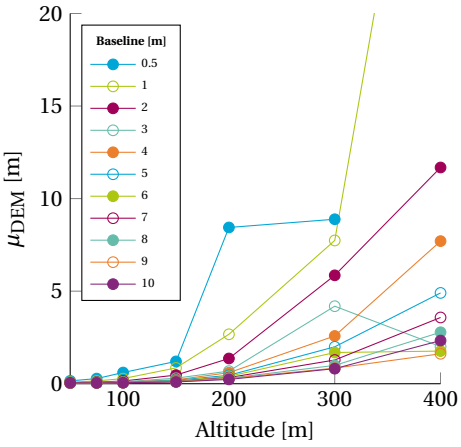


Figure 2.51: DEM error Scene 5.

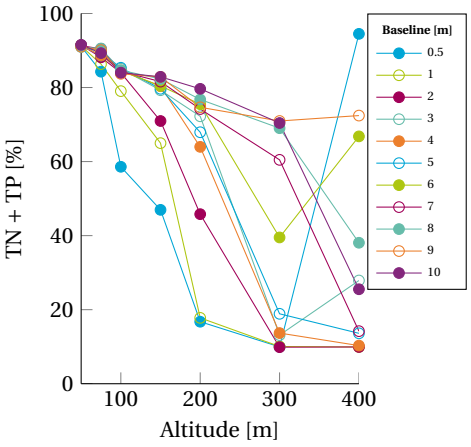


Figure 2.52: Correct detections (TP + TN), Scene 5.

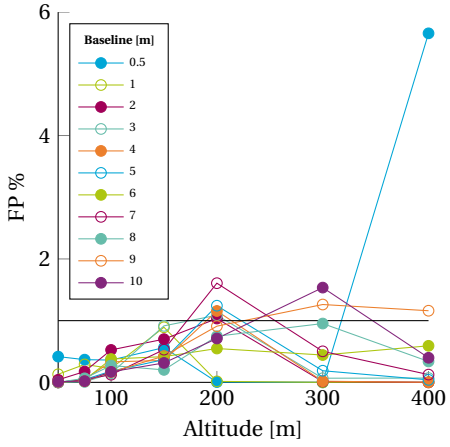


Figure 2.53: FN scene 5.

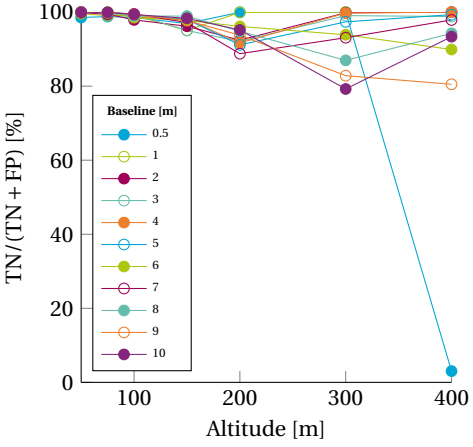


Figure 2.54: Correctly detected hazards, Scene 5.

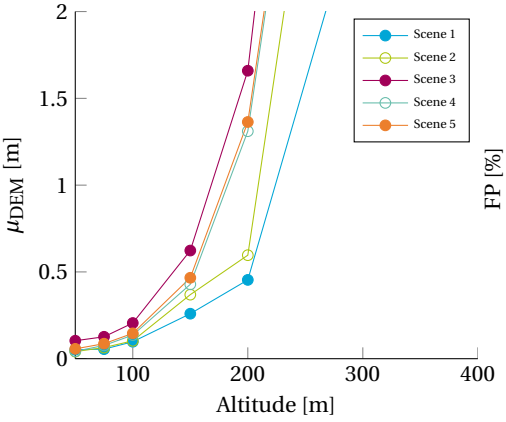


Figure 2.55: Mean DEM error at 2 m baseline.

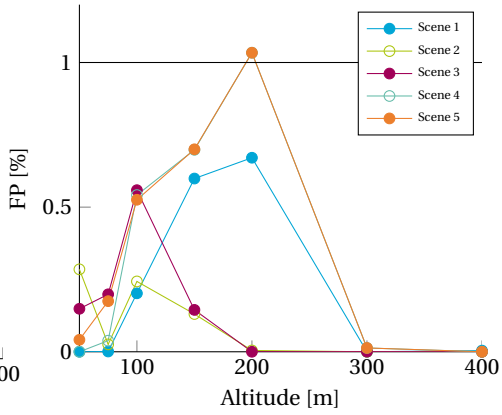


Figure 2.56: FN error at 2 m baseline.

below the required 1%. Therefore the previously stated conclusion that Scene 5 is an outlier, as it is the most complex one, is supported. Based on FN percentage only, mapping should be possible at even higher altitudes. However, investigating the DEM error presented in Fig. 2.55, it is clear that at altitudes above 200 m the DEM errors get too large. The investigation of the remaining results for Scene 1, Scene 3, and Scene 4, support this finding. Therefore, for the given requirements, it can be concluded that the algorithms can be used at altitudes smaller than or equal to 200 m with a baseline of 2 m. Using larger baselines mapping at higher altitudes is achievable. The use of smaller baselines is possible if lower mapping altitudes are used.

BOULDERS

In addition to investigating how good the DEMs and from those the overall roughness, slope, and hazard maps are, it is also important to test how well rocks and boulders are detected, as these pose the main hazards for landings on mapped bodies, such as Mars and the Moon. This is done by creating a scene with rocks and boulders, counting the features detected by the texture detection, the roughness detection, and comparing the detected number to the number of rocks and boulders present in the input image. Figures 2.57 to 2.59 show one set of input image, texture hazard map, and roughness hazard map. In the input image, 19 roughness features are present (overlapping features are counted as one). In the texture hazard map, Fig. 2.58, all 19 features are identified, while in the roughness hazard map, Fig. 2.59, nine of these rocks are identified. As discussed in Section 2.5.2, the texture detection extracts more features than actually present, as it also detects small craters in the PANGU terrain texture. This can clearly be seen in the texture hazard-map, Fig. 2.58.

The same test is repeated for 50 m, 75 m, 100 m, 150 m, 200 m, 300 m, and 400 m altitude. The results are represented in a bar chart in Fig. 2.60. It can be concluded that texture detection always detects all features, even at high altitudes. The roughness estimation, however, does not work as well, but still detects some of the features. Due to the fact that roughness estimation can determine the size of roughness features, while tex-

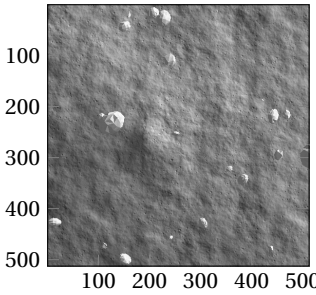


Figure 2.57: Input image.

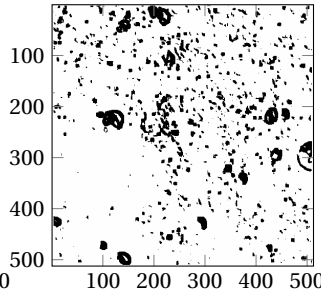


Figure 2.58: Texture hazards.

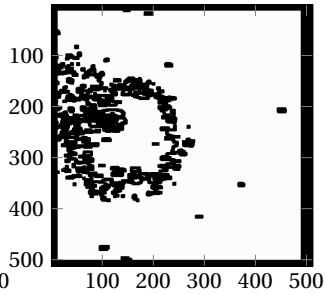


Figure 2.59: Roughness hazards.

ture detection cannot, it is still very much advisable to combine both algorithms. Overall, this shows that the algorithm is capable of detecting the rocks present in the scene.

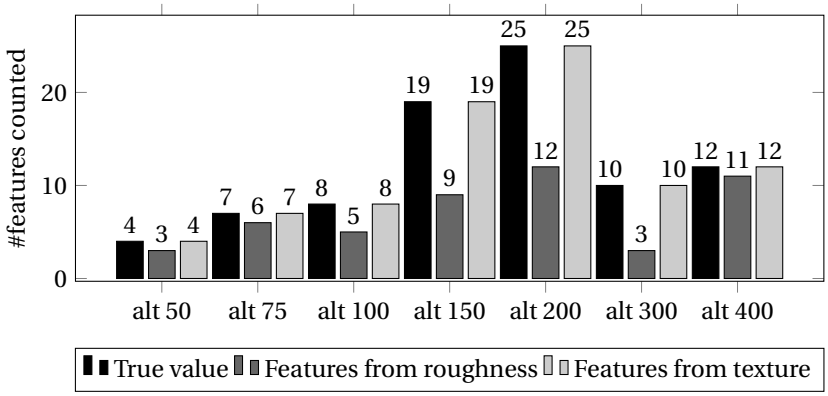


Figure 2.60: Feature counts at different altitudes. Counts vary per altitude.

2.7. NEXT STEPS IN HAZARD DETECTION

To date, in all (design) studies implementing HDA, hazard detection is only employed once or twice during a descent. Also the only flown mission that implemented HDA, ChangE'3, did only perform a single hazard-detection run. It is not used in a more *continuous* fashion. Moreover, the information retrieved during the mapping phase is neither used during later mapping phases, nor are the maps used to construct a large surface map on the go. To achieve this, approaches comparable to simultaneous localisation and mapping (SLAM) as employed in the robotics community for simultaneous map building of unknown environments and localisation within these maps, might be explored. Full SLAM, *i.e.*, building full maps, is too large a burden on the unfortunately not very powerful on-board computers in use on spacecraft today. However tracking some surface points in a SLAM-like manner might be a first stepping-stone for the re-utilisation of hazards maps for rover navigation. The SLAM idea will be followed in the remainder of this work.

It is very important to note that the intention to include an HDA system will have implications for the entire descent system. Since HDA will require the lander to execute a landing at, more or less, an exact location to avoid hazards while landing, the lander has to be able to perform a precision landing. To date, no mission demonstrated such a capability.

A precision landing describes a landing with a very small landing ellipse. Here, small means a couple or at most some tens of meters. Currently, landing ellipses range in the order of multiple kms. This is, for example, caused by inaccuracies in state estimation, environmental models, initial conditions and insufficient guidance. It is therefore needed to improve the contributions that cause errors, *e.g.*, the state estimation.

To improve state estimation, two main technologies are hazard/terrain relative navigation (HRN/TRN) as well as terrain absolute navigation (TAN). Both are methods used to improve the vehicle state by improving the inertial measurement unit (IMU) based state using information derived from surface features. Without improving on current navigation capabilities performing HDA will not be possible.

Therefore, the main focus of the second part of this work is that maps generated for hazard detection can also be used as an input for hazard-relative navigation during the final landing phase. HRN/TRN frequently makes use of DEM or camera images, and it would be highly beneficial if the same maps could be used for both HDA and TRN. This approach will be discussed in more detail in the next chapter.

Next to improving the landing accuracy, HDA will also require a guidance algorithm that can lead the vehicle towards a chosen landing site using trajectories computed on-board. Obviously, this is not achievable using a simple gravity-turn guidance, because that is open loop. However, guidance laws capable of this are available, for example, the Apollo lunar lander guidance (E-Guidance). Gerth and Mooij (2014) give an overview of possible guidance laws and their applicability with respect to missions that include HDA and/or precision landings. Developing new guidance strategies for hazard avoidance will not be addressed in this work, as this was already thoroughly covered and strategies usable in an HDA scenario are available.

Concluding, all of the above-mentioned systems are necessary to make use of the information HDA can provide. This means that these have to be developed to sufficient technology readiness level before HDA becomes feasible for implementation in future missions. The work presented in the next chapters will thus greatly contribute to reaching that goal.

2.8. SUMMARY AND OUTLOOK

Based on a trade-off of three different camera-based hazard-detection techniques, stereo vision, stereo-from-motion, and shape-from-shading, stereo vision was selected as the best feasible candidate for autonomous, real-time, on-board hazard detection. In this chapter, the development of such a stereo-vision hazard-detection algorithm was presented, including the design considerations made and motivating the final choices. It was shown that including a quadratic fit to estimate sub-pixel disparities will greatly increase the application range of such an algorithm. It was concluded that performing hazard mapping at altitudes of 200 m and below is feasible for baselines of 2 m. For lower altitudes, even lower baselines in the order of 0.5 m to 1.0 m can be used. This

is especially interesting for miniaturised spacecraft, which are too small for attaching a stereo set-up with a baseline of 2 m. If mapping is done at altitudes of, or lower than, 200 m, the number of wrong detections is always below 1%. Moreover, given that the number of correct detections is always above 60%, this leads to the conclusion that the selection of a hazard-free landing site should always be possible. For all performed test cases, the execution time stayed below the required maximum of 2 s. Moreover, it was found that boulders and rocks can be detected very reliably when using a combination of texture detection and roughness as the deviation from the mean plane. Summarising, this proves that the presented algorithm is a suitable candidate for hazard detection to increase the autonomy and decrease the risk of a landing failure for future exploration missions.

In the next chapter a navigation method capable of navigation relative to the hazard maps resulting from the HD function in this chapter is presented. This method is more precise and more accurate than the current state-of-the-art navigation and can thus enable the inclusion of HD in the landing GNC loop.

3

HAZARD-RELATIVE NAVIGATION

CURRENTLY, navigation during the final phase of a descent relies on rather inaccurate sensors and sensors which cannot resolve the full state, *i.e.*, IMUs and altimeters, for tracking the lander's position. This is one of the reasons that landings are still inaccurate. With the current accuracies, a precision landing, a requirement for many future missions, and an absolute necessity for a landing system including an HDA system, cannot be achieved. Therefore, more advanced navigation methods need to be employed. One of these systems is a terrain relative navigation system. Here, camera images or DEMs are used to track the landers movement relative to surface features. In this work, a hazard-relative navigation solution is developed. This can be seen as a sub-class of TRN, where navigation is not simply relative to surface features, but relative to the on-board hazard maps. This is a necessary asset for a landing system involving HDA, because only by this approach it is possible to ensure avoiding the detected hazards.

This chapter first discusses the basic principles of navigation and state estimation in Sec. 3.1, and of terrain- and hazard-relative navigation in Sec. 3.2. After this introduction to the topic, the principle of simultaneous localisation and mapping (SLAM) is introduced in Sec. 3.3. Next, the design of the filter used in this work is described in Sec. 3.4. Sections 3.4.2 to 3.4.5 outline the four elements of the filter, the propagation, the measurement, the augmentation of the state based on the measurement, and the state update based on the measurement. After successful design of the filter, it needs to be tuned such that it shows the desired behaviour. Tuning and tuning outputs are presented in Sec. 3.5. To test the algorithm's performance, a reference scenario is setup, which is presented in Sec. 3.6. The software-in-the-loop testing of the HRN function is presented in Sec. 3.7. The chapter closes by summarising the highlights of the HRN method and its performance in Sec. 3.8.

Parts of this chapter have been published in Advances in Aerospace Guidance, Navigation and Control (Woicke and Mooij, 2018).

3.1. NAVIGATION AND STATE ESTIMATION

To understand the context of hazard-relative navigation, the concept of navigation and thus state estimation needs to be briefly introduced. Navigation is a part of the GNC system, tasked with determining the lander's state, $\boldsymbol{x}$. The state may contain many elements, for example, the position, velocity and orientation of the vehicle. As will be discussed later, other elements might be added to the state if this will improve the state estimation. Only by knowing the current lander state, it is possible to successfully perform a landing at a desired final location. The more accurate the state knowledge is, the more accurate the lander can land. In the presence of large state errors, a precision landing is not possible.

Different methods to determine the state exist, however the most common are Kalman filtering techniques. In this work, a Kalman filter approach is used, therefore a discussion of other state estimation methodologies is omitted. A Kalman filter estimates the current state $\hat{\boldsymbol{x}}_i$, based on previous estimate of the state, $\hat{\boldsymbol{x}}_{i-1}$, some inputs, $\boldsymbol{u}$, *e.g.*, the commanded thrust and measurements from an IMU, and measurements, $\boldsymbol{z}$, *e.g.*, from the global positioning system (GPS), a star tracker, or images. Moreover, the Kalman filter also estimates the uncertainty of the state estimate. These uncertainties are described by the error covariance matrix, $\boldsymbol{P}$.

The true state at time i, can be described by the following equation as a function of the state at the previous time step $i-1$, see for example (Musoff and Zarchan, 2009).

$$\boldsymbol{x}_i = \boldsymbol{F}_i \boldsymbol{x}_{i-1} + \boldsymbol{B}_i \boldsymbol{u}_i + \boldsymbol{w}_i \tag{3.1}$$

where $\boldsymbol{F}$ is the system matrix, $\boldsymbol{B}$ is a control matrix, $\boldsymbol{u}_i$ the control vector and $\boldsymbol{w}_i$ is the process noise. Note, that not every system always has a control matrix and a control vector if there are no (known) control inputs.

The measurement at time i, $\boldsymbol{z}_i$ used to update the measurement is described by

$$\boldsymbol{z}_i = \boldsymbol{H}_i \boldsymbol{x}_i + \boldsymbol{v}_i \tag{3.2}$$

where $\boldsymbol{H}_i$ is the measurement matrix and v_i is the measurement noise, see for example (Musoff and Zarchan, 2009).

The basic form of the Kalman filter (KF) only works for linear systems. For these systems the KF will guarantee both optimality and convergence. If a system has non-linear dynamics or measurement models, the basic formulation of the KF cannot be used. Multiple methods exist to use non-linear systems in a KF, but, for example, the most common one, the extended Kalman filter (EKF) is not able to guarantee either convergence or optimality.

In general, there are two approaches to perform Kalman filtering, the direct and the indirect approach. The direct approach is also called "dynamic modelling approach" and the indirect approach "strap-down modelling" approach. The main difference is that the direct approach estimates the full state, while the indirect approach estimates the errors in the state. The classical EKF is a direct approach, while methods such as the Error State Kalman Filter (ESKF) and the Multiplicative Extended Kalman Filter (MEKF) are indirect methods (Madyastha et al., 2011).

A Kalman filter always follows the same steps, only the detailed implementation of these steps differ per approach. A filter first **propagates** the currents state based on a

model of the dynamics. This model may be purely based on equations but can also use additional information, such as the IMU measurements or control commands. This is done in the **prediction step**. During prediction, errors are accumulated (since the model is never 100% accurate, as otherwise there would not be any need for a KF). After a measurement is recorded, the **update step** is executed. In this step the state prediction is updated based on the measurement. To this end a measurement prediction is computed based on the predicted, erroneous, state and a measurement model and compared to the actual measurement. Like the state model, the measurement contains errors. These updates make the state prediction more accurate and decrease the error in the system.

3.2. BASIC PRINCIPLES OF TERRRAIN RELATIVE NAVIGATION

If a lander navigates using IMU measurements only, without using any form of KF, this is called dead reckoning. In this mode, the localisation error will simply accumulate and may even grow over time. With the addition of other sensors, for example, an altimeter, the localisation result can be updated with this measurement and a (E)KF can be used instead of dead reckoning. This assumes that the combination of the current knowledge and the sensor measurement will result in a more accurate state knowledge. Including an altimeter next to the IMU will still lead to rather inaccurate state predictions, especially in x- and y-direction,[1] as a (radar) altimeter will only measure the line-of-sight distance. It is therefore, for example, used to guide and detect touchdown. Other sensors do exist, but most are not usable during the final phase of a descent. Star sensors, for example, cannot operate under the strong vibrations of a powered descent.

However, when a more precise touchdown is desired, especially (x, y) localisation accuracy is very important. But there is no sensor that can simply take a measurement that could be used to estimate these two state elements more accurately. Therefore, terrain-relative navigation systems were developed. For these systems, surface images or 3D maps of the surface are used to estimate translations, velocities and orientations with respect to the surface[2].

Here, two different approaches can be used: either features or landmarks are matched against images obtained prior to the mission, or they are matched against features identified in an image captured earlier in the descent. The first approach is called terrain absolute navigation in this work (see Chapter 1), as it enables measuring the *inertial* state of the spacecraft, since the reference images are very accurately referenced to the body's surface. Therefore, TAN improves the *absolute* accuracy of a landing. But the localisation accuracy is limited by the resolution of the *a-priori* images. Consequently, TAN can only be used at higher altitudes during the earlier phases of a descent. Often, TAN is done by extracting craters from the navigation images and comparing these to a crater catalogue generated pre-mission. TAN is not further treated in this work, as the final descent phase is investigated, during which TAN is not applicable.

At lower altitudes, as considered in this work, consecutive images are compared to each other, from which information concerning the spacecraft movement can be drawn.

[1] The reference system used here has the x- and y-axis in the horizontal plane and the z-axis in the vertical direction.

[2] Note that not all this information is always retrieved from the images, but in theory, these four elements can be computed from image or surface-map measurements.

This will result in a positioning *relative* to the surface and is therefore called terrain relative navigation. More precisely, since the navigation task performed by the filter is to minimise the error accumulated relative to the hazards/landing site identified in the hazard map, the algorithm will be referred to as hazard-relative navigation (HRN) from now on.

The basic principle of all three methods, TAN, TRN and HRN is similar. A measurement of the surface is recorded during the descent, which can either be a 3-D map or a 2-D image, with the latter being more common. This image or map is then compared to an earlier image or map of the surface. This can either be taken on-board a couple of time steps back (HRN/TRN), during a previous mission, or much earlier in the same mission and potentially processed into some other format like a crater catalogue (TAN).

TRN and HRN need the ability to match the current image to a previous image and thus a way to tell which part of the current image is also present in the previous image. Based on this matching, it is then possible to derive where the vehicle is with respect to the earlier image or how much it moved in between. From this it is possible to derive the spacecraft's position, velocity, orientation or rotation. Which of these four values can be computed depends on the measurements and the techniques used.

Note that TRN/HRN techniques cannot link its measurements to absolute locations in a planet-fixed reference system, while TAN can. Therefore, only TAN methods can remove the initial x- and y-errors whereas TRN/HRN can only limit the growth of these by not accumulating more error relative to the previous image (or hazard map).

3.3. SIMULTANEOUS LOCALISATION AND MAPPING

The filter will be updated using features extracted from the images, without linking them to an *a-priori* feature map, but including them in the state. In this section, a brief introduction to SLAM will be given.

If one would like to construct a map of a room, this could be done by placing a sensor (for example, a range finder) in a known position and then measuring the distance to all walls from that position. Based on these measurements, it would be possible to construct a map of the room, the only error in this would be the errors linked to the sensor used.

On the other hand, if the map of the room is perfectly known and a vehicle is placed in the room, it is possible to determine the vehicle's location within the room by taking measurements and using the maps to derive the vehicle's position from these measurements (for example, by multiple range measurements). Again, the only inaccuracies in the acquired pose would be caused by errors in the sensor(s).

As described by the name, in SLAM, localisation of a vehicle is done at the same time when a map of the surrounding environment is build, and the localisation is done within this map. However, at the beginning neither the map nor the vehicle state is known. It is therefore a chicken-and-egg problem: the vehicle state is unknown, as well as the surrounding. Thus the measurements of the surroundings are relative to this erroneous vehicle state and the vehicle error will thus influence the map constructed from the measurements. Of course, also errors in the measurements will influence the map. In Fig. 3.1, the SLAM principle used is presented for a landing sequence as analysed in this work. In the figure, the lander is depicted at three different instances in time, namely t_1,

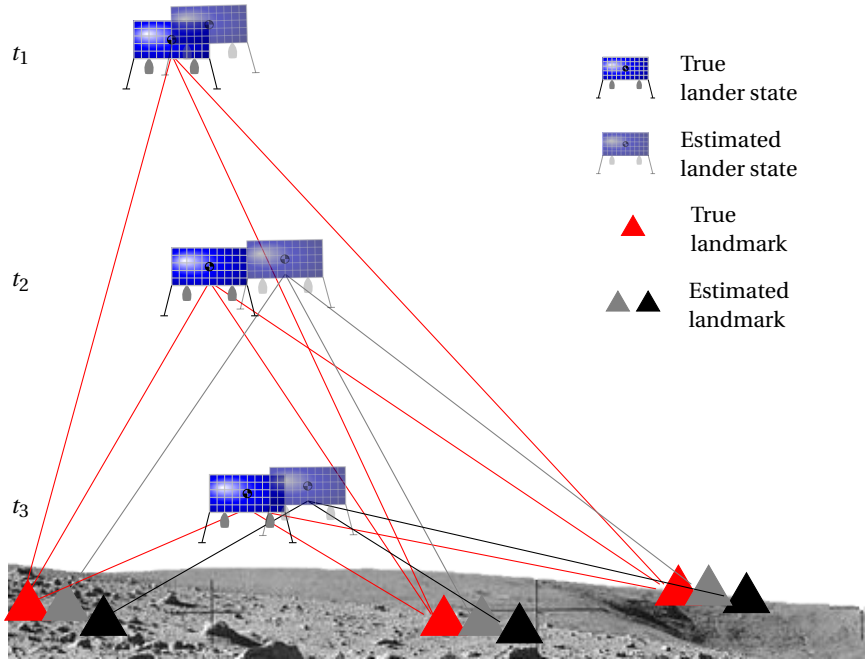


Figure 3.1: SLAM principle

t_2, and t_3. Moreover, the actual (true) position of the lander is shown, which is not the same as the position the system it believes it is in (the estimated lander state). While the landmarks are measured from the true position, the system will map the landmarks to slightly different positions on the surface, since the system references the measurements to the estimated lander state. In the next step, the system measures the distance to the (same) landmarks again. Based on the previous measurement, the system will already have a certain estimate of this measurement, and the closer the estimated lander state will approach the true state, the smaller the difference between estimated measurement and the actual measurement will get, and therefore the landmarks will be mapped closer and closer to their real positions, while also improving the state estimate.

The back-bone of SLAM is always some kind of filter that predicts both the vehicle state and the map/features, and compares the predicted features to the actual measurements. Here, Kalman filters are commonly used, however also more “advanced” methods such as particle filters and bundle adjustment are employed. In this work a Kalman filter approach was chosen, since the particle filter or bundle adjustment will cause a higher computational load and computational efficiency is very important for space applications. Moreover, Kalman filtering is widely used for space applications. Clearly, computational efficiency is the main driver in designing algorithms for space applications, however also the “tried and true” principle of using a method already accepted by the community should not be underestimated.

Since not only the vehicle state, but also the map is to be estimated, the map has to be

included in the state. This means that the map elements will have their own covariances, but also that the link between vehicle state and map elements can be exploited in the filter.

Various versions of maps exist in SLAM. Common are topographical maps, mostly in 2D. In this work the sensor measures full 3D maps of the surface. However, the SLAM algorithm will not be able to include these full maps into the state, as will be discussed in more detail later. In the applied filter, the map is represented by a list of features, and does thus not contain a full/dense map of the surface mapped by the sensor.

In robotic SLAM, an event called loop-closure might occur. Loop-closure describes the situation, where the robot revisits a location it has previously visited and recognises this. It is then possible to update and improve the previously built map based on the more accurate localisation in the event of loop-closure. SLAM can exist without loop-closure. In the case of a planetary descent no loop-closures will and can occur, since the lander will never "revisit" a prior location during a descent (as it is basically only a "forward" movement). In terms of Fig. 3.1, a loop-closure would occur if the lander would return to the location of t_1 after t_3.

3.4. DESIGN OF THE HAZARD-RELATIVE NAVIGATION FILTER

Next, the filter design will be discussed. In the previous sections the basic principles of navigation in general, relative navigation, simultaneous localisation and mapping, and Kalman filtering have already been addressed. These concepts form the basis of the methods applied in the following sections. This section will present the choices made in the design of the HRN filter as well as the filter itself.

3.4.1. LAYOUT

In this work, the proof of concept of a hazard-relative navigation method is presented. The hazard-detection system makes use of stereo-vision based 3-dimensional surface reconstruction. Therefore, both 3-D surface maps and stereo images are available as a measurement for the HRN filter. Moreover, an IMU is available as is true for all conventional landing systems.

Since maps play an important roll in hazard detection, the technique of SLAM, frequently used in robotics for on-board map building and state estimation, served as a reference for this method. No full-scale SLAM can be employed due to the great complexity of this system. Still, a SLAM-like approach, including selected map elements (surface features) to the vehicle state is followed. This should serve as a first step towards more complete SLAM during planetary descent, but also simply improve the state estimation during the descent. Next to that, features included in the state will be known very accurately after touch down, and can therefore be used to assemble maps for rover driving from the hazard-detection DEMs, as the accurate knowledge of feature locations may aid linking individual DEMs.

Since the aim of this work is to demonstrate that HRN is indeed possible using the stereo maps, and not to achieve the best possible performance of this method, it was decided to select a very conventional, robust, and proven method as a navigation filter. Therefore an error-state Kalman filter (ESKF), also called indirect Kalman filter, was cho-

sen. This filter estimates the error in the state rather than the actual state. This turns the non-linear problem of a common Extended Kalman Filter (EKF) into a linear problem, provided that the error remains small. Since the error is reset to zero after every update this assumption is valid. Therefore, no linearisation of the system matrix is necessary in the ESKF opposed to an EKF, recall from the brief introduction of the KF that for linear filters both convergence and optimality of the filter are guaranteed. Moreover, the ESKF avoids the problem of the over-parametrisation of a quaternion for attitude representation as well, as the attitude is represented by an error descriptor, which is a minimal representation. Moreover, in the error formulation the error state values will always stay close to the zero. This ensures that the attitude values will stay far away from the singularities present.

Madyastha et al. (2011) compared the EKF to an ESKF for attitude estimation and give a discussion of both systems. They showed that the ESKF consistently performed better than the EKF, was more robust in its tuning, and more robust to filter failures. Also, both Mourikis et al. (2009) and Delaune et al. (2011) make use of a similar approach for state estimation in their work on TRN. The ESKF is thus a very simple to use method with good robustness. The ESKF is similar to the multiplicative extended Kalman filter (MEKF) used for attitude estimation. The MEKF is also a variant of an indirect Kalman filter, performing state updates based on the error state rather than the full state. The term *multiplicative* is used since the quaternion update is not added to the state, but injected using a quaternion multiplication. This is also true for the ESKF, see Eq. (3.13)[3].

If this research would be continued and more focus on the real-time applicability would be put, other filters might be considered. In robotic SLAM, particle filters are often used instead of EKFs. This is because particle filters are able to treat non-linear process models (and without linearisation around the current state as in EKFs), as well as non-Gaussian systems. Still, the computational load is considerably higher than for an EKF, as the Kalman filter works with a Gaussian distribution (thus with a mean and a variance), while particle filters propagate a large set of individual particles. Moreover, in space applications particle filters are not (yet) frequently employed. Therefore, the choice for an ESKF is also in line with the usual tendency in space applications to stick with heritage and robust methods.

Figure 3.2 shows the set-up of the full HRN system. The vision system and the navigation filter are indicated as distinct modules. The former uses two stereo images, as well as one image taken at the previous measurement step as input. The stereo pair is used for depth estimation (Block **Stereo**, based on the stereo DEM construction as presented in Chapter 2), while the image from the previous measurement step and left image of the current stereo pair are used for feature matching and tracking (Blocks **Matching** and **Features**). In the navigation loop, an IMU is used for state propagation (Block **Propagation**).

The IMU is modelled using a bias and additive white noise for both the accelerometers and the gyroscopes. Misalignment, scale and other error sources are not considered yet, but could be added to the system. The measured acceleration, $\boldsymbol{a}_m$, and angular rate,

[3] In the ESKF formulation, not all elements of the state are updated using a multiplication. Therefore, the term MEKF will not be used further in the work. Still, it is important to note that the attitude part of the filter is similar to a MEKF.

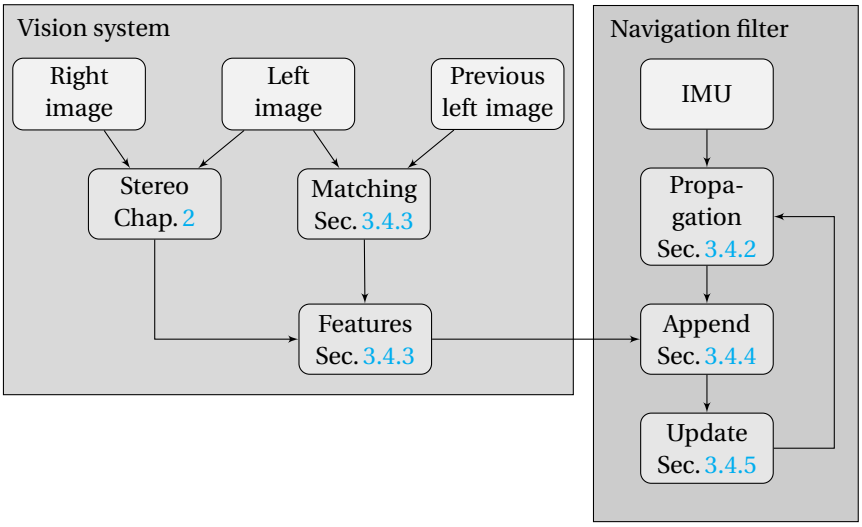


Figure 3.2: System set-up

$\boldsymbol{\omega}_m$, are thus described by

$$\boldsymbol{a}(t)_m = \boldsymbol{a}(t) + \boldsymbol{b}_a(t) + \boldsymbol{n}_a(t) \tag{3.3}$$

$$\boldsymbol{\omega}(t)_m = \boldsymbol{\omega}(t) + \boldsymbol{b}_g(t) + \boldsymbol{n}_g(t) \tag{3.4}$$

where $\boldsymbol{a}$ is the true acceleration, $\boldsymbol{\omega}$ the true angular rate, $\boldsymbol{b}$ are the biases, $\boldsymbol{n}$ are the white noises and the subscripts a and g denote the accelerometer and the gyroscopes, respectively. The biases are described by a random walk process:

$$\dot{\boldsymbol{b}}_a(t) = \boldsymbol{n}_{ba} \tag{3.5}$$

$$\dot{\boldsymbol{b}}_g(t) = \boldsymbol{n}_{bg} \tag{3.6}$$

The biases in discrete time can be found by integrating these equations. Note that an IMU only measures an acceleration in the presence of a thrust force, thus during powered descent, and not during free fall. Thus it is assumed that a powered descent is flown during the last part of the descent trajectory. Nevertheless, the image measurements employed do not have this constraint and could thus also be used in the case of a free fall trajectory (if the descent is sufficiently slow to make use of these measurements, for example a Venus or Titan landing)

Once a measurement is recorded new features are added to the state in a SLAM-like manner (Block **Append**). Re-observed features are used in the update step to update the state (Block **Update**). The navigation loop is executed until touchdown is reached or the filter is stopped. In this chapter, each of the blocks, apart from the already covered **Stereo** block, will be discussed in more detail.

3.4.2. PROPAGATION

In Kalman filtering, propagation, or state prediction, is the process of calculating the spacecraft state at the current time step based on the state information from the previous time step based on mathematical models describing the state evolution. Next to that, also some form of inertial measurement may be used during propagation. This so-called state holds the information representing, for example, the spacecraft's position and orientation, but might also hold other time-varying spacecraft properties, which are estimated by the filter (*e.g.*, instrument biases in some systems, but also mass). Theoretically speaking, the augmented state may contain an infinite number of elements, however, the larger the state the slower the filter will get. This is largely linked to the computation of matrix inverses in the process of state propagation and update.

As stated before, the ESKF is used in this work. This type of Kalman filter tries to estimate the error accumulated in the system in between two consecutive update steps. After this error is estimated, it is injected (*i.e.*, removed) from the propagated full state. Therefore, it is necessary to also propagate the full state, while the filter estimates the error state. In theory also the error state is propagated. However, the mean of this error should always be zero. Thus, the propagation of the error should always return zero and can be omitted. The covariance prediction for the error state is non-zero and cannot be skipped.

The propagation equations presented in the following are commonly used in literature to represent landing systems. The propagation equations and state models presented in this section are similar to those used by Mourikis et al. (2009) and Delaune et al. (2011).

The true, full, state was chosen to contain 16 elements

$$\boldsymbol{x} = [\boldsymbol{r}^T \boldsymbol{v}^T \boldsymbol{q}^T \boldsymbol{b}_a^T \boldsymbol{b}_g^T] \tag{3.7}$$

with:

- $\boldsymbol{r}$ the position of the lander in the global reference frame, G, fixed to the planet's center. The dimension is $[3x1]$
- $\boldsymbol{v}$ the velocity of the lander in the global reference frame. The dimension is $[3x1]$
- $\boldsymbol{q}$ the quaternion representing the rotation from the global to the body reference frame, ($\boldsymbol{q}_G^B$). The dimension is $[4x1]$
- $\boldsymbol{b}_a$ the IMU accelerometer measurement bias. The dimension is $[3x1]$
- $\boldsymbol{b}_g$ the IMU gyroscope measurement bias, also called drift. The dimension is $[3x1]$

If for example IMU misalignment or scale errors were to be estimated as well, these would be added to the state.

The fundamental difference of an ESKF over a conventional EKF or KF is that the error state is used in the measurement and update step rather than the full state. This error state is defined as

$$\delta\boldsymbol{x} = [\delta\boldsymbol{r}^T \delta\boldsymbol{v}^T \delta\boldsymbol{\theta}^T \delta\boldsymbol{b}_a^T \delta\boldsymbol{b}_g^T] \tag{3.8}$$

To avoid the non-minimal representation of the full quaternion the ESKF uses the three-dimensional error $\delta\boldsymbol{\theta}$ in place of the error quaternion $\delta\boldsymbol{q}$. Other methods exist to avoid the over-parametrisation of the quaternion. It is, for example, possible to manually enforce the quaternion to be unit quaternion, while it is also possible to introduce a so-called pseudo-measurement, $\text{norm}(q) = 1$, on the quaternion in the filter. The detailed study of this would have been beyond the scope of this research. Markley (2003) show in their work that all three-element attitude representations studied, were similar in linear EKFs. Thus, it is assumed that the choice presented here should have no relevant negative effect on the filter performance. $\delta\boldsymbol{\theta}$ is related to the error quaternion as follows, assuming small errors. The error quaternion can be written as:

$$\delta\boldsymbol{q} = [\delta\boldsymbol{p}^T \delta q_4] \tag{3.9}$$

The error quaternion can be computed from an axis-angle representation (a notation where a rotation, θ, is described by one rotation angle around one unit vector, $\boldsymbol{k}$) by

$$\delta\boldsymbol{q} = [\boldsymbol{k}\sin\delta\theta/2; \cos\delta\theta/2] \tag{3.10}$$

By assumption the error $\delta\theta$ is small, thus small-angle approximations can be used and Eq. (3.10) can be simplified to:

$$\delta\boldsymbol{q} \simeq [\delta\boldsymbol{\theta}/2; 1] \tag{3.11}$$

where $\boldsymbol{\theta} = \boldsymbol{k}\theta$. The attitude is thus represented by 3 parameters in the error state. Therefore the error state has one element less than the true state and is a true minimal representation opposed to the over-parametrised full quaternion. It should be noted that other methods exist to avoid the over-parametrisation of the quaternion inside the filter. The chosen formulation is a approach commonly used in ESKFs.

To retrieve the full state from the error state, the errors are simply added to the state estimate, $\hat{\boldsymbol{x}}$ which is obtained from state propagation, which is explained in more detail later in this section.

$$\boldsymbol{x} = \hat{\boldsymbol{x}} + \delta\boldsymbol{x} \tag{3.12}$$

However, the true quaternion and its relation with the error quaternion and the estimate is not additive, but has to be computed by a quaternion multiplication:

$$\boldsymbol{q} = \delta\boldsymbol{q} \otimes \hat{\boldsymbol{q}} \tag{3.13}$$

The state estimate uses $\delta\boldsymbol{q}$, while the update (see Sec. 3.4.5) uses $\delta\boldsymbol{\theta}$. This has to be taken into account when merging the predicted state with the error retrieved from the update (see Sec. 3.4.5). Eq. (3.11) can be used to compute the error quaternion from $\delta\boldsymbol{\theta}$.

The following equations describe the evolving state in continuous time:

$$\dot{\boldsymbol{r}}(t) = \boldsymbol{v}(t) \tag{3.14}$$

$$\dot{\boldsymbol{v}}(t) = \boldsymbol{a}(t) \tag{3.15}$$

$$\dot{\boldsymbol{q}}(t) = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}(t))\boldsymbol{q}(t) \tag{3.16}$$

$$\dot{\boldsymbol{b}}_a(t) = \boldsymbol{n}_{ba}(t) \tag{3.17}$$

$$\dot{\boldsymbol{b}}_g(t) = \boldsymbol{n}_{bg}(t) \tag{3.18}$$

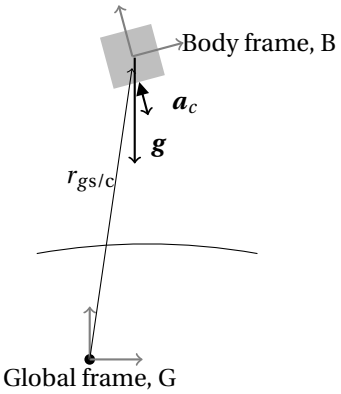


Figure 3.3: Frames and forces.

where $\boldsymbol{a}$ is the full acceleration including the gravity acting on the lander, $\boldsymbol{n}_{ba}$ and $\boldsymbol{n}_{bg}$ are the noises of the accelerometer and the gyroscope respectively, $\boldsymbol{\omega}$ is the angular rate relative to the rotating planet and $\boldsymbol{\Omega}$ is defined as:

$$\boldsymbol{\Omega\omega} = \begin{bmatrix} -[\boldsymbol{\omega}\times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix} \tag{3.19}$$

$$[\boldsymbol{\omega}\times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{3.20}$$

State propagation is done based on the measurements of an IMU, namely the gyroscope and the accelerometer. The gyroscope measurement is given by

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \boldsymbol{R}(\boldsymbol{q})\boldsymbol{\omega}_G + \boldsymbol{b}_g + \boldsymbol{n}_g \tag{3.21}$$

where $\boldsymbol{R}(\boldsymbol{q})$ is the rotation matrix equivalent to the rotation described by $\boldsymbol{q}$, where $\boldsymbol{q}$ described the orientation of the body frame in the global frame, $\boldsymbol{n}_g$ is a zero mean white Gaussian process noise and $\boldsymbol{\omega}_G$ is the planet's rotation. The involved frames are shown in Fig 3.3. The accelerometer measurement is given by

$$\boldsymbol{a}_m = \boldsymbol{R}(\boldsymbol{q})(\boldsymbol{a} - \boldsymbol{g} + 2[\boldsymbol{\omega}_G\times]\boldsymbol{v} + [\boldsymbol{\omega}_G\times]^2\boldsymbol{r}]) + \boldsymbol{b}_a + \boldsymbol{n}_a \tag{3.22}$$

where $\boldsymbol{g}$ is the gravitational acceleration in the local frame. Note that $\boldsymbol{a}$ is the full acceleration of the body relative to the rotating global planet frame and does therefore include $\boldsymbol{g}$. Thus, it is necessary to subtract $\boldsymbol{g}$.

The state estimate can be described using the equations below. Note, that these equations are in continuous time. However, for readability the argument (t) is omitted

in the following equations

$$\dot{\hat{\boldsymbol{r}}} = \hat{\boldsymbol{v}} \tag{3.23}$$

$$\dot{\hat{\boldsymbol{v}}} = \boldsymbol{R}(\hat{\boldsymbol{q}})(\boldsymbol{a_m} - \hat{\boldsymbol{b}}_a) - 2[\boldsymbol{\omega}_m\times]\hat{\boldsymbol{v}} - [\boldsymbol{\omega}_m\times]^2\hat{\boldsymbol{r}} + \boldsymbol{g} \tag{3.24}$$

$$\dot{\hat{\boldsymbol{q}}} = \frac{1}{2}\Omega(\boldsymbol{\omega_m} - \hat{\boldsymbol{b}}_g - \boldsymbol{R}(\boldsymbol{q})\boldsymbol{\omega}_G)\hat{\boldsymbol{q}} \tag{3.25}$$

$$\dot{\hat{\boldsymbol{b}}}_a = \boldsymbol{0} \tag{3.26}$$

$$\dot{\hat{\boldsymbol{b}}}_g = \boldsymbol{0} \tag{3.27}$$

The linear system model for the error state is given by

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{F}\delta\boldsymbol{x} + \boldsymbol{G}\boldsymbol{n}_{IMU} \tag{3.28}$$

where the system matrix $\boldsymbol{F}$ and the system noise matrix $\boldsymbol{G}$ are computed from Eqs. (3.23) to (3.27). $\boldsymbol{n}_{IMU}$ is a vector composed of all IMU noises present, $\boldsymbol{n}_a$, $\boldsymbol{n}_{ba}$, $\boldsymbol{n}_g$ and $\boldsymbol{n}_{bg}$. The Jacobian elements $\boldsymbol{F}$ and $\boldsymbol{G}$ have to be calculated with respect to the error state $\delta\boldsymbol{x}$ and not the full state.

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ -[\boldsymbol{\omega}_m\times]^2 & -2[\boldsymbol{\omega}_m\times] & -\boldsymbol{R}(\hat{\boldsymbol{q}})(\boldsymbol{a}_m - \hat{\boldsymbol{b}}_a) & -\boldsymbol{R}(\hat{\boldsymbol{q}}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -[\boldsymbol{\omega}_G\times] & \boldsymbol{0}_{3\times3} & -\boldsymbol{I}_3 \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix} \tag{3.29}$$

The $\boldsymbol{G}$ matrix is

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{R}(\hat{\boldsymbol{q}}) \\ -\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix} \tag{3.30}$$

Until now, all equations are given in continuous time. However, the IMU measurements are only available at discrete time steps δt. Thus, the state propagation in discrete time (of the full state, the error state is not propagated, as discussed before) is done by numerical integration of Eqs. (3.23) to (3.27). In this work we use an Euler integrator for this task. Also a Runge-Kutta integrator was tested, however this did not lead to any performance improvement, thus the simpler and faster Euler method is used. No further trade-offs with more advanced methods were done, as the Euler integrator performed sufficiently for the goal of demonstrating the feasibility of hazard-relative navigation. For further studies, more advanced integrators might be investigated to see if they can potentially improve the performance even more.

Next to the state and the error state, the Kalman filter makes use of an important matrix: the covariance, $\boldsymbol{P}$. The covariance describes the uncertainties of the state predictions computed. In the absence of a new measurement, the covariances will grow over time, while after a measurement, the covariances will be updated and should decrease. Thus, the error covariance matrix has to be propagated during the propagation

step. Note, that the error covariance matrix is linked to the error state and not the full state. The propagation of the discrete time $\boldsymbol{P}_{k|k-1}$ to $\boldsymbol{P}_k$ is given by

$$\boldsymbol{P}_k = \boldsymbol{F}\boldsymbol{P}_{k|k-1}\boldsymbol{F}^T + \boldsymbol{G}\boldsymbol{Q}_{k|k-1}\boldsymbol{G}^T \tag{3.31}$$

where the matrix $\boldsymbol{Q}$ represents the covariance matrix of the IMU measurement noise. This parameter has to be tuned for best filter performance, which will be addressed later.

The state is propagated with these equations until a image measurement is recorded.

3.4.3. MEASUREMENT

To update the state propagated based on the IMU, to estimate the biases and to get a more precise estimate of the full state, a measurement has to be recorded. Before discussing the update equations in Sec. 3.4.5, first the measurement itself will be presented in this section.

The measurement is performed by combining the stereo maps as presented in Chapter 2 with a feature-matching and tracking algorithm. These features are tracked over time during the descent and are included in the state in a SLAM-like manner, as will be explained in Sec. 3.4.4. From the stereo DEMs, measurements from the camera to the selected features are available. The stereo measurements can be translated to the 3-D location of the features relative to the cameras reference frame, and thus the spacecraft. The z-components of these measurements, however, represent the absolute distance to the surface. This means that the algorithm can only provide a relative localisation in x- and y-direction, while it can provide absolute knowledge in z-direction.

The DEMs created during hazard mapping have a size of 512×512 pixels. Even though it would be potentially very interesting to add these full maps to the state, this would mean adding 786,432 elements to the state (262,144 pixels and 3 coordinates per pixel). Since the resolution constantly improves during the descent, this state would even continue to grow. Even though it sounds very interesting to also predict and improve the full DEMs in a SLAM-like approach during the descent, it is simply not possible to include the full maps into the state. Therefore, only a few distinct features are added. If a limited number of features is selected as a first set, and the lander descends towards the surface, it is to be expected that certain features will fall outside the field of view, but also that certain features will not be re-observed, for example, due to changing light conditions or noise in the images.

To limit the computational burden of the algorithm, it was chosen to not constantly add new features to the state, but to add one large set of features to the state and after that eliminating "lost" features, while not adding new ones. Note, that features are only useful when they are re-observed, thus when they are identified in two consecutive images. Moreover, the higher the numbers of re-observations per feature are, the more accurate the feature is known and the more it will improve the overall state estimation. If one would constantly add features and only keep the "strongest", it is likely that features are only re-observed once or very few times. By initialising the filter with a larger set and only removing features while not adding new features, will ensure that features will be re-observed many times, and thus lead to better performance of the filter.

The algorithm has the ability to introduce a new set of features in case the total number of observed features falls below 10. Therefore, the algorithm should also be robust

against sudden changes in the trajectory and thus sudden larger changes in viewing directions, which would cause the lander to loose all the features from its field of view. It would be possible to run two or more filters stacked, to overcome that introducing new features will always lead to an initialisation phase during which no features can be tracked from the previous step. Since under the conditions tested, this initialisation phase seemed to cause no problems, this possibility was not studied further.

Matching and extracting the features over multiple images is done using the Speeded Up Robust Features (SURF) descriptor. A feature descriptor is a method to mathematically describe a pixel with a set of numbers and thus being able it to find identical pixel by comparing these "tags". It is very important to understand that a feature is not necessarily a physical item (such as a rock) but simply a pixel that is "special" enough to stand out from its surroundings. SURF is known to be very fast and is supposedly also very robust method to extract features (Bay et al., 2006). SURF was selected, since it is known to work well and due to its robustness to larger translations/rotations, which are likely to happen during a descent. At the start of this research, corners (an approach simply testing how different a pixel is from its surroundings) where investigated as a feature extractor. However, no usable results were achieved. If this research will be continued with the aim to build a time-optimised flight-version, it might be interesting to investigate other descriptors.

SURF extracts pixels that stand out from their surroundings and moreover a descriptor is computed, which serves as an identification tag for that specific feature. The general idea is that the same feature is extracted from two different images; they will have the same descriptor and can thus be linked as "the same". After this matching is performed an outlier removal is performed to eliminate faulty matches. This can be done as the shift of feature positions in the first relative to the second image should be caused by the same camera motion in between capturing the two images. Therefore, vectors connecting matched features should result in the same estimated motion. Matches that do not satisfy this assumption are rejected. This three steps, extraction of features, matching of features, and removal of outliers is shown in one explanatory data-set in Fig. 3.4.

Observed, matched and valid features are then added to the state in a SLAM fashion. Namely, they are added to the state and are also propagated in the propagation step. Therefore, positions of re-observed features can be compared to their previously position in the update step. It is expected that this will further improve state estimation. If features are not re-observed, they are discarded and removed from the state. It is possible to retain these features in an off-line list and match newly extracted features not only against the features that are currently included in the state, but also against this list. Since the filter was found to be robust enough even without this addition, this was not implemented, since it was considered to be too computationally expensive. Figure 3.5 outlines the steps taken in extracting, matching the features and maintaining a feature list.

The discrete time measurement model, as a function of the true state, is described by (Crassidis and Junkins, 2004)

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{3.32}$$

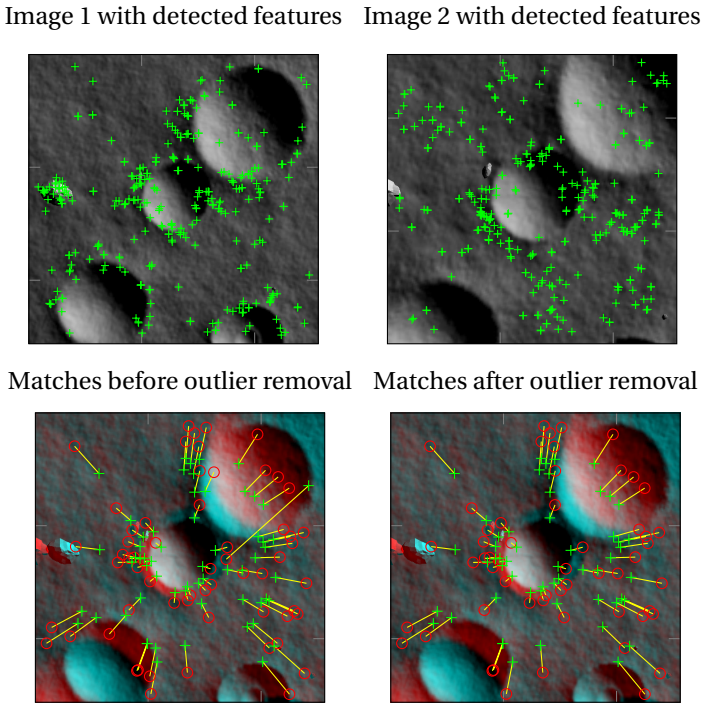


Figure 3.4: Feature extraction, matching and removal of outliers.

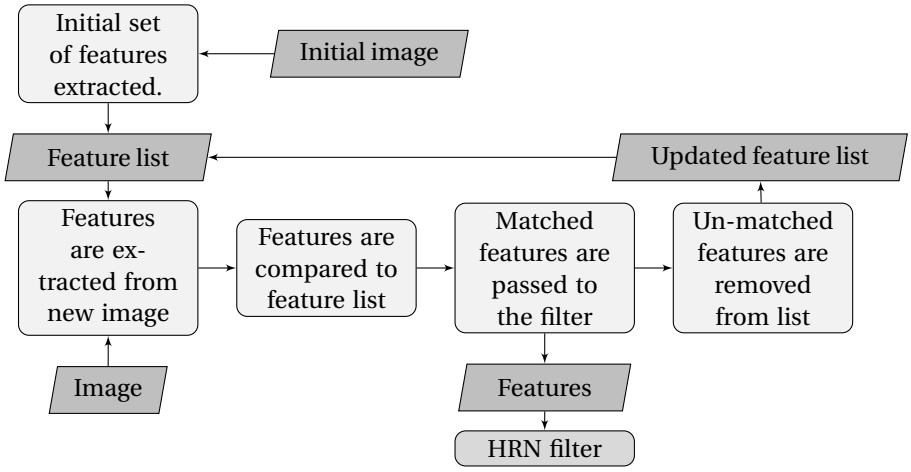


Figure 3.5: Conceptual repsentation of feature map and measurement set up.

The measurement is the feature position in the camera reference frame. Since the feature has already been observed during an earlier time step, it is possible to compute

the expected location of these features in the camera frame based on the location of the features in the global reference frame. Moreover it is necessary to transform the values from metre to pixels. Figure 3.7 shows the layout of the frames. Thus, the expected measurement can be described by the following equation.

$$\hat{\boldsymbol{z}}_k = \underbrace{\text{diag}(1/res, 1/res, 1)}_{\boldsymbol{T}_{r2m}} \boldsymbol{R}(\boldsymbol{q}_B^C)\boldsymbol{R}(\boldsymbol{q}_{G,k}^{\hat{B}})(\hat{\boldsymbol{r}}_{g,\text{Feature},k} - \hat{\boldsymbol{r}}_{g,\text{Body},k}) \tag{3.33}$$

where $\boldsymbol{R}(\boldsymbol{q}_B^C)$ and $\boldsymbol{R}(\boldsymbol{q}_{G,k}^B)$ describe the rotation from the body to the camera frame and from the global to the body frame, respectively. Combined, they represent the rotation from the global frame to the camera frame. The transformation matrix $\boldsymbol{T}_{r2m}$ maps a point from the 3D world, in metres, on the 2D image plane, in pixels.

The resolution, res, can be calculated using simple trigonometry as shown in Fig. 3.6.

$$res = \frac{\tan\frac{FOV}{2}(x_{3,k} - r)}{2s} \tag{3.34}$$

where r is the radius of the planet and s is the image size.

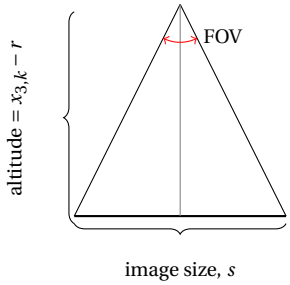


Figure 3.6: Resolution of an image.

The resolution is thus linked to the camera's position above the ground, which can be derived from the third element of the state, the vehicle's z-position. This means that bad estimates of z will in turn lead to bad estimates of x and y. Unfortunately, these values cannot be decoupled. Figure 3.7 depicts the measurements and the different frames involved. It can be seen that the measurement is recorded in the camera frame, while the camera (and thus the camera frame) is fixed with respect to the body frame of the spacecraft. The location of the spacecraft (and thus the body frame) is expressed with respect to the global frame.

In this research, it is further assumed that the surface of the planet is flat and that its rotation can be neglected. Since only the very last part of a descent is simulated, which only lasts a few seconds, this assumption is valid - if the rotational rate of the body is sufficiently small. Should this not be the case, for example, when landing on a fast rotating minor body, it is possible to adapt the filter quite easily for this need. The only necessary adaptation would be to move the features with the surface rotation in the propagation step. If the rotation is only inaccurately known, it might be valuable to estimate it along with the remaining states.

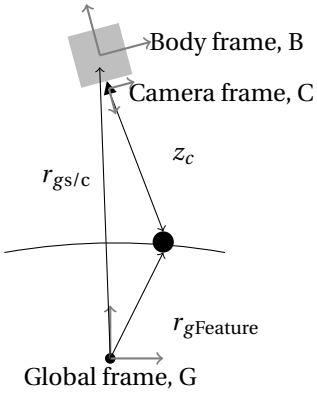


Figure 3.7: Frame definitions

Since the update, as discussed in Sec. 3.4.5, is done using the error state, also the measurement matrix $\boldsymbol{H}$ needs to be computed based on the error state, rather than the nominal state. This means that also the Jacobian, of the measurement function $\boldsymbol{h}$, has to be computed with respect to error state:

$$\boldsymbol{H} \triangleq \frac{\partial \boldsymbol{h}}{\partial \delta \boldsymbol{x}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \delta \boldsymbol{x}} = \boldsymbol{H}_x \boldsymbol{X}_{\delta \boldsymbol{x}} \tag{3.35}$$

Here, the first part $\boldsymbol{H}_x$ is the "normal" Jacobian with respect to the full state, as also used in a standard EKF formulation. $\boldsymbol{X}_{\delta \boldsymbol{x}}$ is the Jacobian of the state with respect to the error state. As discussed previously, all elements of the corrected state, except for the term for the angles, are defined as summations of the error state and the propagated full state. For example, the position (the first three elements of the state) is defined as $\boldsymbol{r}+\delta \boldsymbol{r}$. Therefore, the Jacobian element of the position with respect to the error in position becomes

$$\boldsymbol{X}_{\delta \boldsymbol{x}, \delta r} = \frac{\partial (\boldsymbol{r} + \delta \boldsymbol{r})}{\partial \delta \boldsymbol{r}} = \boldsymbol{1} \tag{3.36}$$

The same holds true for all these derivatives, but $\boldsymbol{X}_{\delta \boldsymbol{x}, \delta \theta}$, since the error link to the attitude is not described by a sum, but a quaternion multiplication.

The Jacobian $\boldsymbol{X}_{\delta \boldsymbol{x}}$ is

$$\boldsymbol{X}_{\delta \boldsymbol{x}} = \begin{bmatrix} \boldsymbol{I}_{6\times 6} & \boldsymbol{0}_{6\times 4} & \boldsymbol{0}_{6\times (6+3n)} \\ \boldsymbol{0}_{4\times 6} & \boldsymbol{X}_{\delta \boldsymbol{x}, \delta \theta} & \boldsymbol{0}_{4\times (6+3n)} \\ \boldsymbol{0}_{(6+3n)\times 6} & \boldsymbol{0}_{(6+3n)\times 4} & \boldsymbol{I}_{(6+3n)\times (6+3n)} \end{bmatrix} \tag{3.37}$$

where n is the number of features in the state and $\boldsymbol{X}_{\delta\boldsymbol{x},\delta\theta}$ is given by:

$$\boldsymbol{X}_{\delta\boldsymbol{x},\delta\theta} = \frac{\delta(\boldsymbol{q}\otimes\delta\boldsymbol{q})}{\delta\delta\boldsymbol{\theta}} \tag{3.38}$$

$$= \frac{\delta(\boldsymbol{q}\otimes\delta\boldsymbol{q})}{\delta\delta\boldsymbol{q}}\frac{\delta\delta\boldsymbol{q}}{\delta\delta\boldsymbol{\theta}} \tag{3.39}$$

$$= \frac{(\boldsymbol{q})_L\delta\boldsymbol{q}}{\delta\delta\boldsymbol{q}}\frac{\delta\begin{bmatrix}\frac{1}{2}\delta\theta\\1\end{bmatrix}}{\delta\delta\boldsymbol{\theta}} \tag{3.40}$$

$$= (\boldsymbol{q})_L\frac{1}{2}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\0&0&0\end{bmatrix} \tag{3.41}$$

$$= \begin{bmatrix}q_4&-q_3&q_2&q_1\\q_3&q_4&-q_1&q_2\\-q_2&q_1&q_4&q_3\\-q_1&-q_2&-q_3&q_4\end{bmatrix}\frac{1}{2}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\0&0&0\end{bmatrix} \tag{3.42}$$

$$\boldsymbol{X}_{\delta\boldsymbol{x},\delta\theta} = \frac{1}{2}\begin{bmatrix}q_4&-q_3&q_2\\q_3&q_4&-q_1\\-q_2&q_1&q_4\\-q_1&-q_2&-q_3\end{bmatrix} \tag{3.43}$$

The next step is to compute the Jacobian element $\boldsymbol{H}_x$. For Eq. (3.33) it is known that the measurement is only a function of the spacecraft position, $\boldsymbol{r}$, the feature position $\boldsymbol{r}_{\text{Feature}}$, the resolution, and $\boldsymbol{q}$. Therefore, all elements, but for the Jacobian elements with respect to these state variables, $\boldsymbol{H}_r$, $\boldsymbol{H}_{r,\text{Feature}}$ and $\boldsymbol{H}_q$, are zero. The Jacobian of the measurement function with respect to the spacecraft position is:

$$\boldsymbol{H}_r = \boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})\boldsymbol{R}(\hat{\boldsymbol{q}}_{\boldsymbol{G}}^{\boldsymbol{B}})\begin{bmatrix}-1/res&0&\boldsymbol{r}_1u\\0&-1/res&\boldsymbol{r}_2u\\0&0&-1\end{bmatrix} \tag{3.44}$$

$$u = \delta\frac{1}{res} = \frac{512\tan(\text{FOV}/2)}{(\tan(\text{FOV}/2)\hat{x}_3 - \tan(\text{FOV}/2)r)^2} \tag{3.45}$$

Next, the Jacobian with respect to the current feature's position is

$$\boldsymbol{H}_{r,F} = \boldsymbol{T}_{r2m}\cdot\boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})\boldsymbol{R}(\hat{\boldsymbol{q}}_{\boldsymbol{G}}^{\boldsymbol{B}}) \tag{3.46}$$

where the transformation Real2Meas, which projects the coordinates of a feature point from the 3D camera reference frame on the 2D image plane, is given by

$$\boldsymbol{T}_{r2m} = \begin{bmatrix}1/\text{res}&0&0\\0&1/\text{res}&0\\0&0&1\end{bmatrix} \tag{3.47}$$

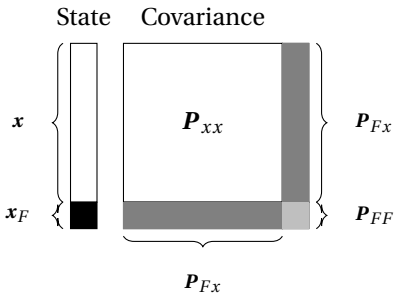


Figure 3.8: State and Covariance augmentation

And lastly the Jacobian with respect to the quaternion is:

$$\boldsymbol{H}_q = 2 \cdot \boldsymbol{T}_{r2m} \cdot \boldsymbol{R}(\hat{\boldsymbol{q}}_B^C)\{\hat{\boldsymbol{p}}(\bar{d}^T)\boldsymbol{I}_3 + \hat{\boldsymbol{p}}(\bar{\boldsymbol{d}})^T - \bar{\boldsymbol{d}}\hat{\boldsymbol{p}}^T + 2\hat{q}_4[(\bar{\boldsymbol{d}})\times]\hat{q}_4(\bar{\boldsymbol{d}}) + [\bar{\boldsymbol{d}}\times]\hat{\boldsymbol{p}}\} \tag{3.48}$$

$$\bar{\boldsymbol{d}} = \hat{\boldsymbol{r}}_F - \hat{\boldsymbol{r}}_B \tag{3.49}$$

where $\boldsymbol{H}_q$ was derived from the attitude matrix $\boldsymbol{R}(\boldsymbol{q}) = (q_4^2 - \boldsymbol{p}^T\boldsymbol{p})\boldsymbol{I}_3 + 2\boldsymbol{p}\boldsymbol{p}^T - 2q_4[\boldsymbol{p}\times]$, and $[\times]$ denotes the matrix form of a cross product in terms of a skew-symmetric matrix, see Eq. (3.20).

3.4.4. COVARIANCE AND STATE AUGMENTATION

In SLAM a key element is that observed map points (features) are added to the state. Therefore, after every measurement the state may have to be augmented: if a point in space is observed, it is added to the state or removed from the state if not re-observed, *i.e.*, the state vector is *augmented*. This means that the features have to be propagated. However, in this work features are not dynamic and therefore their locations do not change in the propagation step, and only change when updating. Here, it should be noted that this is only the case, because the final phase is so short that it is assumed that the rotation of the planet is negligible. If a fast-rotating body would be the target, or the algorithm would be applied for a longer period, the rotation might have to be accounted for, and thus the features would need to be propagated as well.

When the state is augmented, the error covariance matrix, $\boldsymbol{P}$, has to be augmented as well. This is depicted in Fig. 3.8. Here, it is important to note that the covariances of the added features are not independent. This is because feature locations are computed from the measurement, using information stored in the state.

For each measured feature, the feature's position in the global reference frame is appended to the state using the inverse measurement matrix, which is the inverse of the measurement function given by Eq. (3.33):

$$\boldsymbol{r}_{Feature} = \hat{\boldsymbol{r}}_{Body} + [\boldsymbol{T}_{r2m} \cdot \boldsymbol{R}(\boldsymbol{q}_B^C)\boldsymbol{R}(\hat{\boldsymbol{q}_G^B})]^{-1}\boldsymbol{z} \tag{3.50}$$

From this equation, the Jacobian elements with respect to the spacecraft state and the measurement z are computed, $\boldsymbol{J}_z$, since the inverse measurement function is a function

of the state and the measurement. Apart from $\boldsymbol{z}$, the inverse measurement function is only a function of the position and the quaternion, the Jacobian with respect to all other elements are zero. Therefore, only $\boldsymbol{J}_z$, $\boldsymbol{J}_{x,P_{\text{body}}}$ and $\boldsymbol{J}_{x,\theta}$ need to be computed.

$$\boldsymbol{J}_z = (\boldsymbol{T}_{r2m}\boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})\boldsymbol{R}(\hat{\boldsymbol{q}}_{\boldsymbol{G}}^{\boldsymbol{B}}))^{-1} \tag{3.51}$$

$$\boldsymbol{J}_{x,P_{\text{body}}} = \boldsymbol{I}_3 \tag{3.52}$$

The Jacobian $\boldsymbol{J}_{x,\theta}$ is more difficult to compute, as the equation is a function of $\boldsymbol{q}$ while the state contains $\delta\boldsymbol{\theta}$. In principle, the same approach as for the Jacobian $\boldsymbol{H}_{\delta\boldsymbol{x}}$ is followed, see Eq. (3.35). Therefore,

$$\boldsymbol{J}_{x,\theta} = \boldsymbol{H}_q\boldsymbol{X}_{\delta\theta} \tag{3.53}$$

$$\boldsymbol{H}_q = 2\boldsymbol{R}(\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{C}})^{-1}\left[\hat{\boldsymbol{p}}^T\boldsymbol{z}\boldsymbol{I}_3 + \hat{\boldsymbol{p}}\boldsymbol{z}^T - \boldsymbol{z}\hat{\boldsymbol{p}}^T + 2\hat{q}_4[\boldsymbol{z}\times] \quad \hat{q}_4\boldsymbol{z} + [\boldsymbol{z}\times]\hat{\boldsymbol{p}}\right] \tag{3.54}$$

where $[\boldsymbol{p}^T q_4]$ are the elements of the inverse of $\boldsymbol{q}_{\boldsymbol{G}}^{\boldsymbol{B}}$.

The new augmented covariance can be computed from (Bailey and Durrant-Whyte, 2006):

$$\boldsymbol{P}_{xx_{\text{aug}}} = \begin{bmatrix} \boldsymbol{P}_{xx} & \boldsymbol{P}_{Fx}^T \\ \boldsymbol{P}_{Fx} & \boldsymbol{P}_{FF} \end{bmatrix} \tag{3.55}$$

with

$$\boldsymbol{P}_{FF} = \boldsymbol{J}_x\boldsymbol{P}_{xx}J_x^T + \boldsymbol{J}_z\boldsymbol{R}\boldsymbol{J}_z^T \tag{3.56}$$

$$\boldsymbol{P}_{Fx} = \boldsymbol{J}_x\boldsymbol{P}_{xx} \tag{3.57}$$

If a feature is not re-observed or falls outside of the area imaged, it will be removed from the state. This means that the related elements will also be deleted from $\boldsymbol{P}_{xx}$. As mentioned before, it might be beneficial to not entirely remove these features, but to store them and compare them with the next sets of extracted features. They can than be re-entered in the state. Note that in this case, also the associated covariances have to be stored, as otherwise it will not differ from entering an entirely new feature.

3.4.5. STATE UPDATE

After state augmentation the state update can then be performed using the following equations (e.g., (Crassidis and Junkins, 2004)), where k denotes the current time step:

$$\boldsymbol{S}_k = \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k \tag{3.58}$$

$\boldsymbol{P}_{k|k-1}$ is the error covariance matrix as computed during the propagation step and $\boldsymbol{R}_k$ is the covariance of the measurement, like the process covariance $\boldsymbol{Q}$, this is a value that needs to be tuned. The tuning is addressed later. From $\boldsymbol{S}_k$ the Kalman gain can be computed

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k\boldsymbol{S}_k^{-1} \tag{3.59}$$

Using the Kalman gain, the error state can be computed. Since the prediction of the error is always zero, the error state update does not involve a term for the prior of the error state (opposed to the EKF formulation where the state update is computed based on the Kalman gain, the measurement and its prediction and the prediction of the state).

$$\delta\hat{\boldsymbol{x}}_k = \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k)) \tag{3.60}$$

Knowing the estimate of the error state, the estimated full state can be updated.

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \delta\hat{\boldsymbol{x}}_k \tag{3.61}$$

The final step is to update the predicted covariance:

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k|k-1}(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + (\boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k) \tag{3.62}$$

After this, the filter continues propagation of the state estimate until a new measurement is recorded.

Here, one has to be careful when merging the attitude error with the quaternion in the nominal state. The value in the error state is the three-dimensional $\delta\boldsymbol{\theta}$, which has to be translated to the four-dimensional $\delta\boldsymbol{q}$ first. Measurements from other sources could also be included during the update step, for example, from a star tracker or an altimeter. However, this is not necessary for the goal of this work, which is to proof the concept of HRN that is sufficiently accurate for a precise landing and not to develop the most accurate filter. It might be interesting for future work to trade-off the added value of additional sensors versus the added cost.

To summarise, the navigation system follows the following steps

1. Every time step: propagation of the state using a new inertial measurement.
2. Every x (a pre-set number) time steps: image-based measurement, compute the stereo DEM, extract and match features from images, link features to 3D coordinates from DEM.
3. After every measurement: state and covariance update based on the measurements.
4. After every update: Augmentation of the state and covariance by adding new features and removing lost features.

The filter presented in the following runs at 200 Hz, thus the IMU is sampled at this interval and the state is propagated, while the image measurements are taken at 0.8 Hz.

3.5. TUNING OF THE FILTER

As introduced in Secs. 3.4.2 and 3.4.3, the filter makes use of the noise covariances of both the system and the measurement, $\boldsymbol{Q}$ and $\boldsymbol{R}$. Here, both matrices express the confidence in the system equations and the measurement, respectively. Very advanced tuning methods exists, nevertheless it is till common practice to tune by trial and error based on a first good guess. The same approach was followed in this work.

For $\boldsymbol{R}$ a good first guess is based on the calibration of the sensor, however in this case the measurement is not directly coming from the sensor but post processed data. It is thus not possible to compute the sensor error directly. All non-diagonal elements of $\boldsymbol{R}$ are kept zero. The matrix is a 3×3 matrix, since every individual measurement consists of 3 elements, the x, y, z coordinates of the landmark. In the filter used during SILT and HILT the following covariance is used.

$$\boldsymbol{R} = \text{diag}(\begin{bmatrix} 0.05 & 0.05 & 0.05 \end{bmatrix}) \tag{3.63}$$

During tuning the filter showed no sensitivity to small changes in these values, however, large changes eventually lead to instabilities in the filter.

The matrix $\boldsymbol{Q}$ represents the uncertainties in the state model. Since the filter only has only few image measurements to interoperate, the filter is tuned to trust the measurements quickly. Normally it is common to give the filter time to converge to eventually reach a stable steady state but in this case it is desired that the filter converges fast while the steady-state is of less importance, as the filter will only run for a short while. Since the basic error state, before addition of any feature to the state, has 15 elements, $\boldsymbol{Q}$ is a 15×15 matrix.

$$\begin{aligned} \boldsymbol{Q} = \text{diag}(\begin{bmatrix} 1\cdot10^{-10} & 1\cdot10^{-10} & 1\cdot10^{-10} & 1\cdot10^{-10} & 1\cdot10^{-10} & 1\cdot10^{-10} & \dots \\ 1\cdot10^{-4} & 1\cdot10^{-4} & 1\cdot10^{-4} & 1\cdot10^{-8} & 1\cdot10^{-8} & 1\cdot10^{-8} & \dots \\ 1\cdot10^{-8} & 1\cdot10^{-8} & 1\cdot10^{-8} \end{bmatrix}) \end{aligned} \tag{3.64}$$

$\boldsymbol{Q}$ is robust to small changes in the inputs. Large changes will keep the filter from converging.

Lastly, it is also necessary to tune the values of the initial covariance $\boldsymbol{P}_0$, which is later updated during the prediction and update step. $\boldsymbol{P}_0$ is a measure for the confidence in the initial estimates were values close to zero express high trust in the initial estimates while large values show little confidence in the initial choice of values.

$$\begin{aligned} \boldsymbol{P}_0 = \text{diag}(\begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.5 & 0.5 & 0.5 & \dots \\ 1\cdot10^{-10} & 1\cdot10^{-10} & 1\cdot10^{-10} & 1\cdot10^{-4} & 1\cdot10^{-4} & 1\cdot10^{-4} & \dots \\ 1\cdot10^{-4} & 1\cdot10^{-4} & 1\cdot10^{-4} \end{bmatrix}) \end{aligned} \tag{3.65}$$

$\boldsymbol{P}_0$ is sensitive in the elements related to the orientation, $\boldsymbol{P}_{0_{7,7}}$ to $\boldsymbol{P}_{0_{9,9}}$. Increasing these values will lead to oscillation of the filter. The remain elements are reasonably robust to changes.

It is often practised to show the covariances and the filter residuals in one plot to demonstrate convergence of the filter, as this demonstrate that the filter will indeed satisfy the theoretical bounds after a certain period of initialisation. However, the filter is run with extremely few external (image) measurements and thus update steps. Therefore, filter convergence cannot be proven since the filter would require more measurements to reach a steady state. During the SILT presented in the following, it is concluded that only very few outliers (thus filter executions that do not lead to results better than the benchmark) are detected. Even though this is no conventional proof of convergence

it nevertheless shows that the filter performs well and does converge to a good solution almost always.

The SILT and HILT results show that the filter does, nevertheless, enable precision landing and thus performs as desired. It might be a starting point for future work to investigate methods entirely without a (Kalman) filter for using the image measurements for state estimation.

3.6. REFERENCE SCENARIO

During SILT, one can freely design a scenario for the descent, while during HILT there are constraints from the hardware/laboratory used that require the chosen scenario to be adapted to fit into the performance envelope of the set-up. Nevertheless, the basic considerations are the same.

Opposed to terrain-absolute and terrain-relative navigation methods, which are usually performed at altitudes of 10 km to 5 km and 1 km to 0.5 km, respectively, hazard-relative navigation is performed at even lower altitudes, namely where hazards can be identified by the hazard-detection system. This altitude is not a fixed value, but can vary depending on the HDA set-up, most importantly based on the sensors chosen. The main factors influencing at which altitude hazard detection becomes feasible is the vertical resolution, as well as the depth resolution of the corresponding DEMs. If either of the two resolutions is too low to resolve rocks and boulders, which would cause a landing failure, a conclusive dense hazard survey is not yet possible. Nevertheless, it should be noted, that a first coarse survey, identifying larger features and hazardous large scale slopes, might already be feasible at higher altitudes. The applicability and necessity of such a coarse survey strongly depends on the mission scenario, *e.g.*, the extent of *a-priori* knowledge of the surface, the descent time and the resulting constraints on execution time, the agility of the system in responding to landing site changes, but also on the sensing systems and their limitations.

In this research a stereo set-up was chosen as the sensing instrumentation. In Chapter 2 it was demonstrated that this kind of set-up can provide sufficient depth resolution for hazard detection from altitudes of 200 m and below. Since this work focuses on the mission phase where navigation with respect to identified hazards is desirable, the reference scenario was chosen to focus on this dense hazard-detection stage. Thus, a descent from 200 m downwards will be investigated.

If a sensing set-up would be chosen that would allow for dense hazard detection at higher altitudes, *i.e.*, due to a higher (depth) resolution at higher altitudes, the algorithm could obviously be used starting at higher altitudes. However, the ability to map early mainly relaxes the performance requirements for the HRN systems, as more time would be available. But it might also reduce fuel cost since easier diverts may be flown. More measurements, while keeping the same sampling rate, may be collected or the same number of measurements may be taken at a lower sampling rate. Overall, it can be concluded that starting HRN at higher altitudes will stress the algorithm less. Therefore, demonstrating the algorithm's capabilities on this more complex scenario, is not only showing the performance on the most challenging scenario, but will also allow to conclude that the same method may be applied to more relaxed scenarios.

3.7. SOFTWARE IN THE LOOP TESTING

The first step towards concluding whether the developed HRN method is a feasible candidate for future missions is SILT. During this testing, it is desired to simulate all input in a realistic manner and therefore to generate outputs that are representative for those achieved during a real mission.

3.7.1. TEST SET-UP

The set-up of the software-in-the-loop test is described in the following. As mentioned in Sec. 3.6 a final descent scenario from 200 m towards the landing site is simulated. This descent is performed over a Lunar(-analogue) surface. To assess the performance of the HRN method and judge if it will actually be a feasible candidate, it is necessary to compare the output to a benchmark. Since it should be tested how this method compares to conventional methods, the benchmark should be representative for those. To date, this very final phase of the descent is basically flown on IMU-only propagation, thus dead-reckoning. As a benchmark a IMU-only propagation is used. The IMU propagation models used for this and the HRN filter are identical, and were already discussed in Sec. 3.4.2. The IMU propagation is therefore identical to running the HRN filter without any updates. It should be noted that this benchmark does not include a altimeter which can reduce the error in z-direction and is necessary to ensure the spacecraft is not crashing into the ground. The point of this work was not to compare the HRN performance to the performance of an altimeter and therefore no altimeter was included for the benchmark.

Moreover, it is necessary to model the "truth", thus the actual flight path and values of all state elements. This truth is modelled assuming ideal control, *i.e.*, the accelerations commanded by the guidance law are applied perfectly and with no delay by the thrusters. The gravity is modelled as a function of altitude only, no spherical harmonics or perturbations are included. The current mission assumes a landing on the Moon and only the last 200 m of this near-vertical descent are studied. For such a scenario, this assumption is valid. However, when landing on a small irregular body, *e.g.*, an asteroid, this may not be valid depending on rotational rates, size and shape of the body. The propagation of the true state is done using a 4th-order Runge-Kutta integrator, the step size is equal to the IMU sampling rate, 0.005 s. Smaller step sizes did not improve the performance.

The just mentioned accelerations are commanded by the guidance function. Since a HDA scenario is simulated, the guidance law should be able to lead the vehicle to a selected landing location. The most simple law to do so is the E-guidance law. Note that the guidance law employed should not make a difference for the algorithm, therefore the choice of guidance law is arbitrary. The important point was not to model a pure vertical descent, such that the filter's performance in the xy-plane can be tested. Even if a purely vertical trajectory would be commanded in a real scenario, small perturbations and control inaccuracies will lead to small movements parallel to the image plane, therefore these should be covered during testing. E-Guidance will be discussed in more detail in Sec. 3.7.3.

The lander is equipped with two stereo cameras, rigidly attached to the vehicle with a 2 m baseline, and an IMU. The details of these instruments are discussed in the next

Table 3.1: IMU (accelerometer and gyro) error model based on (Geller and Christensen, 2009)

Error	Value	Unit
Accelerometer bias	30	μg
Accelerometer velocity random walk	20	m/s/$\sqrt{\text{s}}$
Gyro bias/drift	0.02	deg/h
Gyro angle random walk	0.00005	deg/$\sqrt{\text{s}}$
Sampling rate	200	Hz

section.

Due to the very low altitude at the start of the simulation, and the very short descent time remaining, a flat non-rotating lunar surface is assumed. This assumption was already made and discussed during filter design (see Chapter 3), and led to the conclusion that features stay constant during the propagation step. The equatorial rotational speed of the Moon is 4 m/s while for the Earth this is 400 m/s. Even for terrestrial applications of feature tracking and matching methods, the Earth rotation is not included. Since the Moon's surface moves even slower, this is a valid assumption.

3.7.2. INSTRUMENTATION

The simulated vehicle is equipped with an inertial measurement unit, for state propagation, and a stereo camera pair for HDA and HRN measurements. In conventional landing set-ups, an altimeter is always included in the descent instrumentation to measure the distance to the surface and thus identify (almost)touchdown. However, as the HDA measurements also provide line-of-sight measurements, as an altimeter does, the addition of an altimeter to the sensor suite is not necessary, but could be considered for redundancy, when actually implementing this method for a mission.

The IMU used is based on the values given by Geller and Christensen (2009). These are based on the NASA ALHAT program data, which represents the performance of a low fidelity IMU. In the ALHAT project, the state is propagated based on a 200 Hz IMU sampling rate, which is a realistic sampling rate for descent scenarios and is therefore used in this work. Table 3.1 gives the IMU error model as used in the simulations.

Next to the IMU, the filter uses a stereo camera set-up. It is thus necessary to simulate the camera images at every measurement time step. These images are obtained using PANGU, an external software capable of creating both lunar-analogue 3D surface models and images of them. In Sec. 2.2, PANGU was already discussed in detail. Figure 3.9 shows the scene used during SILT viewed from 210 m.

As discussed in Chapter 2, a 2 m baseline is used for the stereo cameras. Perfect knowledge of this baseline is assumed. The image size is 512×512 pixels. Again, this size was selected as a compromise between computation time, image resolution and the size of the actual area images. The stereo-vision algorithm would not be able to convert larger images into DEMs fast enough.

The cameras are based on the pinhole model. No distortions or sensor misalignments are modelled, as it is assumed that good camera calibration will be able to remove

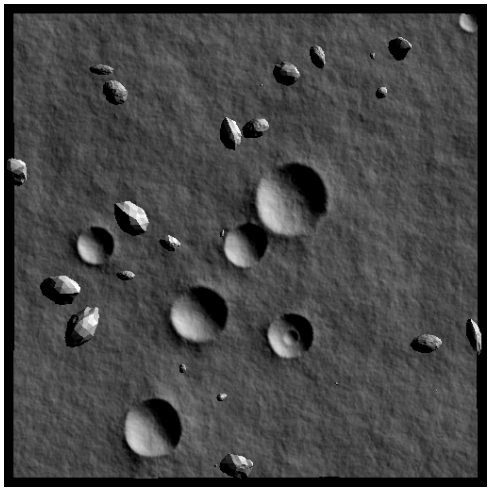


Figure 3.9: Image of the surface used during SILT captured at 210 m altitude with a nadir view.

these effects. If calibration would not be able to remove the alignment errors, this would have strong implications for the algorithm up to the point were it would not be able to compute any useful results. However, it is very difficult to simulate camera calibration and specifically the effects of erroneous calibration. Modelling this would be a study of its own. Therefore, it will be part of the HILT to investigate how camera errors will effect the performance. The camera field-of-view was chosen to be 30°, as this provides a good compromise for resolution per pixel and size of the imaged region in the selected operating range from 200 m downward. Figure 3.10 shows the resolution and image size as a function of altitude. Moreover, it shows the theoretical stereo depth-resolution as a function of altitude. It should be noted that the actual depth resolution is much better due to the quadratic fit involved (see Sec. 2.5). Note that based on the reported depth resolutions it is clear that HDA would not be possible at altitudes of 100 m and higher without this quadratic fit. From 200 m downwards, the vertical resolution is always below 20 cm, which is sufficient to detect all hazardous boulders/rocks.

Camera images are acquired at 0.8 Hz, which means that an image measurement is processed every 250 IMU sampling steps, while the IMU is sampled at a higher rate of 200 Hz. Camera images are taken down to an altitude of 40 m. It is assumed that below this altitude, acquiring images might be problematic due to dust from the thrusters. Moreover, this demonstrates the performance of the filter when stereo measurements will stop for a couple of measurement steps. The camera parameters are summarised in Table 3.2.

Only one scene is used during the SILT testing. In Chapter 2.5 it was demonstrated that the hazard-detection function performs equally well on different terrains. It is thus established that the HD function works independent of the terrain choice. Therefore, it is not necessary to change the terrain to prove that the hazard-relative algorithm will work on other terrains too, since the terrain related part of the system is the DEM generation. Moreover, a different terrain will be used during HILT.

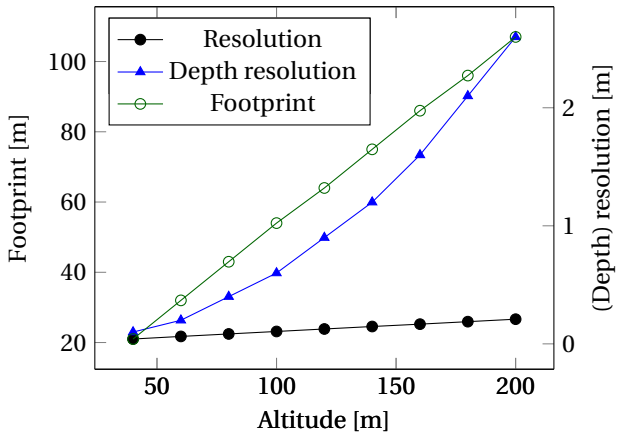


Figure 3.10: Image resolution and footprint as a function of altitude.

Table 3.2: Camera model

Parameter	Value	Unit
Field of view	30	°
Image size	512 × 512	pixel
Baseline	2	m
Sampling rate	0.8	Hz

3.7.3. SIMULATION ENVIRONMENT

The complete algorithm is tested in a simulated landing environment. As described in the previous section, both the IMU measurements and the camera images are simulated. Camera images are pre-generated using the PANGU utility. All DEMs used in the filter were computed using the algorithm presented in Chapter2. They do therefore contain reconstruction errors and are not perfect. It is to be expected that the DEMs used during HILT will contain comparable errors. IMU measurements are computed using the real state and the errors as presented in the previous section. The measured acceleration and angular rate are described by Eqs. (3.3) to (3.6), which are repeated for completeness:

$$\boldsymbol{a}(t)_m = \boldsymbol{a}(t) + \boldsymbol{b}_a(t) + \boldsymbol{n}_a(t)$$
$$\boldsymbol{\omega}(t)_m = \boldsymbol{\omega}(t) + \boldsymbol{b}_g(t) + \boldsymbol{n}_g(t)$$

where $\boldsymbol{n}$ is the Gaussian white noise and $\boldsymbol{b}$ is the bias. The bias is described by a random walk process:

$$\dot{\boldsymbol{b}}_a(t) = \boldsymbol{n}_{ba}$$
$$\dot{\boldsymbol{b}}_g(t) = \boldsymbol{n}_{bg}$$

The bias used for integrating the IMU in the propagation step can be found by integrating these equations. The values for these errors are given in Table 3.1.

A trajectory from 200 m downwards is simulated. This trajectory is computed based on the E-guidance law (Cherry, 1964). This guidance law defines the acceleration as a linear polynomial:

$$\mathbf{a}(t) = \mathbf{a}_c + \mathbf{g} = \mathbf{c}_1 + \mathbf{c}_2 t_{go} = \mathbf{c}_1 + \mathbf{c}_2 (t - t_0) \tag{3.66}$$

Solving the two-point boundary-value problem leads to the following equation for the two coefficients:

$$\begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix} = \begin{bmatrix} -\frac{2}{t_{go}} & \frac{6}{t_{go}^2} \\ \frac{6}{t_{go}^2} & -\frac{12}{t_{go}^3} \end{bmatrix} \begin{pmatrix} \mathbf{v}_f - \mathbf{v}_0 \\ \mathbf{r}_f - \mathbf{r}_0 - \mathbf{v}_0 t_{go} \end{pmatrix} \tag{3.67}$$

where, $\mathbf{v}_f$ and $\mathbf{r}_f$ are the final (desired) velocity and position, in the given case the landing location and $\mathbf{v}_f = \mathbf{0}$, as selected by the hazard-detection system, and $\mathbf{v}_0$ and $\mathbf{r}_0$ are the initial conditions as estimated by the navigation system. The remaining free variable, t_{go}, the time between initial position and final position, can be computed in various ways or could even be guessed. Different choices of t_{go} will lead to different descent duration and trajectories. Since it is not the subject of this work to study E-guidance in depth, the selected method is sufficient to generate a trajectory. In this work the following equation is used (Câmara et al., 2005).

$$t_{go} = 2 \frac{\sqrt{(x_f - x)^2 + (y_f - y)^2}}{\sqrt{(u_f - u)^2 + (v_f - v)^2}} \tag{3.68}$$

where u and v are the x and y components of the velocity $\mathbf{v}$. This is a simple formula based on the distance covered in the xy-plane, divided by the velocity reduction over this distance. A flow-chart of the full simulator environment is given in Fig. 3.11. The figure shows the interaction between the true state, PANGU, the IMU model and the filer to compute a state estimate. The true state is needed to generate the IMU outputs as well as the images needed for the stereo measurement. The filter has no knowledge of the truth, only of the previous state estimate. The current true state $\boldsymbol{x}_i$ is calculated based on the previous state $\boldsymbol{x}_0$, the time step, the environment and the commanded acceleration.

3.7.4. RESULTS

To analyse the performance of the HRN method, the main question is whether or not the HRN method actually improves navigation, namely by being more precise while not being more inaccurate than the benchmark algorithm (thus the conventional approach of IMU propagation)[4]. Moreover, the intention is to also demonstrate the robustness of the algorithm, *i.e.*, none to few outliers are generated, and show that the performance of the algorithm is sufficient for HDA. Here, frequently a precision of 10 m is quoted as being the limit for successful hazard avoidance (Kerr et al., 2013).

[4] *Accuracy* describes the difference to the true solution, *i.e.*, the mean error, while *precision* means how close individual solution are to the mean of all solution, *i.e.*, the standard deviation of the error.

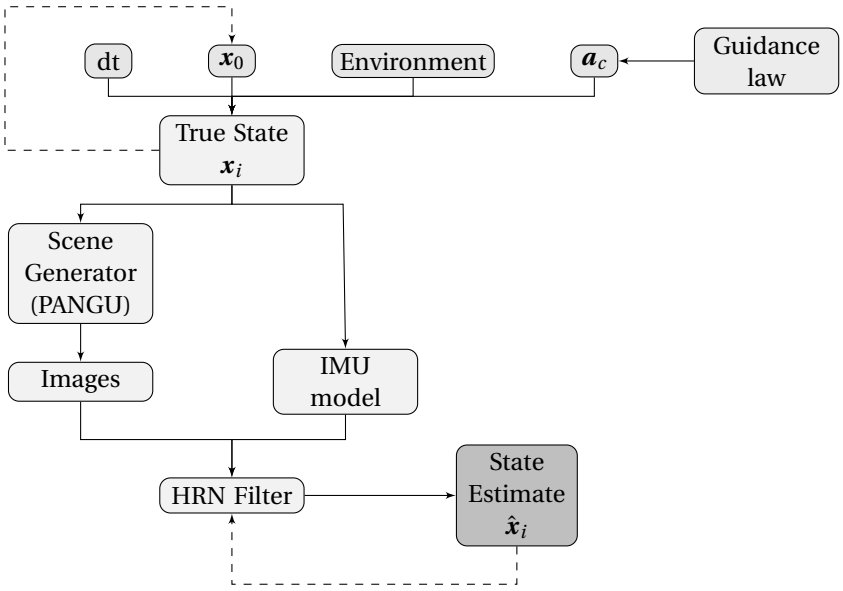


Figure 3.11: Simulator set-up

In the following discussion and the remainder of this work the term error always describes the estimation error. Thus the difference between the real position, in the following referred to as "true state", of the lander and the position estimate of the HRN filter or the benchmark. Only by decreasing this value as much as possible the lander can perform a precision landing and thus touch down at a preselected safe landing site in a hazardous region.

To investigate this, the algorithm is executed 500 times. Each time the initial errors are varied using a uniform distribution and are given in Table 3.3, whereas the initial values for the biases are zero. The quaternion is normalised after the error is added. Also, the IMU performance is different for every run, since it is modelled based on Gaussian noises. Moreover, the starting point for the stereo-measurement is changed such that different images from the data set are used every run.

Table 3.3: Initial error

Parameter	Lower and upper bound	Unit
Position (x, y, x)	$[-20, 20]$	m
Velocity	$[-0.5, 0.5]$	m/s
Orientation per quaternion element	$[-0.1, 0.1]$	–

The resulting mean estimation errors, μ, and their standard deviations, σ, are given

in Table 3.4. These errors describe the difference between the true state of the spacecraft and where the navigation filter estimates the spacecraft to be relative to the origin of the reference frame, which is defined by the initial x- and y- position of the lander and the plane defined by the landing site normal. Note, that the simulations are ended when the true state reaches the landing site, *i.e.*, $z = 0$ and not when the estimated state reaches the surface. From the values given in this table it can be seen that the accuracy is better for the HRN method (the mean error is smaller), but more importantly the precision is a lot higher (meaning that the standard deviation of the error is lower).

This proofs that more accurate navigation is possible with the HRN method and that it can be used for a precision landing. Moreover, the method clearly outperforms the benchmark, which is based on an IMU-only propagation.

Table 3.4: Results of HRN SILT (500 runs).

	μ_x [m]	μ_y [m]	μ_z [m]	σ_x [m]	σ_y [m]	σ_z [m]
HRN	2.88	1.25	0.16	2.45	2.88	0.16
Benchmark	7.25	7.36	12.27	9.11	9.38	15.37

Even though Table 3.4 clearly summarises the performance of the HRN method, some additional figures are given to draw a more complete picture of the performance of the algorithm. One of the main expectations for the filter is that it will enable more precise landings and thereby enabling a safe touchdown within a hazardous area. Figure 3.12 shows the accumulated additional touchdown error in the x-y plane (thus the surface plane) over the execution time of the algorithm. The origin of the graph is defined by the initial conditions of the lander projected on the landing site plane, thus the landing site is at $z = 0$. The selected landing site is indicted by the small star. The estimation results of the HRN and the benchmark are represented by the circles and triangles, respectively. As discussed before, the HRN cannot remove any initial position estimation error. Therefore this error is subtracted from the final error as only the additional accumulated estimation error relative to the landing site is relevant in this work. It can be seen that the HRN function has very little spread with only 4 outliers on a total of 500 runs, thus less than 1%. This figure also shows the 3σ landing ellipses as computed from the navigation accuracy. The benchmark ellipse is approximately 60×60 m and the HRN landing ellipse is 20×20 m. This means that a potential landing site should stay 10 m clear of any possible hazard. Moreover, this is in line with the accuracy of 10 to 20 m desired for TRN methods and 10 m precision necessary for successful hazard avoidance (Kerr et al., 2013). This leads to the conclusion that landing safely within the constructed hazard map is possible.

From the numbers given in Table 3.4 it can be concluded that the algorithm's performance with respect to landing accuracy is improved by a factor of 2.5 in the x component, 5.8 in the y component and a factor of 77 in the z component. Moreover, the precision, *i.e.*, mean error, is improved by a factor of approximately 3 in both x and y-direction, and a factor of 96 on the z component (which is an almost 99% reduction of the error). Overall, this shows that the HRN method will lead to more accurate and more precise landings than the benchmark.

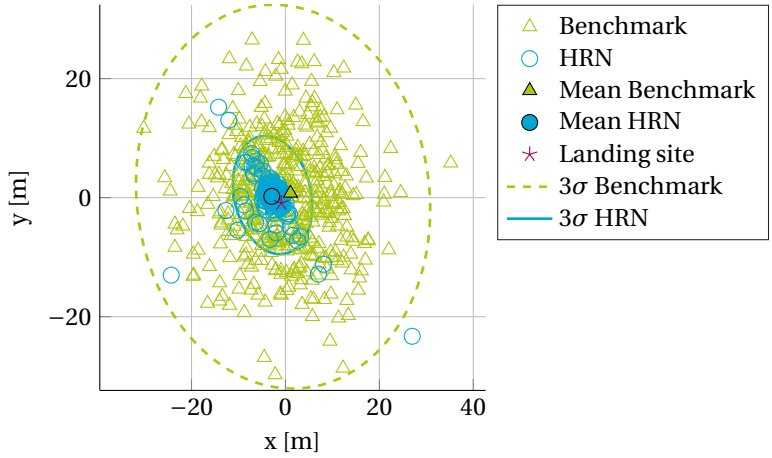


Figure 3.12: Results of the HRN SILT (500 runs), $x-y$ plane.

However, not only the performance in the $x-y$ plane should be investigated, also the z-component should be analysed. As discussed before, the algorithm is capable of removing the final error in the z direction due to the absolute line-of-sight measurements obtained from the stereo DEMs. Therefore, for the z component not the error accumulated relative to the start point is investigated, but the absolute error. Figure 3.13 shows the results of all 500 runs. It can be seen that the algorithm performs extremely well in removing the initial error in z-direction, and as already indicated in Table 3.4 by the very low standard deviation, the mean error is very low.

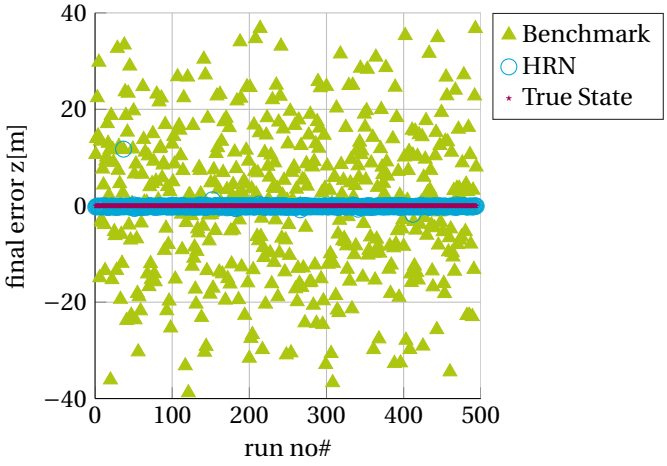


Figure 3.13: Results of the HRN SILT (500 runs), z-component.

Lastly, it is also interesting to look at the behaviour of the filter over an individual run, to demonstrate that performance is not random and that the algorithm can indeed

track the true performance. Figures 3.14 to 3.16 present the performance in x, y- and z-direction respectively. Again, note that only the initial error in z can be removed. In the other two directions, it is only desired to limit the error growth with respect to the initial error and thus the hazard maps computed. Only terrain absolute methods are able to remove the initial error in the surface plane (see Chapter 1).

For all three graphs the sampling points are indicated. It can be seen that the method is able to limit the accumulation of additional error in x and y, as well as removing the initial error from the z estimates.

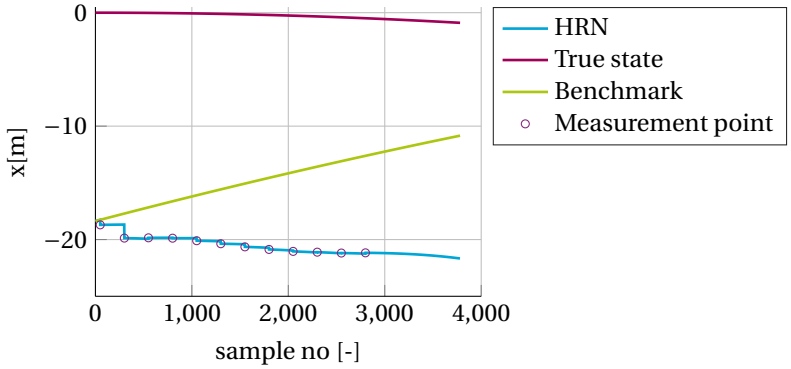


Figure 3.14: Results of a single run SILT, x-component.

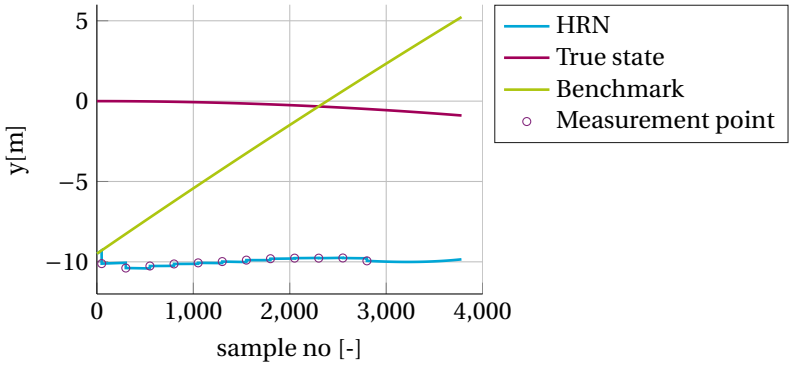


Figure 3.15: Results of a single run SILT, y-component.

Figure 3.17 shows the number of features tracked over time when running the HRN filter. Here, it can be seen that at the eleventh measurement a new set of features is introduced as the number of re-observed features was too low, *i.e.*, fell below 10. Obviously, the point where introduction of new features becomes necessary depends on the size of the initial data set, and the rate at which features are lost. This may be different for different scenarios. The purpose of this plot is to demonstrate that the algorithm can successfully introduce a new set of features, this will not necessarily happen during the

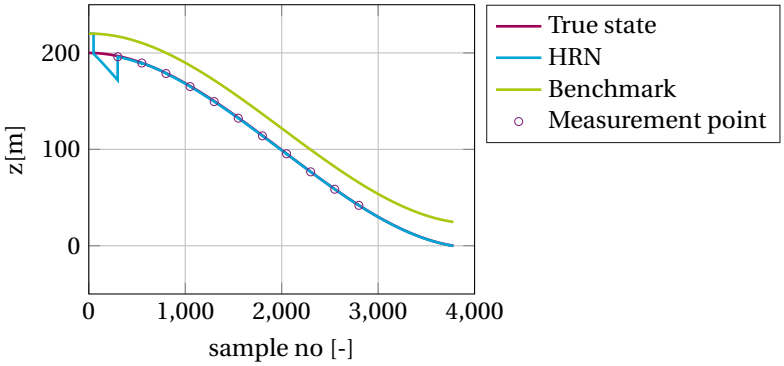


Figure 3.16: Results of a single run SILT, z-component.

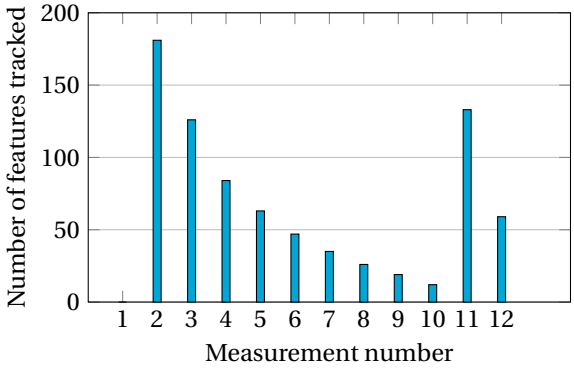


Figure 3.17: Features tracked over time.

eleventh measurement.

3.8. SUMMARY AND OUTLOOK

An HRN filter was developed, which follows the approach of reusing surface DEMs computed from a hazard-detection function to perform a SLAM-like localisation with respect to these hazard maps. In a landing scenario, where hazard detection is employed, this could ensure a safe landing.

As was already discussed in Chapter 1 most other TRN research does not use full 3-D measurements of the features for navigation, but only uses information retrieved from images. This will only help to remove relative error in x and y direction by using velocity measurements retrieved from these images. Due to the unique fact that this algorithm also has knowledge of the z component of the features from the hazard mapping stereo DEMs, the algorithm is capable of also limiting the terrain relative error in the z direction.

Moreover, since both the construction of hazard maps and HRN are performed using the same input images, the navigation is truly hazard (map) relative and can thus

ensure safe touchdown within an on-board hazard map. In other methods, where HRN is performed more independently of the HDA method, map-tie problems might come into play, when linking the HDA results and the navigation outcome.

During SILT testing it was shown that the algorithm is robust against outliers and can achieve more precise and more accurate touchdowns than the current state-of-the-art. Overall, the hazard-relative landing ellipse size was reduced by a factor of 3. The final hazard-relative landing precision achieved is 10 m (3σ), which will be sufficient for successful hazard avoidance.

During the SILT testing, it was assumed that the camera images are error free. It will be part of the HILT to study how the presence of errors in the images as well as calibration problems will effect the outcome of the HRN method. Only very few filter executions lead to outliers, *i.e.*, solutions that were larger than the benchmark. HILT will show if this is also the case when using real images. Since simulating camera images and artificial surfaces is not trivial and has its limitations, for example with regards to resolution and surface file size, it is very difficult to perform SILT for image-based methods that is fully representative of reality. HILT, camera-in-the-loop testing, of such methods is thus always necessary.

The next chapter will present the hardware-in-the-loop testing of the HD function and the HRN algorithm, to address these questions. Moreover, these tests will be used to verify the findings from the SILT testing presented in this chapter.

4

HARDWARE-IN-THE-LOOP PERFORMANCE EVALUATION

DESIGNING algorithms does not simply end by showing once that the method does what it was set-up for. Thorough testing of the final product is needed to show its robustness and judge its overall performance. This testing should be as close to the real use-case of the method as possible. For space applications this is a real challenge as no (or few) opportunities for testing new methodologies in-flight exist. Therefore, testing is usually done in a software simulator, trying to mimic the real environment.

More realistic testing should, of course, always be preferred over pure software simulations. However, this kind of testing often allows for one fixed scenario only, and might be too expensive and time consuming to be performed multiple times during the design.

Therefore, often the design is done in a software simulation environment, while only the final product will be tested using real inputs and hardware. This approach was also followed in this work. Both the HD function and the navigation filter were first tested in a SILT environment, using a MATLAB simulator and PANGU to simulate the camera outputs. Only after successful SILT, a limited and smaller HILT will be performed.

In the previous chapters, a hazard-detection method, as well as a hazard-relative navigation technique building upon the HD method were presented. For both methods a thorough SILT was performed and presented. This chapter will close the development loop by presenting the hardware-in-the-loop testing of the final HRN method. The results of the SILT showed that both methods are robust and feasible candidates for future missions, while the HILT in this chapter aims to underline this conclusion by showing that the algorithms work using real inputs.

To start with, the used testing facility is described in Sec. 4.1. This is followed by a detailed description of the specific set-up used in Sec. 4.2. In Sec. 4.3 the findings from the HILT of the hazard-detection method are presented, followed by the results of the

Parts of this chapter have been published in the proceedings of the AIAA Guidance, Navigation and Control Conference 2018 and in the proceedings of the International Astronautical Congress 2018.

HILT of the hazard-relative navigation method in Sec. 4.4. Section 4.5 summarises the findings of this chapter and thus concludes this research.

4.1. TESTBED FOR ROBOTIC OPTICAL NAVIGATION (TRON)

The Testbed for Robotic Optical Navigation (TRON) is a HILT facility with the purpose of supporting the development of optical navigation technology at the German Aerospace Center (DLR) in Bremen. TRON provides an environment, which allows for qualifying breadboards to Technology Readiness Level (TRL) 4, and qualifying flight models to TRL 5-6. Typical sensor hardware, which can be tested in TRON, are active and passive optical sensors, like lidars and cameras. The major components of the laboratory are a robot on a rail for dynamic positioning of the sensors tested, a dynamic lighting system for illumination of the targets, laser meteorology equipment for high precision ground truth (referred to as laser tracker in the following), a dSPACE real-time system, a rapid prototyping environment, for test observation, test control, and synchronisation of the ground truth and sensor data. The laboratory can be customised with user-defined hardware, such as models of Lunar or Martian surfaces. Due to this flexibility, as well as its extensive dimensions, TRON is well suited for simulating missions representative of the ones encountered by optical sensors during exploration missions.

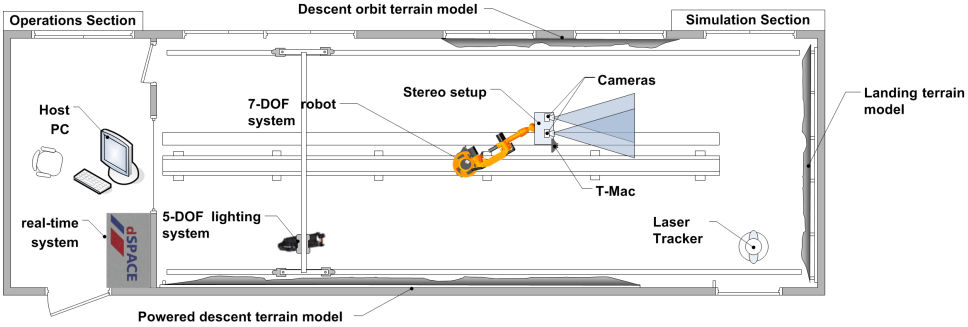


Figure 4.1: Layout of the TRON facility (source: DLR).

As shown in Fig. 4.1, TRON features three different terrain models located at different walls inside the laboratory. Each of these models is designed for a specific phase of approaching and landing on a planetary body. Figure 4.2 shows a picture of the facility. All three terrain models are visible, as well as the robot, the laser tracker and the lighting system. Figure 4.1 also shows the laser-tracker target, the so-called T-Mac, mounted on the payload platform together with the stereo cameras, as used during the tests presented in this work. The T-Mac serves as a target for the laser tracker to measure the distance from the target to the tracker, as well as its orientation with respect to the laser tracker. From these measurements, the camera position and orientation in the global laboratory reference frame can be determined. Moreover, this information is also needed for referencing the results to the ground truth data.

The landing terrain model (terrain 3), on the right wall indicated in Fig. 4.1 is designed for simulating the last phase of a landing mission. In this way, the terrain can be

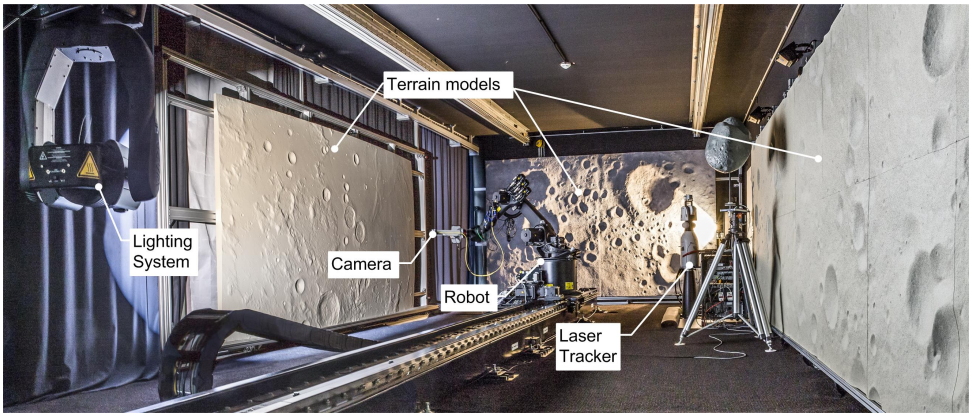


Figure 4.2: Picture of the TRON facility (source: DLR).

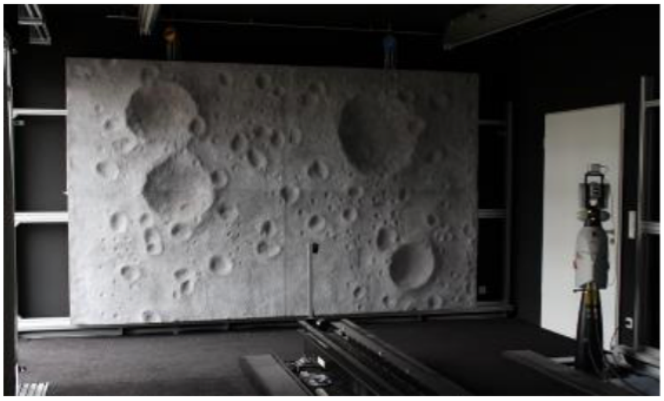


Figure 4.3: Picture of surface model 3 (source:DLR).

used to not only perform hardware-in-the-loop tests of hazard-detection and avoidance methods, but also methods for terrain relative navigation with respect to the landing site. Therefore, terrain model 3 is used for this research. An image of this model is given in Fig. 4.3.

Terrain model 3 is installed at the front wall of TRON. Its size is about 4.2× 2.2 m, the maximum elevation range of the terrain is ≈0.26 m. The model reference data were obtained entirely by DLR via a process beginning with hand-modeling, and ending with 3D scanning and post-processing, as described by Lingenauber et al. (2013). The model was then manufactured in two steps. At first, a coarse milling step obtained the rough terrain structure. Afterwards, a finishing surface layer was applied manually. Due to the hand-made finishing, no manufacture marks such as milling lines are visible, leaving the model with a practically infinite resolution. The self-similarity with respect to scale of the model and the Moon can be exploited by applying a different scale to this model, as will be done in this research. This landing-site model is not only representative for

the Lunar surface, but also for many asteroid surfaces. Combining this model with low scale factors makes it a useful sensor target for 3D imaging sensors. Also, the infinite resolution makes it more flexible and superior to the PANGU images used during SILT.

Moreover, a full ground-truth model of the surface was recorded during this research. Its location is precisely known within the laboratory reference frame. This enables a very accurate comparison of simulation results to reality. The ground truth of Terrain 3 is presented in Fig. 4.4.

The robot, carrying the (optical) sensors, can be moved along a rail from a distance of 11 m to a distance of 1 m to Terrain model 3, as also indicated in Fig 4.1. At the same time, it can be moved with a radius of ≈1 m perpendicular to the rail.

More detailed information on the set-up of TRON and the surface model can be found in (Krüger et al., 2014).

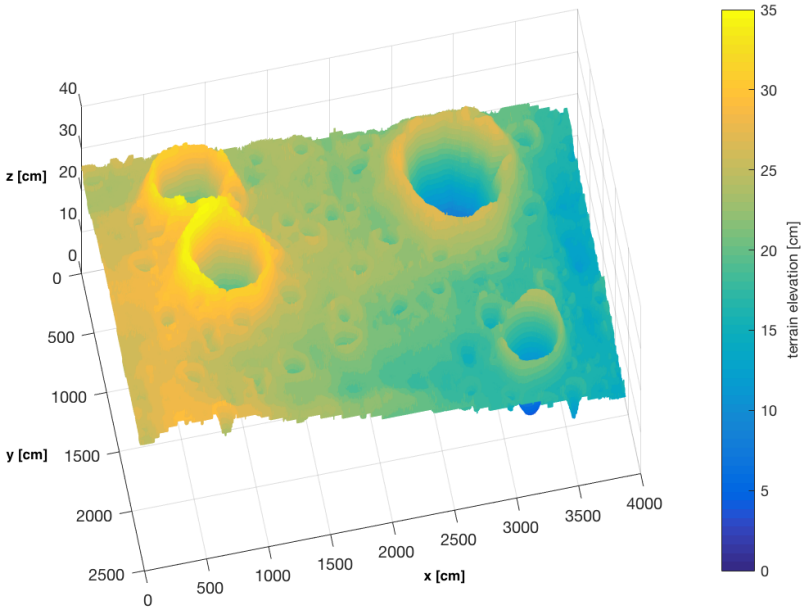


Figure 4.4: Laser scan of Terrain 3, the terrain dynamics in z-direction are exaggerated with respect to the other axes for better visibility.

4.2. HARDWARE-IN-THE-LOOP TEST SET-UP

Building a hardware-in-the-loop test set-up is not trivial for space applications, however, it is especially complex for stereo set-ups.

The problem is that it is (close to) impossible to test the algorithms at full scale. For the given algorithm this would require a lunar analogue surface of at least 110 $m^2$and an altitude of 200 m would result in a camera footprint of approximately 107 m^2 and the

ability to position the cameras accurately at 200 m distance of this surface.[1] Moreover it would be necessary to move the camera closer towards the ground while accurately monitoring its state.

While this is not impossible when sufficient funds are available, it was clearly not feasible within the context of this project.[2] Therefore, an approach, which is easier (and less costly) to implement had to be chosen. Here, an indoor laboratory, where tests could be performed in a scaled set-up, was a feasible candidate. To this end a collaboration with the German Aerospace Center in Bremen was set-up to make use of TRON (Krüger and Theil, 2010; Krüger et al., 2014). In this laboratory it is thus possible to perform hardware-in-the-loop testing of the HRN algorithm in a scaled environment. An accurate ground truth of this surface was recorded in the context of this project to have a reference for the DEMs computed by the HD function.

Since the rail is 10 m long, it was determined that a scale factor of 20 might be feasible to approximate the working envelope of the algorithm. However, since the focal length of the real cameras is different from the simulated cameras in the SILT set-up, the baseline has an additional scale factor, namely 1/3 since the real focal length is a factor of 3 larger than the SILT focal length, to achieve comparable depth resolution. As a result a baseline of ≈0.3 m was selected for the HILT.

One of the challenges during the HILT testing was to generate usable light conditions. In a real Lunar landing scenario, parallel light emitted from a very distant source illuminates the landing region. During laboratory testing, it is impossible to place the light source at a 1 : 20 scale to the Sun-Moon distance (at a minimum of 17.825 km), or anywhere close to the possibility of achieving parallel light. A main focus was thus put on attaining an illumination condition with as little as possible saturation and an even illumination of the entire scene. To this end some shielding was used to create a bit of indirect light. This is not realistic, but neither is a situation where one end of the surface is four times as close to a direct light source than the other end.

During preliminary testing it was proven that the algorithm is very sensitive to very uneven illumination. The situation used in the end leads to successful algorithm performance and was the best that could be reached with the resources at hand. In the case of further work with this algorithm, more investigation might be done into achieving more realistic light conditions. However, the real conditions are very close to what is simulated in PANGU and thus used during SILT. Non of the light-related performance issues detected during SILT will occur during a real flight. The problems encountered during SILT were caused by the close proximity of the light source to the terrain models. In a real landing scenario the Sun will be very far away and will illuminate the scene with parallel, even, light. Therefore, it is concluded that the real lighting conditions will cause no problems for the algorithm.

For testing the stereo-vision algorithm, two cameras (the stereo pair) are attached

[1] Assuming a field of view of 30°, an image size of 512 ×512 pixels

[2] The NASA Morpheus Lander, is an example of a system where an algorithm like the one presented in this work could be tested at full scale. The development of the system till final test flights took four years and the estimated cost was 14 million dollar (see: https://www.nasaspaceflight.com/2015/03/nasa-morpheus-project-templates/ retrieved Feb. 2018)). Note that the test of the ALHAT suite was still performed with a scale factor on Morpheus, nevertheless the work presented here could have been tested at full scale on the vehicle.

to a platform on the robot's arm, see Fig. 4.5. These cameras and the laser tracker are hardware-triggered by the dSPACE system, which enables synchronised image capturing of both cameras, as well as recording the robot/camera position during the simulation. The cameras' full position and orientation with respect to the laboratory reference system is determined by the laser tracker, which measures the T-Mac position and orientation at 100 Hz. The T-Mac is fixed to the camera platform on the robot. This information is necessary for comparison of the simulation results to the ground-truth data, to perform a quantitative assessment of the algorithm's performance.

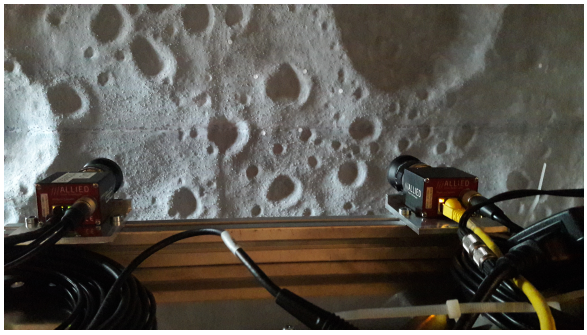


Figure 4.5: Camera set-up on payload platform.

The cameras used are two Allied Vision Proscila GC1380H cameras, both with a Schneider Kreuznach Cinegon 1.9/10 10.4mm-F1.9 lens. The cameras are set to deliver an image of 1024×1024 pixels, which is down-sampled to 512×512 pixels to limit execution time and is thus the same as during SILT. The field of view is ≈37°, which is slightly larger than the ≈30° used during SILT.

Before performing the HILT testing it is also necessary to perform both extrinsic and intrinsic calibration of the cameras. The extrinsic calibration resolves the position and orientation of the cameras relative to each other as well as within the laboratory reference frame, which is necessary for comparing the final results to the laboratory ground truth. The intrinsic calibration is necessary to obtain the cameras' internal parameters (such as focal length, principle point offset, skew and distortion). Intrinsic calibration is very important for stereo applications, since images have to be rectified prior to matching. This requires removing distortion and skew from the images and equalizing the images for principle point locations and differences in focal length.

Both extrinsic and intrinsic calibration can be performed using off-the-shelf software[3]. As an input, images of so-called checker-board calibration targets are needed. Moreover, during calibration the cameras' position and orientation need to be tracked and recorded within the laboratory reference frame. Tracking is performed using the T-Mac and the laser tracker. From these images and state information the software can calculate the camera extrinsic and intrinsic parameters.

[3] In this work DLR's camera calibration toolbox, DLR CalDe and DLR CalLab (Strobl et al., n.d.) was used, but a multitude of others exist.

4.3. HAZARD DETECTION HARDWARE-IN-THE-LOOP TESTING

Chapter 2 presented the successful SILT of the hazard-detection function. In that chapter it was proven that hazard detection is possible at altitudes of 200 m and lower for camera baselines of 2 m. Moreover, it was shown that the algorithm is capable of successful hazard detection on different surfaces.

In this section the HILT testing of the same algorithm is presented. The aim is to show that the algorithm is capable of successfully constructing a DEM of sufficient quality to perform hazard detection using these maps. Also, it is investigated whether the resulting hazard maps (thus the composition of slope, roughness, texture and shadow map) will be supporting safe landings.

TRON features one surface, therefore only one kind of terrain can be used for HILT. Since SILT proved that the algorithm performs equally well, independent from the surface chosen, it can be concluded that it is sufficient to run a test on one surface to determine if the stereo HD algorithm also performs well with real images.

4.3.1. HAZARD DETECTION TEST SET-UP

In a real landing scenario, slopes of more than 15° over the size of the lander footprint (commonly in the order of 2 to 5 m) and roughness of more than 0.3 to 0.5 m, are considered to be hazardous (see Chapter 2). As with the baseline, these values have to be adapted to the scaled environment. For the 1:20 scale used in this set-up, this leads to the following hazard thresholds:

- Slope: ≥15° over a patch of 10 to 25 cm
- Roughness: ≥1.5 cm

Unlike the SILT scenario the lunar-analogue surface does not cover the full image footprint during the laboratory testing. This means that only a subset of the computed DEM actually contains the surface. Only this subset is investigated for HDA purposes (since no ground-truth exists for the laboratory surrounding the surfaces such as the walls, the carpet, the ceiling and other equipment inside the room).

4.3.2. HAZARD DETECTION RESULTS

To elaborate on the performance of the algorithm, first a single data-set is presented and discussed. In this section, the DEM and the hazard map created from pictures taken at approximately 3 m from the terrain model are presented. At this distance, the camera images show only the terrain model. Since a 1:20 scale environment is used, this is equivalent to a 60 m distance (at 2 m baseline). Such a close distance was chosen, as only in close proximity the entire image contains just surface projections, *i.e.*, the picture does not contain any parts of the laboratory apart from the surface. To analyse a complete HD dataset this is clearly the best choice.

After one stereo pair of two images is acquired, these images first need to be rectified, using the results from the intrinsic calibration. As mentioned earlier, calibration was done using the CalDe/CalLab tool based on images of a checkerboard target. Rectification is necessary in the hardware test, opposed to the SILT, as epipolar lines need

to be aligned for stereo reconstruction (see Sec. 2.5). For purely translated cameras, as used during SILT, the epipolar lines are already aligned by default.

Even though re-done multiple times and with great care, camera calibration proved to be a bottle neck in the performance of the algorithm. This was partially due to time constraints, but also because the calibration had to be done on the robotic arm while tracking the cameras in the laboratory systems. This left very little fidelity in the systems and made calibration very difficult.

In this research it was possible to overcome problems caused by slight offsets in the calibration by down-sampling the input images before computing the stereo DEMs. This did not only overcome the calibration problems but also led to a set-up more comparable to the SILT set-up since the down-sampled imaged had 512×512 pixels like the SILT images.

Lastly, linking the ground truth to the computed DEMs proved to be more complex than initially assumed. Especially very small shifts (1 pixels to 2 pixels) can show as quite extreme errors in the final reconstruction while being almost impossible to detect by visual inspection.

Figure 4.6 shows a composition of the rectified right and left stereo image. Careful visual inspection will lead to the conclusion that the pixels belonging to the same physical points are indeed on the same row. This leads to the conclusions that a) rectification worked successfully and b) camera calibration was successful. Careful inspection of the image also show the two orthogonal lines forming a cross on the surface. These are the boundaries of the four individual panels out of which terrain model 3 is assembled. These lines are also included in the ground truth and have a hand finish applied to them (i.e., they are as irregular as the rest of the surface). Therefore, these lines have should not have any influence on surface reconstruction and thus algorithm performance.

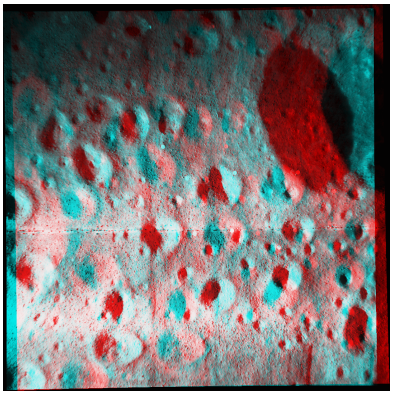


Figure 4.6: Rectified stereo pair (red - left image, cyan - right image).

Figure 4.7 presents the DEM as computed from the input pair shown in Fig. 4.6. The corresponding ground truth is given in Fig. 4.7.

It can be concluded that overall the scene is reconstructed well, for example, all craters are visible in the reconstructed DEM. It is clear that some errors are present in the computed DEM close to the image borders, which is due to the stereo reconstruction

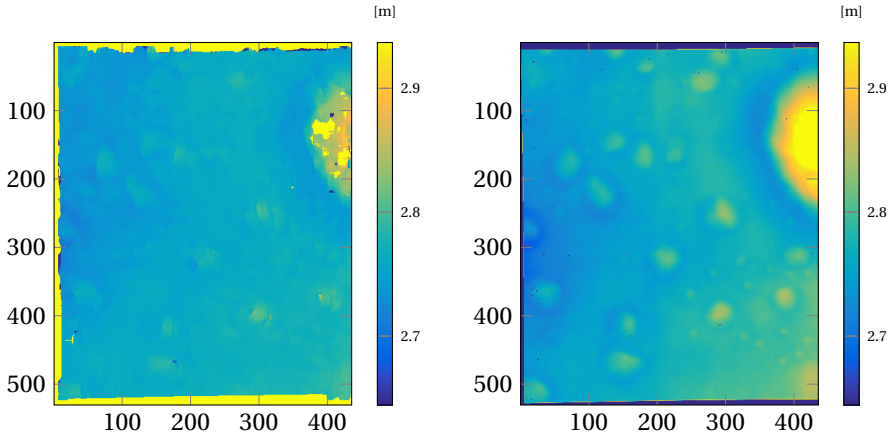


Figure 4.7: DEM from HILT using TRON measurements.

having more difficulties finding the appropriate matches there. However, it should also be noted that these regions are by design more prone to problems with calibration and optics. The extensive testing of the algorithm on a larger number of HILT stereo images showed that insufficient and/or very uneven illumination can also result in reconstruction errors. It is thus very important to create good illumination conditions during HILT. In theory, one would try to achieve a parallel lighting condition. However, this is very difficult to reach in a laboratory where the distance between the light source and the terrain is small and limited by its size. The DEM mean and median error in the z-direction of a rectified Cartesian coordinate system are 0.037 m and 0.066 m, respectively, with a standard deviation of 0.034 m and a median absolute deviation (MAD) of 0.016 m. The error histogram in Fig. 4.8 shows that the error is not completely Gaussian, which is why it was chosen to give both the mean and the median. These error values permit successful hazard detection as envisioned in this work.

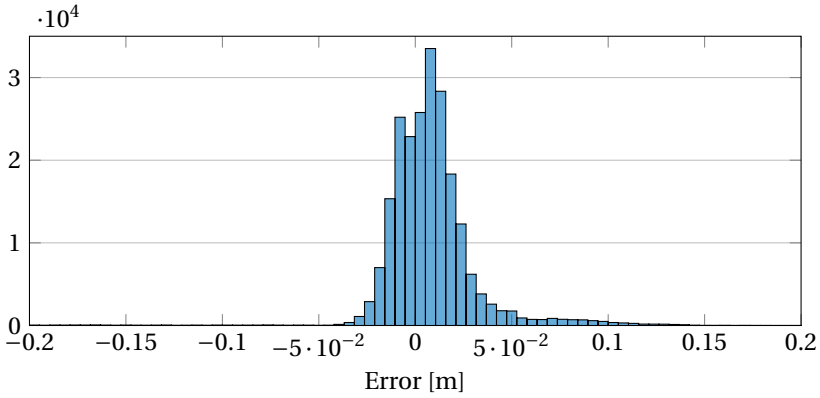


Figure 4.8: Error histogram of the DEM error in z-direction in rectified Cartesian coordinates (HILT).

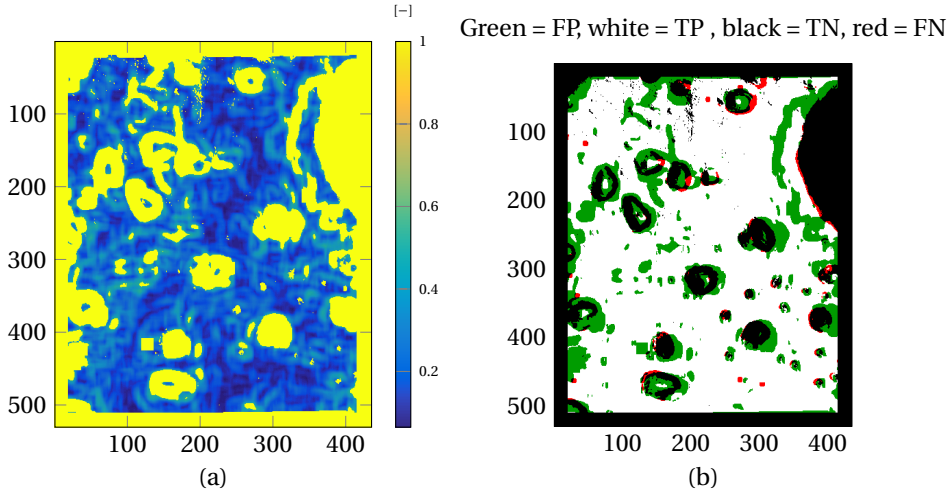


Figure 4.9: Hazard map, and hazard map compared to real hazard map (HILT).

Figure 4.9a presents the computed hazard map based on the output of the algorithm. This hazard map is based on slope, roughness and illumination extracted from both the DEMs and the input image. The hazard thresholds are set as discussed in Sec. 4.3.1 and the hazard maps are constructed as described in Chapter 2. A value of 1 represents a location that would be hazardous. It can be seen that the craters are visible in the hazard map. Also, the higher slope on the left of the DEM is identified and labelled as hazardous. In Fig. 4.9b a comparison of the computed hazard map and the true hazard map is shown. This figure is colour-coded as follows: a red region represents a hazard, undetected by the algorithm (false negative, FN). A green region is a safe site, which is erroneously labelled as unsafe by the algorithm (false positive, FP). A white pixel represents a safe location that is correctly identified by the algorithm (true positive, TP). A black pixel is hazardous and correctly detected as such (true negative, TN). False negatives are the most dangerous errors, as these can lead to a mission loss when selecting a landing site that is actually unsafe.

The scene contains a total of 15% of hazardous and 85% of safe sites. This means that the region over-all offers sufficient safe landing opportunities but is nevertheless too unsafe for a blind landing. The hazard detection percentages are 1.7%, 16.5%, 68.5%, and 13.3% for FN, FP, TP and TN, respectively. Overall, 88.6% of all hazards in the scene are correctly identified, while 80.6% of all safe sites are correctly identified. Concluding, 81.8% of all detections are correct.

Closely investigating Fig.4.9b, one can see that both FN and FP errors are not randomly occurring, but undetected hazards occur in two different situations:

- At locations right next to correctly detected hazards. This could be accounted for by applying a safety factor to not select landing sites right next to detected hazards.
- The algorithm sometimes fails to detect small boulders. This is due to the smoothing effect of the window size used during stereo matching and the comparably

small size of the boulder in the scene (just 2 pixels to 3 pixels each), which makes them very difficult to distinguish from image noise, while in the high-quality ground truth they can be identified very accurately. This problem might be overcome by implementing a more advanced boulder-detection algorithm independent of the DEMS in addition to the roughness detection already included. This way, boulders could potentially be identified more robustly and distinguished from noise.

Figure 4.10 shows the individual contributions that lead to the hazard map as presented in Fig. 4.9a, namely the slope in Fig. 4.10a, the roughness in Fig. 4.10c, with the ground truth given in Fig. 4.10b and Fig. 4.10d, respectively. The texture-based roughness is shown in Fig. 4.10e and the shadow map in Fig. 4.10f, where yellow indicates detected shadows. It can be seen that the illumination conditions during the HILT were chosen in such a way that quite some shadows were present on the model. Moreover, DEM-based roughness (Fig. 4.10d) and texture-based roughness (Fig. 4.10e) show comparable, but not identical, results. Therefore, it is clearly advisable to use both methods simultaneously. This conclusion was also already reached during the earlier SILT evaluation.

Comparing the true slope in Fig. 4.10a to the computed slope in Fig. 4.10b, one can see that the algorithm performs quite well in computing the terrain's slope. As the computed DEMs are slightly smoother than the ground truth DEM, the computed slope is smoother than the true slope as well.

Figure 4.10c presents the true roughness. Comparing this to the computed roughness in Fig. 4.10d, one can see that the overall performance of roughness detection is good for larger-scale features, while the algorithm fails to detect small roughness features, *e.g.*, very small rocks. This behaviour was already addressed above.

Running the algorithm repeatedly for the same same input data will result in identical outputs. This is no surprise as the stereo-vision hazard-detection has no random processes involved.

In summary, the results presented in this section demonstrate that stereo vision is a suitable solution for hazard detection, not only for simulated images, but also when using real images.

4.3.3. HAZARD DETECTION ANALYSIS

To draw more general conclusions on the performance of the algorithm, eight datasets were recorded and used to reconstruct stereo DEMs, as well as the resulting hazard maps. Both DEM errors and hazard-mapping errors were computed for all of theses datasets. The datasets were recorded at terrain distances of 2.5 m, 3 m, 4 m, 5 m, 6 m, 7 m, and 8 m. At a 1:20 scale, as discussed earlier, this corresponds to 50 m, 60 m, 80 m, 100 m, 120 m, 140 m, and 160 m above the planet's surface during a real scenario (at a 2 m baseline).

Figure 4.11 reports the DEM errors of the stereo reconstruction. Both mean, as well as median errors, are given, because very small regions of strong noise may cause strong effects on the mean while the median is affected less by these regions. Therefore, the combination of the mean and the median together allow a more precise assessment of the actual performance of the reconstruction. Moreover, as discussed in Sec. 4.3.2, the error distribution is not fully Gaussian. Based on Fig. 4.11 it can be concluded that the

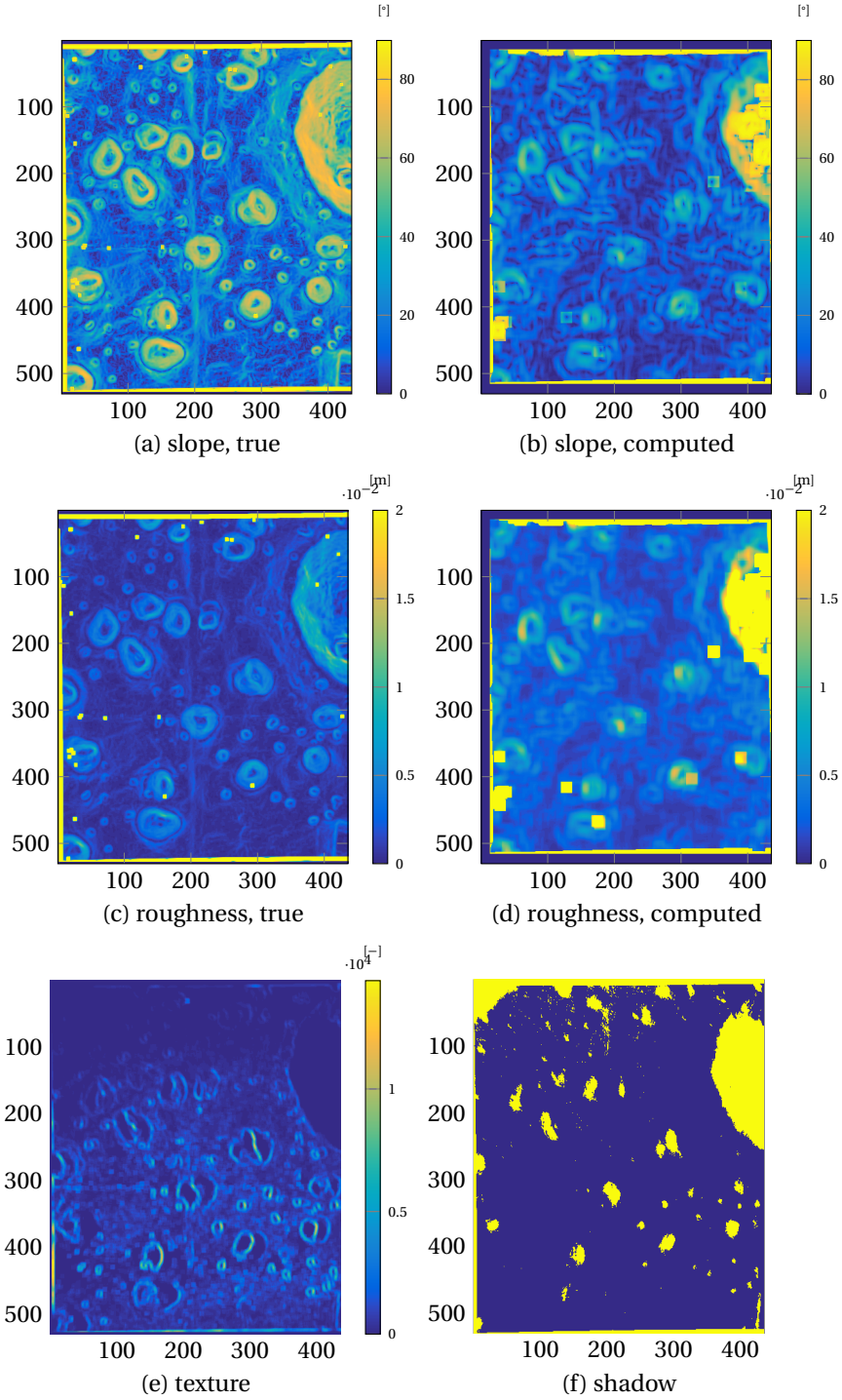


Figure 4.10: Hazard map contributions (HILT).

DEMs are reconstructed accurately enough to perform HD. The large standard deviation for 2.5 m and 5 m distance are caused by some very large errors in a very small region of the DEM (≤10 pixel), most likely due to problems with the lighting conditions of the image. Because of the challenges to set-up the light in a realistic way it was not possible to have no saturation while also not having too much unrealistic shadows. It was thus also not possible to clearly isolate and identify the effects of the light in the reconstruction. Overall, it can be seen that the reconstruction improves if the cameras are moved closer to the surface, which is to be expected as depth resolution is a function of the square-root of the distance to the surface.

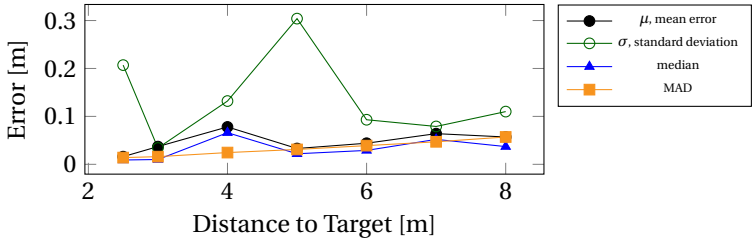


Figure 4.11: DEM mean error and standard deviation (HILT).

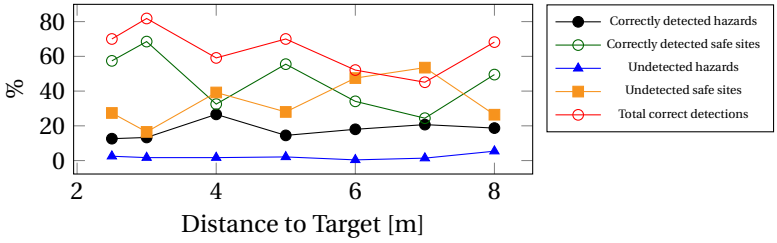


Figure 4.12: Detection rates (HILT).

Figure 4.12 shows the percentages of correctly detected hazards, correctly detected safe sites, undetected hazards, and undetected safe sites. Moreover, the sum of all correct detections is given. It is important to note that there are never more than 2.5% undetected hazards in the scene. This means that the probability of a safe landing based on these hazard maps is always greater than 95.6% (when randomly picking from all sites that are assumed to be safe), assuming that the function selecting the landing site will only select FN sites proportionally to the size of the DEM/hazard map. It has to be noted that this is higher than the requirements stated in Chapter 2.1, where less than 1% of undetected hazards are required. Since so far only a single scene was tested and not all HILT-related error sources were studied to sufficient depth, this is not a killer-criterion, yet. We are confident that more detailed investigation of the HILT-specific errors will lead to a further reduction in the undetected hazardous sites, as was demonstrated during SILT (see Chapter 2.6). Unfortunately, a more detailed study of this was not possible within the scope of this thesis work.

Also, it is also clearly noticeable that the algorithm does miss a substantial number of safe sites in the scene. This can be seen as a safety factor, as these areas largely correspond to regions that are close to unsafe (*e.g.*, containing large slopes slightly below the threshold) and are close to regions, which are actually unsafe. Again, performance does improve when moving closer to the surface, as the DEMs are more accurate at smaller distances.

Lastly, Fig. 4.13 shows the percentage of all hazards detected. It can be seen that this number is always above 75%, but there is no clear trend in this performance.

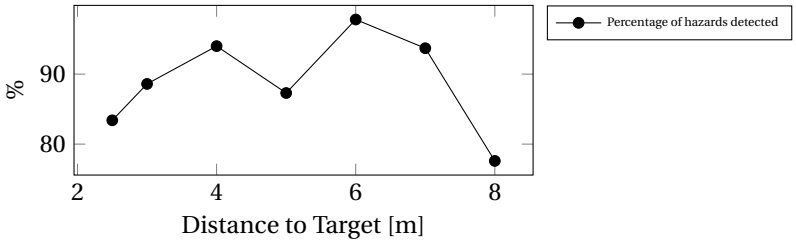


Figure 4.13: Percentage of hazards detected of all hazards present in the scene (HILT).

Overall, this analysis demonstrates that HDA is possible with the result of a stereo-vision surface-reconstruction. Therefore, the HILT supports the conclusion from the SILT performed previously, that stereo vision is a feasible candidate for on-board hazard detection.

It is a known fact, that stereo reconstruction becomes challenging – if not impossible – for scenes with little to no texture. During the laboratory experiments at the TRON facility, this behaviour was demonstrated on the black wall behind the terrain (see Fig. 4.14, the area highlighted by the red box), as well as on the very smoothly painted ceiling of the laboratory (see Fig. 4.14, the area highlighted with the red ellipse), whereas reconstruction on the strongly textured carpet works well (indicated with blue ellipse). This demonstrates that the lunar analogue terrain has sufficient texture for conducting stereo-based hazard-detection. This supports the assumption that stereo-based hazard-detection is possible for lunar landings.

It comes as no surprise that performing hazard detection on real images is more challenging than when using artificial imagery, such as those presented in Sec. 2.6. It is also far more challenging to trade the robustness to noise against the ability to resolve very small features in the scene.

4.3.4. HAZARD-DETECTION CONCLUSIONS

Based on the hardware-in-the-loop testing it is possible to confirm the conclusions obtained during software-in-the-loop testing: stereo vision is a capable
method for performing on-board hazard detection during planetary descents.

It was demonstrated that for all test cases considered, successful hazard detection was performed. As expected, the algorithm performs better, the closer the cameras are to the terrain. Both mean and median DEM errors always remain below 10 cm. The algorithm always detects more than 75% of all hazards. Overall, the algorithm tends

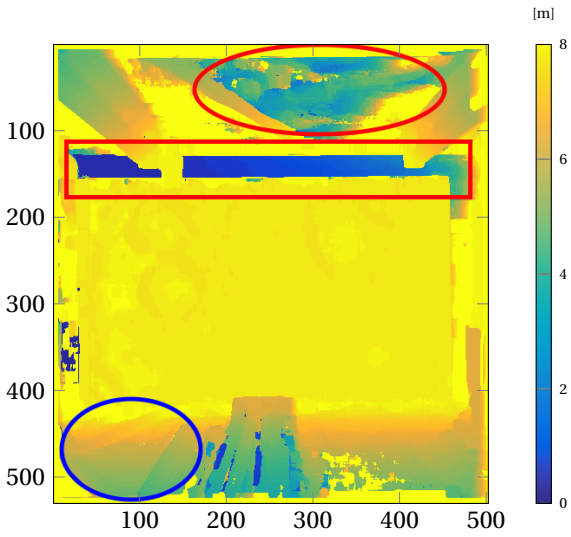


Figure 4.14: DEM of the surface including parts of the laboratory surrounding. It can be seen that sufficient texture is needed for stereo reconstruction.

to overestimate the number of hazards, which serves as a safety factor. These findings support the feasibility of stereo-based hazard-detection for planetary landing.

Moreover the successful test proved that the set-up was designed well and thus as a next step the full hazard-relative navigation function could undergo HILT testing.

As expected the SILT results were slightly better since they were obtained under more optimal conditions: no noise, no misalignments and perfect calibration. Also perfect linking of the results and the ground-truth were given during SILT. Still, considering all these added difficulties the HILT test showed very good results making stereo-HD a feasible solution for future missions.

Obviously, larger baselines would lead to better results, but would not be representative of the baseline possible during a real mission. Smaller baselines would lead to worse results. If the baseline would be inaccurately calibrated, this would always lead to wrong reconstructions. The set-up is reasonably robust to errors in the translation of the two cameras, which would simply lead to a small error in scale. However, the algorithm is sensitive to errors in orientation. During HILT is was shown that calibration is accurate enough to determine the camera orientation; such that stereo reconstruction is possible. If extrinsic calibration would fail, stereo hazard-detection is impossible to perform. A thorough re-calibration plan should be developed for a mission using the HD function, to ensure that the camera extrinsic (and intrinsic) calibration is still correct.

Also rovers use stereo set-ups for navigation and path planning on the surface. Since these cameras can still be used after touch down, it is assumed that a stereo set-up will not loose its calibration during a landing. This assumption should be thoroughly tested prior to a real mission, though.

4.4. HAZARD-RELATIVE NAVIGATION HARDWARE-IN-THE-LOOP TESTING

After establishing that stereo hazard-detection is possible with the HILT inputs, the next and final step could be taken, the HILT of the full hazard-relative navigation algorithm.

4.4.1. HAZARD RELATIVE NAVIGATION SET-UP

To test the HRN performance using the laboratory data, the same TRON set-up as for the HD test is used. However, when testing the HRN not only the individual data-sets are important, also the trajectory information is needed. Here, an almost vertical descent was used, the trajectory was not computed based on an actual guidance law (even though this is possible), but was entered manually. It is thus assumed that the thrust is only applied in z direction. The IMU output is modelled based on the true measured state scaled with a factor of $1:20$ as discussed earlier. Like for the SILT, the filter runs at 200 Hz and the camera images are sampled at 80 Hz. It would be possible to mount a real IMU on the payload platform in TRON (the payload platform was designed to be large enough to actually fit an IMU next to the stereo set-up). However, at the current stage of the development there would have not been an added value in this. After all, it would make it more complex to compare the SILT and HILT results.

4.4.2. HAZARD RELATIVE NAVIGATION RESULTS

The presentation of the HRN HILT follows the same scheme as the presentation of the HRN SILT, as the same data was created and the same performance characteristics are to be analysed. In total 50 runs are performed.

Like for the analysis of the SILT section, the errors presented in the following represent the difference between the true (commanding-error free commanded) state, so the position where the spacecraft actually is, and the estimates state by the HRN filter, so the position where the GNC system would believe the spacecraft to be. All result are given in a reference frame for which the z-plane is in the landing site plane, with its origin at the initial conditions projected on this plane. The mean estimation errors and standard deviations in $x-$ and $y-$ direction are computed from the error minus the initial error, since this error can not be removed by the HRN system, whereas the z-errors are absolute errors, since the HRN filter can remove absolute errors in this direction.

The HILT results presented in Tab. 4.1 support the findings from the SILT as presented in Tab. 3.4. The HRN filter results in a more precise landing and the touchdown is more accurate with respect to the on-board generated hazard maps than the benchmark solution. Like during SILT, the benchmark is computed from an IMU-only propagation. The accuracies in the xy plane are a factor of 7 and 3.5 more accurate, while the surface normal component (z) is 7.7 times as accurate. For the SILT these were factors of 2.5, 5.8 and 77, respectively. The large difference in the z component is caused by a systematic offset not present during SILT, which will be discussed later in this chapter. As with the SILT also the HILT precision is better for the HRN than for the benchmark method. The mean error in x and y are decreased by a factor of 6 and 10.5, receptively, while the z-component is more precise by a factor of 47, an almost 97% reduction. During SILT the x and y precisions were improved by a factor of 3, while the z component showed

Table 4.1: Results of HRN HILT

	μ_x [m]	μ_y [m]	μ_z [m]	σ_x [m]	σ_y [m]	σ_z [m]
HRN	1.38	2.29	1.70	1.55	0.98	0.35
Benchmark	9.97	8.41	13.10	10.37	10.34	16.55

almost identical improvements with 99% reduction. This shows that the algorithm can replace a landing altimeter. The HILT results in the $x-y$-plane are thus even better than the SILT results, which is likely due to the infinite resolution of the laboratory surface opposed to the PANGU surface, therefore measurements from the HILT DEMs are more precise than the SILT DEMs. This will lead to more precise state estimation and thus more precise final touchdowns.

The accumulated touchdown error from the initialisation of the filter until the termination of the filter (touchdown) in the $x-y$ plane is shown in Fig. 4.15. The higher precision of the algorithm, as well as the better accuracy can be clearly seen in the figure. For the low number of runs, the algorithm did not generate any outliers at all. During the SILT it was found that overall less than 1% of outliers occurred. On a total of 50 runs, it is thus realistic that no outlier occurred.

Similar to the SILT, the 3σ landing ellipse is approximately 60×60 m for the benchmark (pure IMU propagation). However, the HRN ellipse is considerably smaller than during SILT (in the order of 9×6 m). There are two possible causes for this: first of all, the surface used during HILT has infinite resolution, whereas the surface used during the SILT has a limited resolution. It is possible that the limited resolution during SILT leads to more inaccurate predictions in the $x-y$ plane. Second, the trajectory used during this HILT is a pure vertical descent, while the SILT trajectory also has contributions in the $x-y$-plane. Potentially, it is more difficult for the filter to estimate these movements. Both hypotheses could be tested, the first by using higher resolution reference DEMs during the SILT, and the second by recording data sets using different trajectories in TRON. Unfortunately, neither input data was available and the generation was either not possible or not possible within the available time frame of this research. Concluding, this shows that the SILT should have been performed with higher resolution DEMs. However, this would require a different generator than PANGU, or a newer version than the one used in this research, as it was not possible to generate even larger DEMs in PANGU than those currently used.

Looking at the final error in the z direction in Fig. 4.16, one can see that there is a small systematic error of approximately 1.5 m to 2 m. During SILT this was not detected, however, during SILT the origin of the surfaces was also better aligned with the mean of the surface, while the surface origin in TRON is at/below the deepest point of the terrain. It is very likely that this is the cause of the slight offset. In Tab. 4.1 the z accuracy reported (1.70 m) is also higher than the accuracy found during SILT (0.16 m, see Tab.3.4). This is clearly caused by this systematic error.

Figures 4.17 to 4.19 show the position estimation performance for a single run. It can be seen that the algorithm limits the error growth in both x and y, and is successful in removing the initial error in z, while also not accumulating new additional errors.

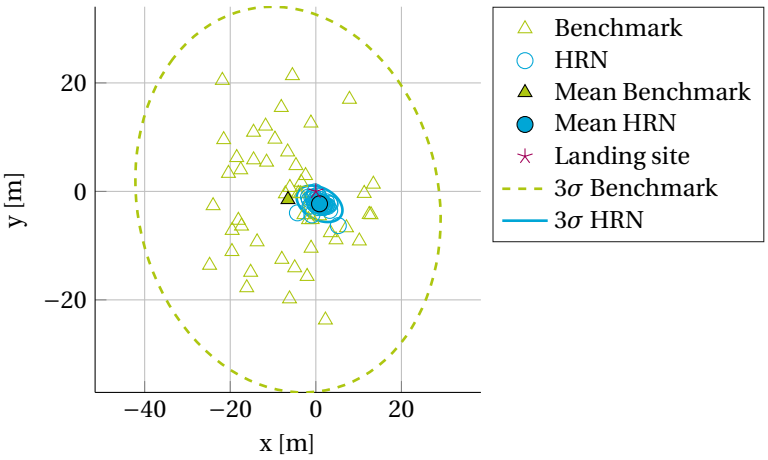


Figure 4.15: Results of the HRN HILT

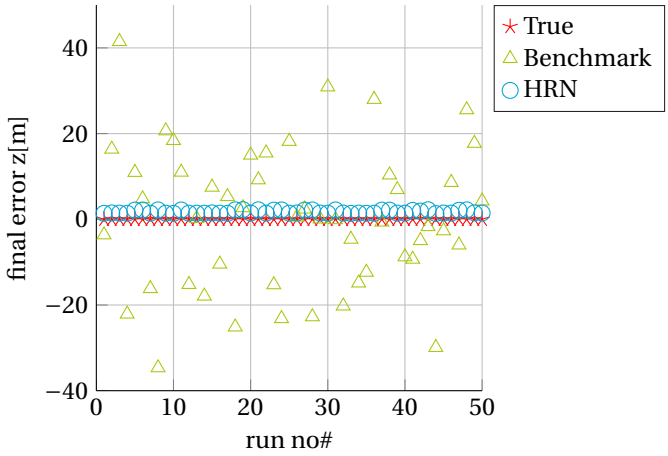


Figure 4.16: Results of the HRN HILT

4.5. SUMMARY AND OUTLOOK

Using the TRON facility at DLR Bremen, hardware-in-the-loop testing of the stereo-vision hazard detection and the full hazard-relative navigation filter were performed. The tests were performed at a 1 : 20 scale featuring a stereo set-up with 0.30 m baseline. A lunar analogue surface of infinite resolution was used as a target.

The HILT testing of the hazard-detection method showed that the algorithm is capable of reconstructing the scene sufficiently well to detect hazards and select a safe landing site. The remaining undetected hazards are slightly higher than during the hazard detection sensitivity study as presented in Chapter 2, which is likely also still linked to very slight inaccuracies in linking the ground-truth information and the computed DEMs/hazard maps. Moreover, the TRON terrain contains a lot more very small boul-

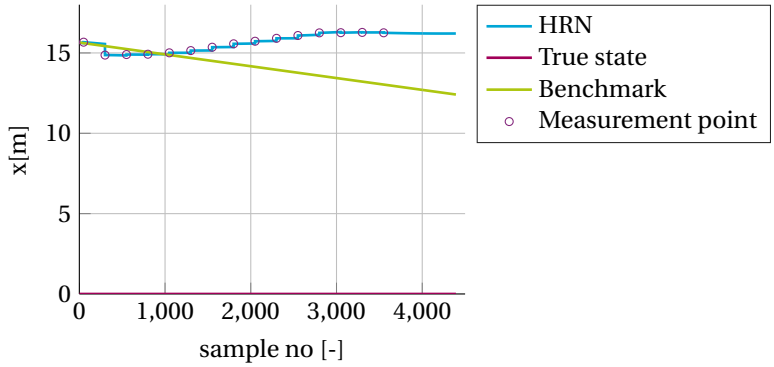


Figure 4.17: Results single run of the x-component during HILT.

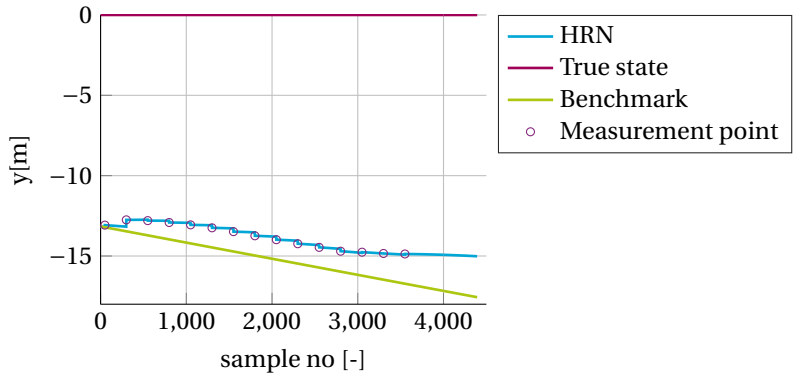


Figure 4.18: Results single run of the y-component during HILT.

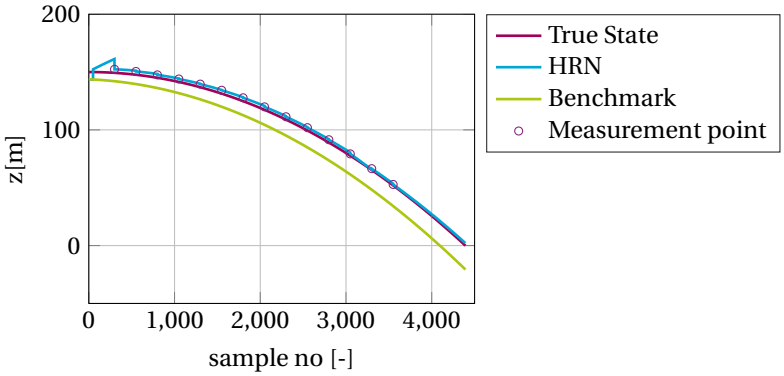


Figure 4.19: Results single run of the z-component during HILT.

ders opposed to the surfaces generated in PANGU. During SILT, no more than 1% of all

sites were wrongly labelled as safe, while during HILT this increased to 2.5%. Nevertheless, this is sufficient for successfully selecting a safe landing location.

The HILT testing of the hazard-relative navigation algorithm also successfully validated the findings from the SILT presented in Chapter 3. Due to the infinite resolution of the surface used during HILT, even better accuracies and, more importantly, precisions were achieved than during SILT. The final hazard-relative landing ellipse is 6 m × 9 m, being an improvement of more than a factor of 2 with respect to the SILT results (20× 20 m). Like during SILT, the altitude proofed to be very accurate and precise. No outliers were detected during HILT.

Overall, it is now possible to conclude that both the developed HD and HRN methods are feasible candidates for enabling future missions to land safely in unknown hazardous environments. This concludes this research.

In the next chapter the complete conclusions and recommendations for future work will be presented.

5

CONCLUSIONS AND RECOMMENDATIONS

In this work two contributions to enable next-generation landing missions were presented: a hazard-detection method based on a pair of stereo images as input and a hazard-relative navigation method utilising the same stereo maps. As a result, this makes it possible for landing vehicles to autonomously detect hazards and perform a precise landing relative to these detected hazards. Both systems were successfully tested by software-in-the-loop testing, and also during hardware-in-the-loop testing using real stereo images of a Lunar-analogue surface. All performed tests supported the feasibility of the proposed methods for future missions.

Section 5.1 presents the conclusions from this work while Sec. 5.2 states the recommendations for future work.

5.1. CONCLUSION

This work tried to answer the following question:

> Are autonomous safe landings in hazardous and potentially unknown environments possible?

To answer this question, two sub-questions have been addressed:

1. How can a landing vehicle autonomously assess the safety of a potentially unknown and unmapped landing site?

2. How can a landing vehicle ensure a safe touch down, avoiding autonomously detected hazards?

In the process of answering the research questions, an autonomous, on-board hazard-detection function to assess the safety of a landing region was developed. Moreover,

precise landing was achieved by designing a hazard-relative navigation function. The following detailed achievements were reached:

1. Development of a hazard-detection function
2. Sensitivity study of the hazard-detection function
3. Development of a hazard-relative navigation function
4. Software-in-the loop testing of the hazard-relative navigation function
5. Set-up and execution of hardware-in-the-loop testing of both the hazard-detection function and the hazard-relative navigation function

In the following sections the findings with respect to hazard detection (sub-goal 1 and 2 and 5) and hazard-relative navigation (sub-goal 3, 4, and 5) are presented.

5.1.1. HAZARD DETECTION

The first part of this research required the development of a hazard-detection method to enable autonomous landings in unsafe terrain. Due to their budgetary advantages it was decided to use a camera-based method for hazard detection. Moreover, using camera-based techniques and thus creating hazard maps with a direct link to images of the landing region was a crucial element for the development hazard-relative navigation method.

As a next step, the three feasible camera-based candidate methods capable of constructing dense hazard maps were studied, namely stereo vision, stereo-from-motion, and shape-from-shading. The former one employs a set-up of two cameras while the latter two methods use just one camera.

From the study of these three methods, stereo vision was found to be the method best suited for the described use-case: hazard detection at low altitudes to detect both slopes, but also hazardous boulders. It was found that shape-from-shading was more suitable for detecting mainly slopes at higher altitudes, while stereo-from-motion showed to be problematic for many aspects. For example, stereo-from-motion requires a very accurate knowledge of the spacecraft state and cannot reconstruct the scene in the middle of the DEM during (near-) vertical descents, as likely encountered during the final descent phase of a mission. Stereo vision showed limitations with regard to the maximum altitudes it can be used at. However, for the low altitudes needed for detecting hazardous boulders in the landing region it is sufficient.

As a next step the stereo-based hazard-detection functionality was developed further, with a detailed study of how to generate a hazard map based on the DEM obtained from the stereo-matching. As a result, a hazard-detection function, computing slope and roughness maps from a local mean plane of the DEM, as well as texture and shadow maps from the input images, was developed.

Following this, a sensitivity study investigating different landing-site geologies, different baselines and varying altitudes during software-in-the-loop tests was executed.

From this analysis, it was concluded that hazard detection at altitudes of 200 m and below are possible at baselines of 2 m and more. A 2 m baseline was deemed feasible

based on current lander designs (even baselines of 2.5 m to 3 m would be feasible on landers like MSL or the ESA Lunar Lander). In this set-up, the percentage of undetected hazards will stay below 1% as required by former ESA studies. Previous developments of stereo-based hazard-detection methods were not able to achieve hazard detections at altitudes larger than 100 m. Therefore, this work was able to double the performance envelope of this method.

After having proved that the method works as desired and can successfully construct hazard maps to enabling the on-board and autonomous selection of safe landing sites in unsafe landing regions, the next step had been to establish the working envelop of the method. After completion of both tasks, it was possible to perform hardware-in-the loop testing of the method to confirm the findings from the software test.

During hardware-in-the-loop testing, the findings from the previous software-in-the-loop tests were confirmed. Slightly higher percentages of undetected hazards were found ($\leq 2.5\%$). However, these are very likely linked to remaining problems in referencing the results to the ground truth. Still, the overall performance was found to be good enough to be the first step to enable safe autonomous landings in hazardous terrain.

Recommendations for on-board hazard detection resulting from the hardware-in-the-loop testing are discussed in Sec. 5.2.

5.1.2. HAZARD-RELATIVE NAVIGATION

The next step to enable safe autonomous landings is to enable precise touchdown relative to the detected hazards.

Here, the idea was to follow a novel approach of using the hazard-detection output, *i.e.*, the DEMs in combination with the image, as a measurement source for a relative navigation function and thereby enable more accurate and more precise landing relative to the hazard maps. This will thus enable hazard avoidance during a landing.

This approach of linking stereo-vision hazard detection and relative navigation for improving the final landing precision of a planetary lander presented in this work is investigated for the first time. Therefore, the aim is to present the feasibility of this approach rather than the development of a fully validated flight-ready filter.

There have been developments to improve the navigation accuracy using images of the terrain in so-called terrain relative navigation methods. However, these method obtain measurements by comparing the images (or information extracted from these) to *a-priori* information. In this work, landing in potentially unknown environments are considered. For such environments no or only very limited information exists.

As this implies that the proposed method was not able to obtain measurements by comparing image information and a so-called catalogue, another strategy had to be developed. Here, a technique frequently used in robotics, simultaneous localisation and mapping, was chosen to use image information for more accurate navigation.

The state estimation and measurement handling is done by an error-state Kalman filter, an approach chosen for its robustness and simplicity, since it overcomes the non-linearity of the problem by estimating the linear errors rather than the non-linear state.

Measurements are obtained by tracking SURF features over multiple descent images and linking these features to their 3-D information via the stereo hazard-detection DEMs. This is the point where the navigation becomes truly hazard relative. In a SLAM-

manner these features are not only used for a measurement once, but are appended to the state and are thus used for multiple measurements. This improves the feature knowledge, which in turn improves the measurement and thus improves the state prediction.

Like the hazard-detection function the navigation filter was first tested using an extensive SILT approach. Here, 500 runs of the full descent were simulated, varying the initial parameters and the IMU errors. These tests have clearly proven the robustness of the methods. In total less than 1% of outliers were generated, and these few outliers did not lead to performance worse than the benchmark. Moreover, it was shown that the method leads to improvements in the accuracy and the precision of the results. The size of the 3σ landing ellipses was reduced by a factor of 3, from 60×60 m to 20×20 m, and thus satisfying the often mentioned 10 m to 20 m relative accuracy for terrain relative methods. In the z-direction (thus normal to the surface) the method is capable of not only removing the relative error, but can remove absolute error and is thus capable of replacing an altimeter. Here the accuracy was improved by more than a factor of 75 and the precision was improved by almost a factor of 96, being a reduction of almost 99% with respect to pure IMU-propagation, resulting in final accuracies in the order of $0.16\,\text{m}\pm 0.16\,\text{m}$.

The next and final step in the development was then to verify the findings from the SILT in a hardware-in-the-loop test. The same laboratory and set-up as used for testing the hazard detection was used. Due to the larger complexity of these test a smaller sized analysis than for the SILT testing was performed. 50 runs were executed.

It was found that in the xy-plane the HILT results were consistently exceeding the results from SILT, which means that even better final accuracies and precisions than found during SILT are possible. The final landing ellipse found during HILT was 9×6 m, which is less than half of what was found during SILT. This additional improvement was caused by a finite surface resolution opposed to infinite surface resolution used during HILT. Overall, this leads to the conclusion that the final accuracies can be even better than those found during SILT. However, already the accuracies found during SILT would suffice to successfully avoid hazards. The precision on the altitude estimation was slightly worse during the HILT testing. Nevertheless, very accurate altitude estimations were achieved.

5.2. RECOMMENDATIONS FOR FUTURE RESEARCH

In this work the feasibility of the method was successfully proven. However, this does not leave all questions answered and all problems solved. Especially further improving the algorithms' performance by optimising the implementation and making small algorithmic changes are recommended for future work. In the following the recommendations for future work on the proposed algorithms are given, as well as recommendations for setting up similar research.

Constructing landing maps based on successive descent DEMs Based on the SLAM implementation, the location of the surface features used for tracking the vehicle state is known very accurately at the end of the descent. As a next step this knowledge can be used to assemble larger maps of high accuracy of the landing region. These maps can

potentially be a valuable asset for surface mobility after touchdown. These maps may considerably speed up rover operations as safe paths can be planned using these 3-D surface maps. Currently, no "aerial" maps for rover driving are available. Speeding up surface motion, may potentially decrease the time required to traverse on the surface and thus decrease mission cost.

Flight implementation/More optimised implementation This work presented a feasibility study of the proposed method. None of the functionalities developed and tested was so far optimised for computational performance. The next logical step would thus be to take the presented method as a baseline and improve both efficiency and performance. Here one should, for example, investigate a more supervised selection of features such that the filter can run with less features over longer periods. Also the computational efficiency of the filter itself, especially of the matrix inversion involved should be analysed.

Investigate other stereo matching methods Currently block matching, one of the simplest matching techniques, is used for reconstructing the scene. This method proved to be a suitable solution during SILT of the HD system. Other methods seemed to be infeasible due to the high complexity. However, after HILT of the HD system, it was found that aligning the epipolar lines accurately enough can be difficult. Stereo-matching methods do exist that are not only searching along these lines, but more globally and two dimensional, *e.g.*, (Roy, 1999; Hirschmuller, 2005). Since these techniques involve global optimisations they may be considerably slower than the simple block matching employed in this thesis. Still, for a continuation of this work, such methods should be investigated.

Other techniques than KF During the work on the HRN method, it was found that the measurements that can potentially be retrieved from the images and DEMs are very accurate. It might thus be possible to develop HRN techniques that do not need a KF, but could run only based on information retrieved from the images. However, this might not necessarily be a good option with respect to robustness.

Other HD methods With the advent of machine-learning-based image processing, it should be investigated whether this might be a possible candidate for hazard detection. If so, this would mean that hazard detection is possible while skipping the DEM-generation step. Still, this may have shortcomings with respect to full DEM reconstruction. Moreover, it would be interesting to study how limiting the required training data sets are for hazard detection using machine learning techniques.

Use of intelligent mean plane when using real images During SILT it was concluded that there was no need for using more advanced slope computation methods. Therefore a mean-plane solution was chosen over a more complex method, as it would not have sufficient benefits to justify the increase in computational load. During processing of the HILT dataset for hazard mapping, it was found that plane fitting was more difficult and noisy on the real images. Thus, an intelligent mean plane would potentially lead to better performance, however, at an increase in computational load. However this may result in

a large study of its own, trading computational load against performance, tuning the intelligent planes, and studying further methods to remove outliers from plane fitting.

BIBLIOGRAPHY

Araki, H., Tazawa, S., Noda, H., Ishihara, Y., Goossens, S., Sasaki, S., Kawano, N., Kamiya, I., Otake, H., and Oberst, J. (2009). "Lunar global shape and polar topography derived from Kaguya-LALT laser altimetry". *Science* Vol. 323, No. 5916, pp. 897–900.

Bailey, T. and Durrant-Whyte, H. (2006). "Simultaneous localization and mapping (SLAM): Part II". *IEEE Robotics & Automation Magazine* Vol. 13, No. 3, pp. 108–117.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). "Surf: Speeded up robust features". *European conference on computer vision*, pp. 404–417.

Biele, J. and Ulamec, S. (2008). "Capabilities of Philae, the Rosetta lander". *Origin and Early Evolution of Comet Nuclei*. Springer, pp. 275–289.

Bilodeau, V. S., Beaudette, D., Hamel, J., Alger, M., Iles, P., and MacTavish, K. (2012). "Vision-based Pose Estimation System for the Lunar Analogue Rover Artemis". *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pp. 4–6.

Bilodeau, V. S., Clerc, S., Drai, R., and Lafontaine, J. de (2014). "Optical navigation system for pin-point lunar landing". *IFAC Proceedings Volumes* Vol. 47, No. 3, pp. 10535–10542.

Black, M. J. and Sapiro, G. (1999). "Edges as outliers: Anisotropic smoothing using local image statistics". *Scale-Space Theories in Computer Vision*, pp. 259–270.

Brady, T. and Paschall, S. (2010). "The challenge of safe lunar landing". *IEEE Aerospace Conference, 2010*.

Braun, R. D. and Manning, R. M. (2006). "Mars exploration entry, descent and landing challenges". *2006 IEEE Aerospace Conference*.

Braun, R. D. and Manning, R. M. (2007). "Mars exploration entry, descent, and landing challenges". *Journal of spacecraft and rockets* Vol. 44, No. 2, pp. 310–323.

Brown, M. Z., Burschka, D., and Hager, G. D. (2003). "Advances in computational stereo". *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 25, No. 8, 993–1008.

Bush, G. H. (1989). "Remarks by the President at 20th anniversary of Apollo Moon landing". *Public Papers of the President, Presidential Document No. 1128*, p. 3.

Câmara, F., Rogata, P., Di Sotto, E., Caramagno, A., Manuel, J., Rebordão, B. C., Duarte, P., and Mancuso, S. (2005). "Hazard avoidance techniques for vision based landing". *GNC 2005: 6th International ESA Conference on Guidance, Navigation and Control Systems*.

Carpenter, J. D., Fisackerly, R., De Rosa, D., and Houdou, B. (2012). "Scientific Preparations for Lunar Exploration with the European Lunar Lander". *Planetary and Space Science* Vol. 74, No. 1, pp. 208–223.

Cheng, Y., Goguen, J., Johnson, A., Leger, C., Matthies, L., Martin, M., and Willson, R. (2004). "The Mars exploration rovers descent image motion estimation system". *IEEE Intelligent Systems* Vol. 19, No. 3, pp. 13–21.

Cherry, G. W. (1964). *E Guidance-A General Explicit, Optimizing Guidance Law for Rocket-Propelled Spacecraft, Apollo Guidance and Navigation*. Tech. rep. tech. rep., MIT Instrumentation Lab.

Cohanim, B. E., Dhillon, S., and Hoffman, J. A. (2012). "Statistical hazard detection using shadows for small robotic landers/hoppers". *2012 IEEE Aerospace Conference*.

Crane, E. S. and Rock, S. M. (2012). "Influence of trajectory on accuracy of hazard estimation during lunar landing". *AIAA Guidance, Navigation, and Control Conference*. No. 4554.

Crassidis, J. L. and Junkins, J. L. (2004). *Optimal estimation of dynamic systems*. Chapman and Hall/CRC.

Crisp, J. A., Adler, M., Matijevic, J. R., Squyres, S. W., Arvidson, R. E., and Kass, D. M. (2003). "Mars exploration rover mission". *Journal of Geophysical Research: Planets* Vol. 108, No. E12,

Cuseo, J. and Dallas, C. (1988). "Planetary terminal descent simmulation with Autonomous Hazard Avoidance". *27th Aerospace Scienece Meeting*, p. 515.

de Lafontaine, J., Neveu, D., and Hamel, J. F. (2008). "Autonomous planetary landing using a LIDAR sensor: the landing dynamic test facility". *GNC 2008:7th International ESA Conference Guidance, Navigation & Control Systems*.

De Rosa, D., Fisackerly, R., Pradier, A., Philippe, C., B., H., Carpenter, J., and Gardini, B. (2011). "ESA Lunar Lander Mission". *GNC 2011, 8th International ESA Conference on Guidance, Navigation & Control Systems*.

De Rosa, D., Bussey, B., Cahill, J. T., Lutz, T., Crawford, I. A., Hackwill, T., Gasselt, S. van, Neukum, G., Witte, L., McGovern, A., et al. (2012). "Characterisation of potential landing sites for the European Space Agency's Lunar Lander project". *Planetary and Space Science* Vol. 74, No. 1, pp. 224–246.

Dekker, R. J. (2003). "Texture analysis and classification of ERS SAR images for map updating of urban areas in the Netherlands". *IEEE Transactions on Geoscience and Remote Sensing* Vol. 41, No. 9, pp. 1950–1958.

Delaune, J., Le Besnerais, G., Sanfourche, M., Plyer, A., Farges, J.-L., Bourdarias, C., Voirin, T., and Piquereau, A. (2011). "Camera-aided inertial navigation for pinpoint planetary landing on rugged terrains". *International Planetary Probe Workshop*.

Delaune, J., Le Besnerais, G., Voirin, T., Farges, J.-L., and Bourdarias, C. (2016). "Visual–inertial navigation for pinpoint planetary landing using scale-based landmark matching". *Robotics and Autonomous Systems* Vol. 78, pp. 63–82.

Devouassoux, Y., Reynaud, S., Jonniaux, G., Ribeiro, R. A., and Pais, T. C. (2008). "Hazard avoidance developments for planetary exploration". *GNC 2008: 7th International ESA Conference on Guidance, Navigation & Control Systems*.

Epp, C. D., Robertson, E. A., and III, J. M. C. (2014). "Real-Time Hazard Detection and Avoidance Demonstration for a Planetary Lander". *AIAA SPACE Conference 4-7 August, San Diego, CA*. No. 4312.

Epp, C. D. and Smith, T. B. (2007). "Autonomous precision landing and hazard detection and avoidance technology (ALHAT)". *IEEE Aerospace Conference*.

Ezell, E. C. and Ezell, L. N. (1984). "On Mars: Exploration of the red planet, 1958-1978". *NASA Special Publication 4212*.

Geller, D. K. and Christensen, D. (2009). "Linear covariance analysis for powered lunar descent and landing". *Journal of Spacecraft and Rockets* Vol. 46, No. 6, pp. 1231–1248.

Gerth, I. and Mooij, E. (2014). "Guidance for Autonomous Precision Landing on Atmosphereless Bodies". *AIAA Guidance, Navigation, and Control Conference*. No. 0088.

Golombek, M., Grant, J., Kipp, D., Vasavada, A., Kirk, R., Fergason, R., Bellutta, P., Calef, F., Larsen, K., Katayama, Y., et al. (2012). "Selection of the Mars Science Laboratory landing site". *Space science reviews* Vol. 170, No. 1-4, pp. 641–737.

Graf, J. E., Zurek, R. W., Eisen, H. J., Jai, B., Johnston, M., and Depaula, R. (2005). "The Mars reconnaissance orbiter mission". *Acta Astronautica* Vol. 57, No. 2-8, pp. 566–578.

Gross, R. (2001). *Faster, Better, Cheaper: Policy, Strategic Planning, And Human Resource Alignment*. Tech. rep. Report Number IG-01-009, NASA Office Of The Inspector General.

Grotzinger, J. P., Crisp, J., Vasavada, A. R., Anderson, R. C., Baker, C. J., Barry, R., Blake, D. F., Conrad, P., Edgett, K. S., Ferdowski, B., et al. (2012). "Mars Science Laboratory mission and science investigation". *Space science reviews* Vol. 170, No. 1-4, pp. 5–56.

Group, V. E. A. (2014). *Venus Technology Plan*. NASA.

Halbrook, T. D., Chapel, J. D., and Witte, J. J. (2001). "Derivation of hazard sensing and avoidance maneuver requirements for planetary landers". *24th, Annual Rocky Mountain guidance and control conference 2001*, pp. 347–364.

Hamel, J.-F., Neveu, D., and Lafontaine, J. de (2006). "Feature matching navigation techniques for lidar-based planetary exploration". *AIAA Guidance, Navigation, and Control Conference and Exhibit*.

Hamel, J., Neveu, D., Mercier, G., Nagaty, A., Minville, M., Diedrich, T., Hornbostel, K., Hernandez, M., Sanz, K., De Rosa, D., Lefort, X., and Mongrard, O. (2018). "Validation of the PILOT Hazard Detection and Avoidance Function for Moon Exploration". *69th International Astronautical Congress (IAC), Bremen, Germany, 1-5 October 2018*. No. IAC-18-A3.2A.3.

Haralick, R. M., Shanmugam, K., and Dinstein, I. H. (1973). "Textural features for image classification". *IEEE Transactions on Systems, Man and Cybernetics* No. 6, pp. 610–621.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Hirschmuller, H. (2005). "Accurate and efficient stereo processing by semi-global matching and mutual information". *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE, pp. 807–814.

Holmberg, N. A., Faust, R. P., and Holt, H. M. (1980). "Viking'75 spacecraft design and test summary. Volume 1: Lander design".

Horn, B. K. (1977). "Understanding image intensities". *Artificial intelligence* Vol. 8, No. 2, pp. 201–231.

Howard, A. M., Jones, B. M., and Serrano, N. (2011). "Integrated sensing for entry, descent, and landing of a robotic spacecraft". *IEEE Transactions on Aerospace and Electronic Systems* Vol. 47, No. 1, pp. 295–304.

Huertas, A., Cheng, Y., and Madison, R. (2006). "Passive imaging based multi-cue hazard detection for spacecraft safe landing". *IEEE Aerospace Conference*.

Huertas, A., Cheng, Y., and Matthies, L. H. (2007). "Real-time Hazard Detection for Landers". *Proceedings NASA Science Technology Conference, Adelphi, M.D.*

Huertas, A., Cheng, Y., and Matthies, L. H. (2008). "Automatic hazard detection for landers". *9th International Symposium on Artificial Intelligence, R & A in Space, Los Angeles, California*.

Huertas, A., Johnson, A. E., Werner, R. A., and Maddock, R. A. (2010). "Performance evaluation of hazard detection and avoidance algorithms for safe Lunar landings". *IEEE Aerospace Conference*. No. 1662.

Ip, W.-H., Yan, J., Li, C.-L., and Ouyang, Z.-Y. (2014). "Preface: The Chang'e-3 lander and rover mission to the Moon". *Research in Astronomy and Astrophysics* Vol. 14, No. 12, p. 1511.

Jiang, X., Li, S., and Tao, T. (2016). "Innovative hazard detection and avoidance strategy for autonomous safe planetary landing". *Acta Astronautica* Vol. 126, pp. 66–76.

Johnson, A. E., Klumpp, A. R., Collier, J. B., and Wolf, A. A. (2002). "Lidar-based hazard avoidance for safe landing on Mars". *Journal of guidance, control, and dynamics* Vol. 25, No. 6, pp. 1091–1099.

Johnson, A. E. and Montgomery, J. F. (2008). "Overview of terrain relative navigation approaches for precise lunar landing". *2008 IEEE Aerospace Conference*. No. 1657. IEEE.

Kerr, M., Hagenfeldt, M., Ospina, J. A., Penin, L. F., Mammarella, M., and Bidaux, A. (2013). "ESA lunar lander: approach phase concept and G&C performance". *AIAA Guidance, Navigation, and Control Conference*. No. 5018.

Kirk, R., Howington-Kraus, E., Rosiek, M., Anderson, J., Archinal, B., Becker, K., Cook, D., Galuszka, D., Geissler, P., Hare, T., et al. (2008). "Ultrahigh resolution topographic mapping of Mars with MRO HiRISE stereo images: Meter-scale slopes of candidate Phoenix landing sites". *Journal of Geophysical Research: Planets (1991–2012)* Vol. 113,

Krüger, H. and Theil, S. (2010). "TRON hardware-in-the-loop test facility for lunar descent and landing optical navigation". *IFAC Proceedings Volumes* Vol. 43, No. 15, pp. 265–270.

Krüger, H., Theil, S., Sagliano, M., and Hartkopf, S. (2014). "On-Ground Testing Optical Navigation System for Exploration Missions". *9th International ESA Conference on Guidance, Navigation & Control Systems*.

Kytö, M., Nuutinen, M., and Oittinen, P. (2011). "Method for measuring stereo camera depth accuracy based on stereoscopic vision". *Three-Dimensional Imaging, Interaction, and Measurement*. No.7864. International Society for Optics and Photonics.

Lakdawalla, E. (2014). "China lands on the Moon". *Nature Geoscience* Vol. 7, No. 2, pp. 81–81.

Langley, C., Mukherji, R., Yang, G., Allen, A., and Tripp, J. (2007). "Full-scale Static Testing of the Lidar-based Autonomous Planetary Landing System (LAPS)". *Space Exploration Technologies* Vol. 6960,

Lingenauber, M., Bodenmüller, T., Bartelsen, J., Maass, B., Krüger, H., Paproth, C., KuSS, S., and Suppa, M. (2013). "Rapid modeling of high resolution moon-like terrain models for testing of optical localization methods". *ASTRA 2013 - 12th Symposium on advanced space technologies in robotics and automation*.

Liu, Y., Liu, B., Xu, B., Liu, Z., Di, K., and Zhou, J. (2014). "High Precision Topographic Mapping at Chang'E-3 Landing Site with Multi-Source Data". *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* Vol. 1, pp. 157–161.

Maass, B., Krüger, H., and Theil, S. (2011). "An edge-free, scale-, pose-and illumination-invariant approach to crater detection for spacecraft navigation". *7th International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE, pp. 603–608.

Madyastha, V. K., Ravindra, V. C., Mallikarjunan, S., and Goyal, A. (2011). "Extended Kalman filter vs. error state Kalman filter for aircraft attitude estimation". *AIAA Guidance, Navigation and Control Conference*.

Maimone, M. W., Johnson, A. E., Cheng, Y., Willson, R., and Matthies, L. H. (2006). "Autonomous navigation results from the Mars Exploration Rover (MER) mission". *Experimental robotics IX*, pp. 3–13.

Markley, F. L. (2003). "Attitude error representations for Kalman filtering". *Journal of guidance, control, and dynamics* Vol. 26, No. 2, pp. 311–317.

McEwen, A. S., Eliason, E. M., Bergstrom, J. W., Bridges, N. T., Hansen, C. J., Delamere, W. A., Grant, J. A., Gulick, V. C., Herkenhoff, K. E., Keszthelyi, L., et al. (2007). "Mars reconnaissance orbiter's high resolution imaging science experiment (HiRISE)". *Journal of Geophysical Research: Planets (1991–2012)* Vol. 112, No. E5,

McGee, T. G., Adams, D., Hibbard, K., Turtle, E., Lorenz, R., Amzajerdian, F., and Langelaan, J. (2018). "Guidance, Navigation, and Control for Exploration of Titan with the Dragonfly Rotorcraft Lander". *AIAA Guidance, Navigation, and Control Conference*. No. 1330.

Mourikis, A. I., Trawny, N., Roumeliotis, S. I., Johnson, A. E., Ansar, A., and Matthies, L. (2009). "Vision-aided inertial navigation for spacecraft entry, descent, and landing". *IEEE Transactions on Robotics* Vol. 25, No. 2, pp. 264–280.

Musoff, H. and Zarchan, P. (2009). *Fundamentals of Kalman filtering: a practical approach*. American Institute of Aeronautics and Astronautics.

Neveu, D., Hamel, J., Alger, M., Lafontaine, J. de, Tripp, J., Hussein, M., Hill, B., Dietrich, P., and Kerr, A. (2011a). "Autonomous planetary landing using a LIDAR Sensor: the full scale flight test experiment". *Proceedings of the 8th International ESA Conference on Guidance and Navigation Control Systems, Carlsbad, Czech Republic*.

Neveu, D., Mercier, G., Hamel, J.-F., Simard Bilodeau, V., Woicke, S., Alger, M., and Beaudette, D. (2015). "Passive versus active hazard detection and avoidance systems". English. *CEAS Space Journal* Vol. 7, No. 2, pp. 159–185.

Neveu, D., Hamel, J.-F., Simard Bilodeau, V., Alger, M., and Lafontaine, J. de (2011b). "Guidance, Navigation and Control technologies and facilities to support future lunar exploration missions". *Canadian Aeronautics and Space Journal* Vol. 57, No. 1, pp. 40–52.

Parkes, S., Martin, I., Dunstan, M., and Matthews, D. (2004). "Planet surface simulation with PANGU". *Eighth International Conference on Space Operations*, pp. 389–399.

Parreira, B., Vasconcelos, J. F., Montaño, J., Ramón, J., and Peñin, L. F. (2013). "Hazard Detection and Avoidance in ESA Lunar Lander: Concept and Performance". *AIAA Guidance, Navigation and Control Conference*. No. 5020.

Patil, S., Nadar, J. S., Gada, J., Motghare, S., and Nair, S. S. (2013). "Comparison of various stereo vision cost aggregation methods". *International Journal of Engineering and Innovative Technology* Vol. 2, No. 8, pp. 222–226.

Pentland, A. P. (1990). "Linear shape from shading". *International Journal of Computer Vision* Vol. 4, No. 2, pp. 153–162.

Pien, H. (1991a). "Autonomous Hazard Detection and avoidance for Mars exploration". *8th Computing in Aerospace Conference*. No. 3731.

Pien, H. H. (1991b). "Achieving safe autonomous landings on Mars using vision-based approaches". *Cooperative Intelligent Robotics in Space II*, pp. 24–35.

Pollini, A. (2012). "Flash optical sensors for guidance, navigation and control systems". *Advances in the Astronautical Sciences* Vol. 144, pp. 503–517.

Robinson, M., Brylow, S., Tschimmel, M., Humm, D., Lawrence, S., Thomas, P., Denevi, B., Bowman-Cisneros, E., Zerr, J., Ravine, M., et al. (2010). "Lunar reconnaissance orbiter camera (LROC) instrument overview". *Space Science Reviews* Vol. 150, No. 1-4, pp. 81–124.

Rogata, P., Di Sotto, E., Câmara, F., Caramagno, A., Rebordão, J., Correia, B., Duarte, P., and Mancuso, S. (2007). "Design and performance assessment of hazard avoidance techniques for vision-based landing". *Acta Astronautica* Vol. 61, No. 1, pp. 63–77.

Roy, S. (1999). "Stereo without epipolar lines: A maximum-flow formulation". *International Journal of Computer Vision* Vol. 34, No. 2-3, pp. 147–161.

Shimizu, M. and Okutomi, M. (2005). "Sub-pixel estimation error cancellation on area-based matching". *International Journal of Computer Vision* Vol. 63, No. 3, pp. 207–224.

Simard Bilodeau, V., Neveu, D., Bruneau-Dbuc, S., Alger, M., Lafontaine, J. de, Clerc, S., and Drai, R. (2012). "Pinpoint lunar landing navigation using crater detection and matching: design and laboratory validation". *AIAA Guidance, Navigation, and Control Conference*.

Smith, D. E., Zuber, M. T., Neumann, G. A., and Lemoine, F. G. (1997). "Topography of the Moon from the Clementine lidar". *Journal of Geophysical Research: Planets (1991–2012)* Vol. 102, No. E1, pp. 1591–1611.

Smith, D. E., Zuber, M. T., Jackson, G. B., Cavanaugh, J. F., Neumann, G. A., Riris, H., Sun, X., Zellar, R. S., Coltharp, C., Connelly, J., et al. (2010). "The lunar orbiter laser altimeter investigation on the lunar reconnaissance orbiter mission". *Space Science Reviews* Vol. 150, No. 1-4, pp. 209–241.

Strandmoe, S., Jean-Marius, T., and Trinh, S. (1999). "Toward a vision based autonomous planetary lander". *Guidance, Navigation, and Control Conference and Exhibit*, 1511–1522.

Striepe, S. A., Epp, C. D., and Robertson, E. A. (2010). "Autonomous Precision Landing and Hazard Avoidance Technology (ALHAT) Project Status as of May 2010". *International Planetary Probe Workshop*, pp. 12–18.

Strobl, K. H., Sepp, W., Fuchs, S., Paredes, C., Smisek, M., and Arbter, K. "DLR CalDe and DLR CalLab". http://www.robotic.dlr.de/callab/.

Sun, Z., Jia, Y., and Zhang, H. (2013). "Technological advancements and promotion roles of ChangE-3 lunar probe mission". *Science China Technological Sciences* Vol. 56, No. 11, pp. 2702–2708.

Tchoryk Jr, P., Gleichman, K. W., Carmer, D. C., Morita, Y., Trichel, M., and Gilbert, R. K. (1991). "Passive and active sensors for autonomous space applications". *Orlando'91, Orlando, FL*, pp. 164–182.

Tombari, F., Mattoccia, S., Di Stefano, L., and Addimanda, E. (2008). "Classification and evaluation of cost aggregation methods for stereo correspondence". *IEEE Conference on Computer Vision and Pattern Recognition*.

Ulamec, S. and Biele, J. (2010). "From the Rosetta Lander Philae to an asteroid hopper: lander concepts for small bodies missions". *7th International Planetary Probe Workshop, Barcelona*.

Vaughan, R., Gaskell, R., Halamek, P., Klumpp, A., and Synnott, S. (1991). "Autonomous precision landing using terrain-following navigation". *Spaceflight Mechanics 1991*, pp. 57–75.

Way, D. W., Powell, R. W., Chen, A., Steltzner, A. D., Martin, A., Burkhart, P. D., and Mendeck, G. F. (2007). "Mars Science Laboratory: Entry, descent, and landing system performance". *IEEE Aerospace Conference*. No. 1467.

Woicke, S. and Mooij, E. (2014). "Stereo-Vision Algorithm for Hazard Detection during Planetary Landings". *AIAA Guidance, Navigation, and Control Conference*. No. 0272.

Woicke, S. and Mooij, E. (2015). "Comparative Analysis Of Different Passive Hazard Mapping Techniques". *International Plantary Probe Workshop*.

Woicke, S. and Mooij, E. (2016a). "A stereo-vision hazard-detection algorithm to increase planetary lander autonomy". *Acta Astronautica* Vol. 122, pp. 42–62.

Woicke, S. and Mooij, E. (2016b). "Passive hazard detection for planetary landing". *AIAA Guidance, Navigation, and Control Conference*. No. 1133.

Woicke, S. and Mooij, E. (2018). "Terrain Relative Navigation for Planetary Landing Using Stereo Vision Measurements Obtained from Hazard Mapping". *Advances in Aerospace Guidance, Navigation and Control*, pp. 731–751.

Xiong, W., Rong, S., and Genghua, H. (2013). "Effects of rocket engines on laser during lunar landing". *Physics Letters A* Vol. 337, (38), pp. 2598–2603.

Xiong, Y., Olson, C., and Matthies, L. (2001). "Computing depth maps from descent imagery". *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001*. Vol. 1,

Yan, R., Cao, Z., Zhu, L., and Fang, Z. (2013). "Hazard avoidance via descent images for safe landing". *Eighth International Symposium on Multispectral Image Processing and Pattern Recognition*. No. 89180.

Yoshikawa, M., Fujiwara, A., and Kawaguchi, J. (2006). "Hayabusa and its adventure around the tiny asteroid Itokawa". *Proceedings of the International Astronomical Union* Vol. 2, pp. 323–324.

Zhang, X., Drake, N. A., Wainwright, J., and Mulligan, M. (1999). "Comparison of slope estimates from low resolution DEMs: Scaling issues and a fractal method for their solution". *Earth Surface Processes and Landforms* Vol. 24, No. 9, pp. 763–779.

Zhu, S., Cui, P., and Hu, H. (2012). "Hazard detection and avoidance for planetary landing based on Lyapunov control method". *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, pp. 2822–2826.

ACKNOWLEDGEMENTS

As a daughter of an engineer, I basically always wanted to be an engineer (okay, apart from the brief moments where I wanted to be a vet or an archaeologist). When it was time to choose what to study after finishing high school, I was also considering becoming a mathematician, but decided that engineers had the cooler jobs in the end (so far I think I was right). The landing of Spirit and Opportunity on Mars, and the not-so successful landing of Beagle have fascinated me in my childhood and made me consider Aerospace Engineering as my *engineering profession of choice*. I have never regretted this choice ever since, and it is still the crazy and bold exploration missions that fascinate me most. The planets are still hiding many undiscovered secrets and I hope that I will mange to contribute revealing them. This work was my first, tiny, step towards this goal. I hope many others will follow.

This work was funded via the Beatrix de Rijk Scholarship from the Faculty of Aerospace Engineering of Delft University of Technology. I am very grateful that I have been selected as one of the recipients of this scholarship, based on a proposal that Erwin and me wrote together more than five years ago. Without this funding, my work would not have been possible. I am especially grateful that the scholarship gave me the opportunity to attend a lot of conferences and workshops during my time as a PhD candidate.

This work would have not been possible without the contribution of a lot of people. I met many people along the way who supported me, encouraged me or simply fascinated me. I would like to thank all of them. In the following I would like to express my thankfulness towards the most important ones.

First of all, I would like to thank my promoter Pieter Visser. Pieter always supported my work and encouraged me to go on and pursue my dreams. He always gave me the feeling of being a good PhD candidate and a valued team member in our department. I am also very grateful that he gave me the opportunity to represent our department in the faculty's PhD board, a task which I really enjoyed. In this context, I would also like to thank Boudewijn Ambrosius who served as my promoter until after my Go/No Go meeting – I have been the last master student that graduated under his lead, and I felt honoured to also be the last PhD candidate he took on, even though it has been for only a limited amount of time.

Erwin Mooij, my co-promoter, without you all of this would have never happened. Who knows what I would be doing now, but you were clearly the most important puzzle piece in my professional development so far. I could have not wished for a better supervisor. You have always supported and guided me in whatever I came up with (and at least in theory I now know how correct hyphenation works). As you may know in German the "promoter" is called a Doktorvater, I think this 100 % describes you.

I would also like to thank Hans Krüger and Bolko Maass from DLR Bremen. Both provided me with a lot of valuable feedback and suggestions during my visits to DLR. Especially, I am grateful to Hans for entrusting me his laboratory, TRON, for my hardware-

in-the-loop tests. Without this opportunity my work would have been a lot less founded. I would also like to thank the remaining staff at RY-GNC for making me feel welcome during my stays at DLR. My visit to DLR always helped me to regain momentum and, whenever necessary, motivation. I would like to also thank Marco and Guilherme for providing an open ear and good advice whenever necessary.

Jacco, Tim and Günther, I would have not survived this without you. When I started thinking about the contents of these acknowledgements, I was searching my memories for that one specific moment that I would mention here, representative of all those amazing moments we had. I could simply not single out one (or even two) that were the single "best" or "most representative". I will never forget a single of these great moments we had.

Also a big thank you to Relly for always having an open ear, sorting out our problems, but also arranging everything for my defence while I was already in Germany. I would especially like to thank you for your patience in helping me to improve my Dutch!

My time as a PhD would have never been the same without all the current and past PhDs, Granddad-Mao, Jinglang, Haiyang (I still owe you a dinner!), Hermes (I still miss teasing you. And I will NEVER forget what you said at Bart's wedding!!!), Gourav (talking to you always reminded me of my early PhD days, and that I am old now), Yuxin, Tatie, Bart, Dominic, Teresa, Bas, Sowmini, and Patrik who is not a PhD candidate but does belong in this list, but also the 8th floor PhDs! I enjoyed all our talks, lunches, fun events and dinners. Our multi-cultural group taught me a lot.

I would also like to thank all other staff members of our section, Jose, Wouter, Ron, Marc, Eelco, Daphne, Wim, Berhard, Leonid and Ejo for the great time I had during my time in Delft.

I would also like to mention my room mate Anja, who was doing her PhD at an entirely different faculty. It was always good to have someone with an open ear and an other view point. But even without that you were an important part of my PhD-time and I think you were the best room mate I could have possibly had.

I would like to thank my parents who have supported me in so many different way over all those years. I hope they are proud to see that I have reached this point. But also Ingo's family for their support during these four years of PhD work.

Last, but not least, I would like to thank Ingo for his endless support. He was there for me whenever I needed it. Also, he has contributed to this work not only by reviewing the final draft but also by endless discussion we led about the topic of this work and related research.

And finally to everyone who feels he or she should have been mentioned here, but was not: thank you!

Bremen, Germany, March 2019
Svenja Woicke

Curriculum Vitæ

Svenja Woicke

01-09-1989	Born in Minden, Germany.

Education

1999–2008	Grammar School Städtisches Gymnasium Peterhagen, Petershagen Germany
2008–2011	Bachelor of Science in Aerospace Engineering Delft University of Technology, Delft, The Netherlands
2011-2014	Master of Science in Space Exploration (cum laude) Delft University of Technology, Delft, The Netherlands *Thesis:* Stereo Vision for Hazard Detection Development, Verification and Testing *Supervisor:* Dr. ir. E. Mooij

Awards

2014	2nd price best student paper, IPPW 2014
2015	3rd price best student paper, IPPW 2015
2016	2nd price best student poster, IPPW 2016
2017	2nd price best student paper, IPPW 2017
2018	2nd price best poster, PhD Posterday, Faculty of Aerospace Engineering, TU Delft

Next to her research activities, Svenja also contributed to the education at TU Delft. She supervised multiple MSc students during their thesis work, all in the field of GNC-systems. She also supervised BSc groups of ten students working on a mission design as the final project of their study. In the context of the space minor, she supervised a total of

ten students on assignments based on crater detection algorithms. One of these groups was able to present its work during two conferences.

She contributed content during the course Re-entry Systems, given by Dr. ir. E.Mooij in the form of a lecture on advanced landing systems, but also by supporting exam development and grading.

During her time at Delft University of Technology Svenja has been involved in multiple extracurricular activities. She served as the chairperson of the first PhD board of the Aerospace Engineering faculty from 2014 till 2017. Here she acted as a representative of the PhDs towards the graduate school actively involved in supporting PhDs at the faculty, but also arranged scientific (Symposium and Pitching competition) and social events for the faculty's PhDs. Next to this, Svenja is currently (since 2016) the chair person of the student organising committee and a member of the organising committee of the International Planetary Probe Workshop. In 2017 the workshop was held in The Hague, where Svenja actively supported the local organising committee.

List of Publications

Peer-reviewed Journal and Conference Articles

1. **S. Woicke and E. Mooij**, *Stereo-Vision Algorithm for Hazard Detection during Planetary Landings*, AIAA Guidance, Navigation, and Control Conference, Paper number: 0272 (2014).

2. **D. Neveu , G. Mercier, J.-F. Hamel, V. Simard Bilodeau, S. Woicke**, *Passive versus Active Hazard Detection & Avoidance Systems*, CEAS Space Journal, Vol. 7, Issue 2, pp. 159 to 185 (2015).

3. **S. Woicke and E. Mooij**, *Passive Hazard Detection for Planetary Landing*, AIAA Guidance, Navigation, and Control Conference, Paper number: 1133 (2016).

4. **S. Woicke and E. Mooij**, *A stereo-vision hazard-detection algorithm to increase planetary lander autonomy*, Acta Astronautica, Vol. 122, pp. 42 to 62 (2016).

5. **S. Woicke and E. Mooij**, *Terrain Relative Navigation for Planetary Landing using Stereo Vision Measurements Obtained from Hazard Mapping*, Advances in Aerospace Guidance, Navigation and Control, Vol. 4, pp. 731 to 751 (2017).

6. **S. Woicke, A. Moreno Gonzalez, I. El-Hajj, J. Mes, M. Henkel, R. Klavers, R. Autar** *Comparison of Crater-Detection Algorithms for Terrain-Relative Navigation*, AIAA Guidance, Navigation, and Control Conference, Paper Number: 1601 (2018).

7. **S. Woicke, H. Krüger and E. Mooij**, *Hardware-in-the-Loop Testing of Stereo Vision-Based Hazard-Detection Method for Planetary Landing*, AIAA Guidance, Navigation, and Control Conference, Paper Number: 1602 (2018).

Conference Contributions

1. **S. Woicke and E. Mooij**, *A Stereo-Vision Based Hazard-Detection Algorithm for Future Planetary Landers*, International Planetary Probe Workshop (2014) [presentation].

2. **S. Woicke and E. Mooij**, *Comparative Analysis Of Different Passive Hazard Mapping Techniques*, International Planetary Probe Workshop (2015) [presentation].

3. **S. Woicke and E. Mooij**, *Combining Camera-based Hazard Detection and Terrain Relative Navigation in a SLAM-like approach*, International Planetary Probe Workshop (2016) [poster].

4. **S. Woicke and E. Mooij**, *Combined Algorithm for Final Descent Hazard-Relative Navigation and Hazard Detection*, International Planetary Probe Workshop (2017) [presentation].

5. **S. Woicke and E. Mooij**, *A stereo-vision based hazard-relative navigation-algorithm for planetary landings*, ESA GNC Conference (2017) [poster].

6. **S. Woicke, H. Krüger, and E. Mooij**, *Hazard relative navigation for precision planetary landings*, International Astronautical Congress, Bremen (2018) [presentation and paper].