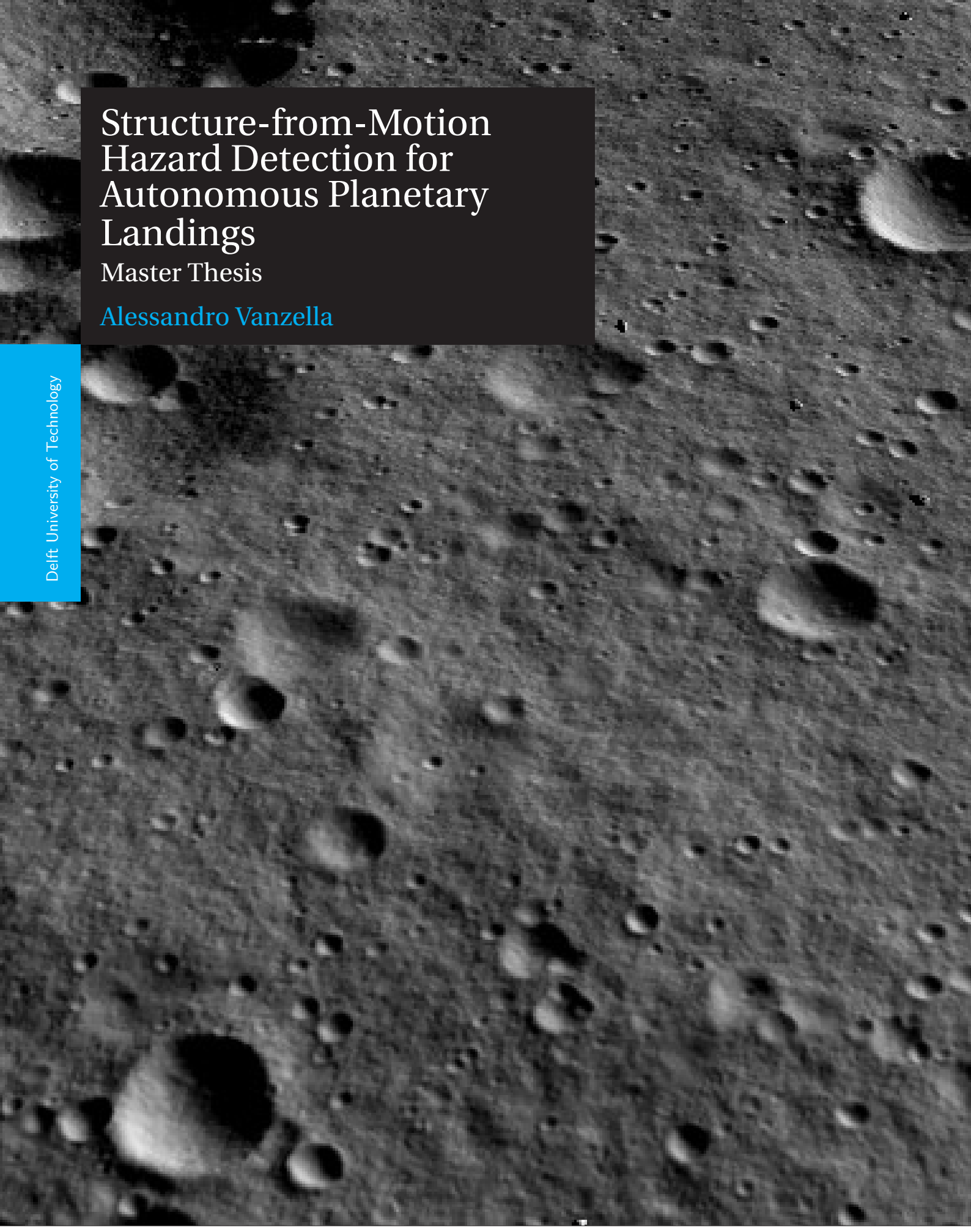

Structure-from-Motion Hazard Detection for Autonomous Planetary Landings

Master Thesis

Alessandro Vanzella

Delft University of Technology

TU Delft Delft University of Technology

Challenge the future

Front-page figure Synthetic image of the lunar surface rendered in the Planet and Asteroid Natural scene Generation Utility (PANGU), altitude 300 m

Structure-from-Motion Hazard Detection for Autonomous Planetary Landing

by

Alessandro Vanzella

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology

to be defended publicly on Monday November 13, 2017 at time 9:30 AM.

Student number:	4417690	
Project duration:	December 12, 2017 – September 20, 2017	
Supervisors:	Ir. S. Woicke,	TU Delft
	Dr. Ir. E. Mooij,	TU Delft
Thesis Committee:	Ir. Woicke,	TU Delft
	Dr. Ir. E. Mooij,	TU Delft
	Prof. Dr. ir. P. N. A. M. Visser,	TU Delft
	Dr. Ir. G. de Croon,	TU Delft

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

Technische Universiteit Delft

To my parents, Piermilo and Antonietta
and my sister Marica

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Ir. Svenja Woicke for her constant technical supervision and for giving me the opportunity to develop such interesting research. With equal merit, I would like to thank Dr. Ir. Erwin Mooij. Without him, I would have probably never discovered the beauty of planetary entry, descent, and landing. I will be certainly missing his enthusiastic lectures on re-entry systems. I also want to thank Ir. Ron Noomen who guided me to perform my internship at the Colorado Center for Astrodynamics Research (CCAR). The period spent working in Boulder will always remain in my heart. A special thank goes to all my friends back home and to all those with which I have shared this long journey. Fortunately, too many to be cited in a single page. To my girlfriend Madie, for being next to me throughout all these months and to make my life colourful. My greatest gratitude to my family. Looking backwards, how I got where I am, this would have not been possible without your unconditional support and efforts. Thanks for your endless love

Summary

Future space exploration missions on solar system bodies will require landing safely and precisely, with an accuracy of ~100 m at touchdown. This accomplishment is made challenging by vehicle design limitations, the dearth of onboard situational awareness, and the limited knowledge of the variability of the landing terrain. To date, only the Chinese Chang'e-3 has implemented hazard detection and avoidance capabilities, within its Guidance, Navigation, and Control (GN&C) subsystem, therefore being able to actively adjust its trajectory. On the contrary, the majority of the space landers only had the ability to execute autonomously a small series of simple and programmed commands. Therefore, past missions have essentially landed "blind" in regions deemed relatively safe, forcing landing site selection to be capability-limited rather than scientifically driven.

In this thesis, hazard detection was investigated as a mean to increase autonomy for planetary landings and to further decrease the risk of a landing failure, employing equipment readily available on space missions. The analysis has been limited to the framework of Structure-from-Motion (SfM) where the input images are acquired from a single moving camera and thus the scene is reconstructed from the resulting video sequence. A software package was developed and tested to compute depth maps from adjacent descent images, captured at half altitude from one to the other. The basic pinhole camera model was selected to address the measurement taken from synthetic surface images, rendered in the Planet and Asteroid Natural scene Generation Utility (PANGU). To assess the hazardousness of the terrain, hazard maps are computed combining slope, roughness, and shadow information. In contrast to the results of the Jet Propulsion Laboratory (JPL) NASA, it has been shown that rocks and boulders are not well resolved from shape recovery with both low- and high-elevation image pairs. Thus, their presence on the surface has been accounted through an adapted version of the Harris Corner detector directly on the input images. Two different mission scenarios were simulated: 1) a perfect vertical motion forward along the camera pointing direction and 2) a 45° angle dropping trajectory for a more realistic approaching descent phase, with a 40° imaging sensor line-of-site offset. Furthermore, the limitations of the developed algorithm were tested under ordinary operative conditions.

For the former scenario, the results show that the overall quality of the recovered depth maps does not appear adequate enough for landing site selection. As a matter of fact, the locations around the image centre can not be correctly assessed. This represents a significant problem since these locations are the most convenient in terms of distance and guidance costs. On the contrary, the latter descent sequence indicates that below 300 m altitude the software is a suitable candidate for hazard detection, with total correct detection on average >94% and the percentage of undetected hazards below the allowable maximum 1%.

To assess the algorithm robustness to errors in camera position, a Monte Carlo simulation was performed. Thereupon, random uncertainties within the interval [-0.5 0.5] meters were taken into account for the altitude of both camera poses. The errors for the computed Digital Elevation Model (DEM) are bounded to the maximum allowable only when both altitudes are affected by small deviation of similar magnitude and same sign (approximately ±10 cm), peaking to 250%-300% increase for the other values of the considered interval. Moreover, concerning the robustness to errors in camera orientation, deviations of the camera pointing direction were considered only along the plane containing both the normal to the surface and the camera axis. Already differences greater than +0.05°, in the imaging sensor line-of-site, are responsible for exorbitant errors in the DEM for all altitudes. These results clearly indicate that the developed SfM algorithm is not suitable as a stand-alone method for hazard detection and landing site selection.

Contents

List of Acronyms

SfM Structure-from-Motion

PANGU Planet and Asteroid Natural scene Generation Utility

JPL Jet Propulsion Laboratory

NASA National Aeronautics and Space Administration

DEM Digital Elevation Model

CCAR Colorado Center for Astrodynamics Research

GPS Global Positioning System

EDL Entry, Descent and Landing

IMU Inertial Measurement Unit

MSL Mars Science Laboratory

HDA Hazard Detection and Avoidance

HD Hazard Detection

LS Landing Site

LIDAR LIght Detection And Ranging

MSSS Malin Space Science Systems

ASC Advanced Scientific Concept

RQ Research Question

TAN Terrain Absolute Navigation

TRN Terrain Relative Navigation

VN Visual Navigation

EDM Entry, Descent and Landing Demonstrator Module

OSIRIS-REx Origins, Spectral Interpretation, Resource Identification, Security, Regolith Explorer

JAXA Japan Aerospace Exploration Agency

PID Proportional-integration-differentiation

S/C Spacecraft

ME Main Engine

FOV Field-of-view

ESA European Space Agency

LEDA Lunar European Demonstration Approach

HASE Hazard Avoidance System Experiment

HM Hazard Mapping

PDI Power Descent Initiated

AG Approach Gate

NMP New Millenial Program

ST9 Space Technology 9

TRGS Terrain Relative Guidance System

ALHAT Autonomous Landing and Hazard Avoidance Technology

NRC National Research Council

TRL Technology Readiness Level

HV Host Vehicle

HDS Hazard Detection System

GSD Ground Sample Distance

HRN Hazard Relative Navigation

ILP Nominal Landing Point

SfS Shape-from-shading

HSE Homography-based Slope Estimation

SLAM Simultaneous Localization and Mapping

TP True Positive

TN True Negative

FP False Positive

FN False Negative

SB Sub-goal

MATLAB MATrix LABoratory

OpenCV Open Source Computer Vision Library

SSD Sum of Squared Differences

SAD Sum of Absolute Differences

NaN Not-A-Number

MAD Median Absolute Deviation

RMS Root-Mean-Square

CDI Correct Detection Index

SVD Singular Value Decomposition

List of Symbols

u	pixel x-position, px
v	pixel y-position, px
f	focal length, px
$\boldsymbol{C}$	vector camera centre, m
I_1	image at camera 1st pose
I_2	image at camera 2nd pose
I_2^k	warped version camera 2nd pose
img	input image
θ	line-of-sensor imaging offset, rad
X	x-coordinate, m
Y	Y-coordinate, m
Z	distance camera terrain (camera frame), m
$\hat{\boldsymbol{Z}}$	unit vector Z-axis
h	altitude camera (world frame), m
$\boldsymbol{X}$	vector arbitrary terrain point, m
$\boldsymbol{x}$	vector arbitrary image point, px
$\boldsymbol{K}$	intrinsic camera matrix, px
$\boldsymbol{R}$	orientation camera matrix, rad
$\boldsymbol{T}$	vector location camera, m
$\boldsymbol{I}$	identity matrix, -
$\boldsymbol{M}$	camera projection matrix
$\boldsymbol{N}$	unit vector normal to the slicing planes
$\boldsymbol{D}$	vector distance travelled by the lander
d	distance travelled along optical axis
$\boldsymbol{H}$	homography matrix
p_x	x-coordinate principal point, px
p_y	y-coordinate principal point, px
$\boldsymbol{v}$	vector coordinates principal point, px
Δx	pixel disparity, px
C	correlation cost
μ	mean value
MAD	median absolute deviation
slp	slope subscript
rgh	roughness subscript
$\hat{\boldsymbol{n}}$	surface normal to plane
$\boldsymbol{A}$	matrix of x,y-coordinates for least-squares fit
p, q, r	plane coefficients, -
S	slope, rad
R	roughness, m
N	number of pixels in window
S	slope, rad
z	elevation, m
z_{mean}	mean elevation, m

Chapter 1

Introduction

Landing safely and precisely on the Moon, Mars, or any Solar System body has been identified as a crucial capability for future robotic and manned space exploration missions [17]. This is clearly not a trivial task and its achievement is made challenging by different aspects. First of all, the distance involved and thus the time delay in communication, between the spacecraft and the operators on Earth, make impossible to actively control the trajectory of the robotic lander during its final descent. Furthermore, the dearth of navigation aids (e.g. GPS or radio-beacons) forces Entry, Descent, and Landing (EDL) systems to rely only on Inertial Measurement Units (IMU) for position and attitude determination [29].

Given an estimate of the initial position and velocity, IMUs simply integrate acceleration measurements thus being in theory fully self-sufficient, with no knowledge about the surroundings necessary [14]. Nevertheless, these sensors are subject to unbounded error accumulation due to the associated integration of noise and errors (e.g. bias, axis misalignment, angle random walk to name but a few). This, when combined with uncertainties in the position and attitude at entry, and in the decelerator models, results in large landing error ellipses (200x100 km for Viking landers, 1976 [14]).

Despite guided entry has considerably shrunk them (20x7 km for Mars Science Laboratory MSL, 2012 [14]), we are still far from the capability of performing pin-point landings. [14] Moreover, to date, only the NASA MSL and the Chinese Chang'e-3 were able to actively adjust their trajectory during the EDL phase. All previous space landers only had the ability to execute autonomously a small series of simple and programmed commands, forcing to choose landing locations based on detailed analysis of coarse-resolution orbital imagery: in plain words, past missions have essentially landed "blind" in regions deemed inherently safe (hazard-free to the larger extent), in compliance with the lander design tolerances.

Still, the next robotic planetary surface missions are becoming much more ambitious in their scientific goals [25], as they attempt to answer questions related to the opportunity of life in our Solar System and how to access and utilize resources from other celestial bodies. Areas near craters, ridges, fissures, and other relevant geological formations are likely to be the most interesting landing sites. Hence, landing next to a surface asset and on bodies where the surface is not sufficiently mapped are key capabilities for future exploration.

Hazard Detection and Avoidance (HDA) might represent the answer to reach landing sites with high safety. As its name suggests, HDA allows to detect surface hazards by means of sensors, protecting the vehicle from landing on top or right next to them. Sensing the surface and extracting local illumination, slope, and roughness information, the safety of the candidate landing site (LS) can be assessed. Only if deemed necessary, a new safe LS can be selected and communicated to the Guidance, Navigation, and Control (GN&C) subsystem. Therefore, it is clear that HDA can be employed only if the vehicle can actively adjust its trajectory.

Limiting the analysis to the framework of Hazard Detection (HD), therefore only to how potential hazards can be identified, several architectures are currently in development. These technologies are proposed in literature as methods employing active sensors, such as radar [3] and Light Detection and Ranging (LIDAR) instruments [31], or passive sensors, like cameras [13]. The latter are optical devices usually working in the visible spectrum: since the output is a simple image, they can only deliver an estimate of the elevation at each pixel through complex algorithms. The former, instead, emit a signal of a certain wavelength and then they detect the returning echo: illuminating the features of interest they can directly measure ranges of the sensed terrain, therefore allowing to obtain the Digital Elevation Map (DEM) directly from the measurements. On the other hand, passive sensors have the advantage of being light weight, cheap, and to consume only a little power. For instance, the modular space camera *ECAM-C30* [1] manufactured by Malin Space Science Systems (MSSS) has dimensions of 7.8(W)×5.8(L)×4.4(H) cm, mass

[1] `http://www.msss.com/space-cameras`

of 0.256kg, and 2.5W for power consumption. The *GoldenEye 3D* [2] by Advanced Scientific Concepts (ASC) represents instead the state of the art for flash LIDAR space cameras and is currently onboard OSIRIS-REx sample return mission. With dimensions 14(W)×21.6(L)×14.5(H) cm, mass of 4kg, and 35W for power consumption the difference stands out. Saving mass and volume, HD systems based on passive sensors comply with the present needs for miniaturization and furthermore, they allow increased redundancy: indeed cameras represent a feasible option as a back-up for active systems, keeping the added weight within limits. [39]

The foregoing discussion has highlighted how landing a spacecraft requires autonomy, and how equipment readily available on space missions can potentially help in reaching landing site with high safety. Therefore, the following high-level Research Question (RQ) is set to be answered

> RQ1 *Is it possible to increase autonomy for planetary landings through equipment readily available on space missions?*

Considering passive sensors, it was previously mentioned that they can only recover depth information through complex algorithms. Being a hot topic in computer vision over the past few decades, different methods have been developed for shape recovery: these are often classified into different categories according to the information they employed (e.g structure from motion, shape from silhouettes, shape from shading). The analysis has been limited to the framework of structure-from-motion (SfM) where the input image pair is acquired from a single moving camera and thus the scene is reconstructed from the resulting video sequence. This imaging technique has already been successfully employed in many fields [36], including robotics, surveillance, and virtual reality. The framework of planetary EDL sets challenges that traditional SfM algorithms cannot overcome: large difference in resolution, lander motion along the camera optical axis, to name but a few. Therefore, the main research goal of the project is *to numerically investigate the limitations of a SfM algorithm for planetary landing applications*. To move a step forward, a lower-level RQ is set as follows

> RQ1.1 *Is structure-from-motion suitable for HD as a stand-alone method?*

When analyzing the flowchart of a possible GN&C system architecture, which employs HDA, this includes additional blocks to perform terrain absolute (TAN) and relative navigation (TRN) (Figure 1.1). The outcome of these additional blocks is combined with the traditional IMU and altimeter measurements to decrease the lander's localization error. Nevertheless, no previous research has answered *quantitatively* whether traditional sensors provide knowledge of the lander's location accurate enough for hazard detection with SfM.

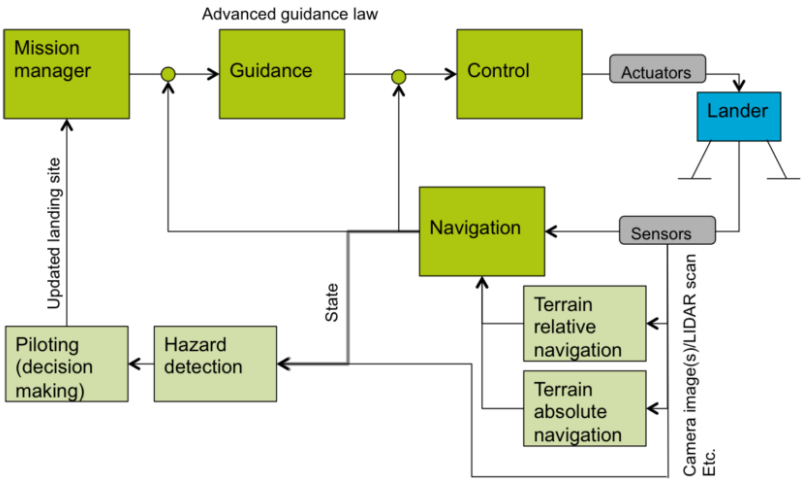

Figure 1.1: Flowchart of a GN&C system architecture [14]

[2] `http://www.advancedscientificconcepts.com/products/portable.html`

Therefore a further research question has been identified and is shown below

> RQ1.2 *Do traditional sensors (e.g. IMU and altimeter) provide knowledge of the lander's location accurate enough for SfM Hazard Detection or Visual based Navigation (VN) has to be added to GN&C architecture?*

To understand the context of this study within the framework of actual space missions, Chapter 2 discusses past, present, and future missions relevant to our research. Chapter 3 extensively describes the modelling of a single moving camera and how to generate depth maps from a sequence of descent images. Chapter 4 illustrates the software architecture and the choices made during its whole development process. Software verification is extensively tackled in Chapter 5. The results obtained with the simulation software are given in Chapter 6, together with a sensitivity analysis highlighting the overall software performances under ordinary operative conditions. Finally, in Chapter 7 the conclusions of our research are presented while Chapter 8 discusses some recommendations for future work.

Part I
Heritage

Chapter 2

HDA Studies

Technologies for planetary landings have been studied and developed since the late fifties during the Moon race [8], which culminated on July 21st 1969 with the first human landing on the Earth's natural satellite. Despite the failure in the manned exploration of the Moon, the Soviets engaged in a highly fruitful exploration of Venus during the same period. Indeed in December 1970, Venera 7 became the first spacecraft to successfully soft land on another planet and the first to transmit data back from there to the Earth.

Nowadays, small probes and landers are sent to far distant planetary bodies such as Mars (e.g. Schiapparelli EDM lander and NASA MSL), asteroids (e.g. NASA OSIRIS-REx on 101955 Bennu asteroid and JAXA Hayabusa on 25143 Itokawa), and comets (e.g. Rosetta's lander Philae on comet 67P/Churyumov-Gerasimenko). Looking backwards, on how we got where we are, it is clear that the space agencies all over the world still cannot master the task of planetary descent and landing: even if we increased our expertise on landing on Mars we still have no idea how to land, for instance, on Europa, one of the Jupiters moons. Besides, the more ambitious the scientific and engineering goals get the less likely it will be that we can design missions solely based on heritage systems [14].

On one hand, it is impressive to acknowledge that to date only a single flying mission has employed HDA during the descent and landing phase, namely the Chinese Chang'e-3 [35]. On the other hand, there is the awareness that much progress needs to be made and therefore the thrill of just beginning. Despite this study specifically investigates SfM systems, this does not imply that only such are important as a reference: as a matter of fact, different methods and also HDA architectures based on active sensors may contribute with useful data and information. In the following, Sections 2.1 to 2.7 provide a short survey on heritage missions and projects related to hazard detection, through which mission and system requirements have been derived and presented in Section 2.8. Finally, Section 2.9 presents the thesis project goals to better structure the research effort.

2.1 Chang'e-3

On December 2013, Chang'e-3 successfully achieved China's first soft-landing on the Moon. Being the first robotic spacecraft adopting autonomous HDA [35], the technology that has been developed represents a critical foundation for any HD study. To ensure a safe landing, HD based on both passive and active sensors was considered. For coarse avoidance, Chang'e-3 adopted a grey-image-based HD and a fourth order polynomial guidance [42]. The algorithm involves the following five steps [35]:

- image histogram analysis
- k-means clustering-based image thresholding
- obstacle recognition according to brighter and darker textures
- obstacle distribution and edge detection
- edge closing of the hazardous area

Therefore, the algorithm relies on local intensity clustering to segment the image into regions and then it classifies them as hazardous or not. To achieve near real-time performances on a flight hardware (less than one second per image) only shadows and a single class of hazards are segmented [26]. Considering an image of 400×400 pixels, local windows varying from 8 to 40 pixels are usually chosen with an overlap of half of their area size. Thus, if a hazard falls precisely on the border of one local area then it will fall in

the center of the next window considered. Local segmentation provides several advantages, for instance, it allows the algorithm to be robust to a gradually changing surface: as a matter of fact, the difference between the mean values of the outliers and the surface tends to be higher using a smaller area. The algorithm selects the locally optimal k mean values, such that the distance of each data point to its nearest mean value is minimized using the following equation:

$$E = \sum_i \min_j |p_i - u_j| \tag{2.1}$$

where p_i is the *i-th* pixel belonging to a specific local window, u_j is the *j-th* mean intensity for a class j=1,...,k, and E represents the error value to minimize with clusters.
Minimization is achieved by looping the following three procedures until E is smaller than a certain tolerance

Assign clusters

$$l_{p_i} = \arg\min_j |p_i - u_j| \tag{2.2}$$

where l_{p_i} is the cluster to which pixel p_i has been associated

Calculate k means

$$u_l = \frac{\sum_i^{N_l} p_i^l}{N_l} \tag{2.3}$$

where u_l is the new mean value for cluster l (with l=1,...,k) and N_l represents its number of samples

Reiterate

After the image has been segmented, the separated regions are classified as either shadows, hazards, or part of the attainable region. Chang'e-3 adopted a spiral search strategy to find the nearest safe LS [42]. As the name suggests, the search is performed spirally outward starting from the pre-selected LS. Figure 2.1 illustrates a sketch for the spiral search strategy together with an example of a computed target area shown in both a hazard map and the input image.

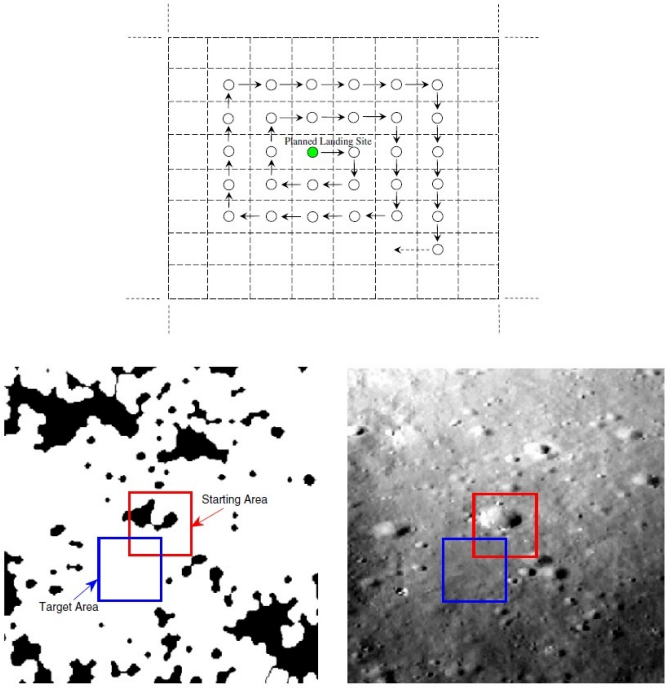

Figure 2.1: Schematic of a spiral search strategy (top) and results on a hazard distribution map (bottom-right) and on the input grey-scale image (bottom-left). [35]

For precise avoidance, a 3D laser radar topographic map with the well-established proportional-integration-differentiation (PID) guidance were selected. The HD process is limited to a 50×50 m^2 terrain area, centered on the currently sub-satellite point [35] and a least-squares fitting is employed to compute the reference level plane. Once the hazard cost map of the landing area has been computed, the spiral search method is again applied for the location assessment of a safe LS. The presented approach enabled Chang'e-3 to identify and avoid rocks larger than 0.2 m and slopes steeper than 8 deg [35].

As a valuable reference, Chang'e-3 mission outline is explained in detail. Captured by the Moon on 6 December at 17:53 UTC+8, Chang'e-3 first achieved a 100 km altitude circumlunar orbit. Four days later, a de-orbit braking maneuver successfully targeted the spacecraft (S/C) into a 100×15 km elliptical orbit. The preparation for the powered descent and soft-landing operations (Figure 2.2) occurred about 10 minutes before, with the upload of the ignition time (14 December 20:59:52 UTC+8), the initial (position 19.0464°W, 28.9989°N, altitude 14.884 km, velocity 1695.7 m/s) and terminal landing states (position 19.509°W, 42.12°N, altitude 2 m, velocity 0 m/s) to the lander.

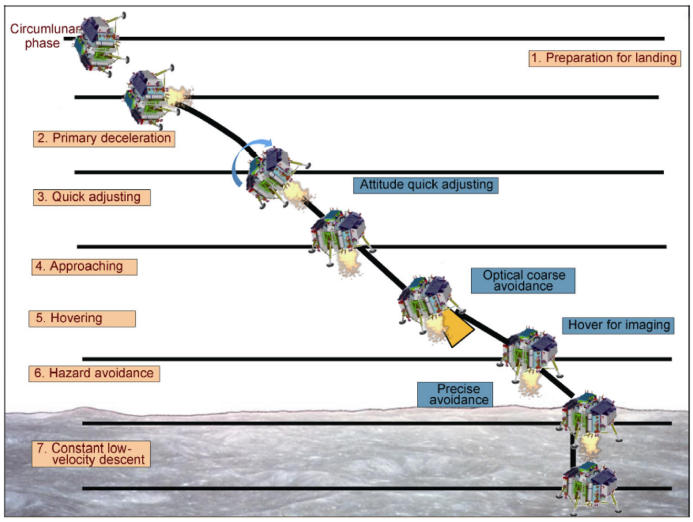

Figure 2.2: Illustration of Chang'e-3 powered descent and landing phases [35]

The ignition of the 7500 N main engine (ME) marked the beginning of the *primary deceleration*: Chang'e-3 velocity was reduced from ~1.7 km/s to ~70 m/s and its altitude from ~15 km to ~3 km above the surface. Concurrently, the pitch angle of the lander dropped from ~85° to ~65° (respect to the vertical to the surface) due to the joint effect of thrust and lunar gravity.

Microwave ranging sensor measurements were introduced at an altitude of ~8 km to correct bias and drift of the IMU [35]. About 487 s after the beginning of the powered descent the GN&C system automatically switched to the *quick adjusting mode*, within which the pitch angle was adjusted to ~9°, while roll and yaw angle were kept 0°.

At 503 s the *approaching descent mode* began, with the main objective to perform coarse HDA, while the lander was descending from ~2.4 km to ~100 m above the lunar surface. According to the requirements of coarse avoidance, the line-of-sight of the imaging sensor was intentionally offset 40° to ensure the imaging sensor 30° field of view (FOV) was directed towards the landing area. Taking optimal fuel consumption into account, the onboard GN&C autonomously selected a new LS approximately 80 m far from the originally updated LS. A 45° angle dropping trajectory was adopted to approach the reselected site.

Reached the 100 m altitude, the lander started to hover above the surface while maintaining the attitude. Subsequently, at 644 s the system switched to the *hazard-avoidance mode*, attaining an altitude of ~30 m. Performing fine HDA the LS was moved approximately 9 m further to the northeast [35].

Thereupon, the GN&C transitioned to the *constant low-velocity descent mode*: the horizontal velocity was reduced to zero while the attitude of the lander was progressively aligned with the normal direction of the surface. In this phase only the IMU was adopted to provide navigation information, because the thrusters effect on the lunar surface dust would have severely affected laser and microwave ranging sensors reliability. The lander descended at a constant velocity of 2 m/s until it reached 2.88 m above the surface, where a gamma sensor gave a shutdown signal, turning off all thrusters. The lander freely fell and touched down the surface 687 s after the beginning of the powered descent.

2.2 ESA Lunar Lander

The ESA Lunar Lander is a technological demonstration mission for autonomous soft, safe precision landing on the Moon. Planned to be launched no later than 2018, the project dates back to the mid '90s where specific concerns in lunar landings led to several technological activities. The Lunar European Demonstration Approach (LEDA) [1] assessed an exploration mission whose main objective was to soft-land a spacecraft at the lunar south pole, using an autonomous onboard vision-based GN&C system, including HD capability and landing site re-targeting. Besides, within the Aurora Core programme, ESA initiated the Hazard Avoidance System Experiment (HASE) study to further mature the vision-based HDA solution proposed for the Lunar Lander mission [7]. Notwithstanding all these efforts, at ESA's Ministerial Council in November 2012, no further funding was allotted for the project.

The mission is yet a valuable case study for the framework of HDA. To introduce functional redundancy for the slope and roughness computation, an architecture employing both a camera and LIDAR was considered. The shadow, slope, and roughness maps computed by the hazard mapping (HM) function are converted into scoring maps, allocating scores to each pixel in compliance to the hazard values. Thereupon, the three maps are merged into a single hazard map which reflects the overall hazard level of the surface terrain.

Alongside, a piloting function determines the attainable region for re-targeting. Taking as input the navigation data, and the terrain DEM directly measured by the LIDAR sensor, the ground coordinates of the LS are evaluated. A related Distance cost map is therefore obtained, assessing the distance of every candidate LS respect to the nominal LS, together with the associated cost: sites that are further away from the nominal are penalized.

The associated guidance costs (function of the amount of fuel required to reach the candidate LS) are taken into account in the guidance cost map. The piloting function indeed computes the trajectory profile that the lander will follow to reach each candidate site, checking for feasibility, available fuel, and thrust authority constraints. For a candidate LS, if the trajectory is not feasible, the guidance will select a maximum cost value, hence indicating the LS is not reachable. The attainable re-targeting map is determined merging the Distance Cost Map and the Guidance Cost Map. [7]

An additional piloting function is responsible for the re-targeting decision. The hazard map (output of the HM function) is interpreted and a risk score for each candidate LS is computed, thus attaining a risk map. A global map, which merges the attainable re-targeting and the previous risk map, is then realized: the aforementioned reflects both the reachability and safety constraints for the LS, enabling the best candidate to be identified. It should be pointed out that re-targeting only occurs if the current LS is found to be unsafe and if the new candidate is meaningfully better. A consistency check is performed to verify that the same LS is chosen consistently for few iterations. Only thereupon re-targeting is commanded.

2.2.1 Mission Outline

Through a de-orbit burn over Moon's North Pole, the lander starts out its descent trajectory. This is followed by a coast phase where Terrain Absolute Navigation (TAN) helps to establish an accurate determination of its position. Close to the South Pole at an altitude of ~ 15km the Powered Descent Initiated (PDI) point occurs, which marks the beginning of the main braking phase. Firing its all 500N ME the vehicle velocity is reduced from ~ 1500m/s to ~ 100m/s, besides additional 220N pulsed assist engines (AE) are ignited allowing thrust modulation and increased controllability. The *GN&C* guides and control the vehicle to reach the Approach Gate (AG) at ~ 2.5km where the approach phase starts. During this phase, LIDAR data and camera images are collected to select autonomously a safe landing site.

Two diverts are allowed, and they are designed to occur within specific altitude zone: the first occurs nominally between altitude ~1.5km and ~1.3km in a zone termed far range, while the second divert occurs in the close range nominally between altitude ~300m and ~150m. With such strategy landing site terrain is observe with progressively increased accuracy, leading precise touchdown capabilities of below 10m accuracy and horizontal divert capabilities of up to ~300m. Lastly, the lander executes a controlled vertical descent and soft touchdown on the lunar surface.

2.3 Space Technology 9 Terrain-Relative Guidance System (TRGS)

Further studies have performed HDA research independently from a specific mission. For instance, the NASA's New Millennium Program (NMP) inaugurated the Space Technology 9 (ST9) Terrain-Relative Guidance System (TRGS), the latest of a series of technology validation activities begun in 1996 with Deep Space 1. NASA selected TRGS from five candidate technology capabilities that had been under preliminary consideration [19]. Within the project, three vision algorithms for HD were developed:

- Stereo-based slope estimation
- Stereo-based rock detection
- Shadow-based rock detection

Slope estimation from stereo range data is achieved through a robust plane fitting [5]. A first least-median square fit is repeated for multiple triplets of points. The triplet for which the median of the squared plane error is minimum is kept, therefore allowing to eliminate outliers. A result of this is shown in Figure 2.3

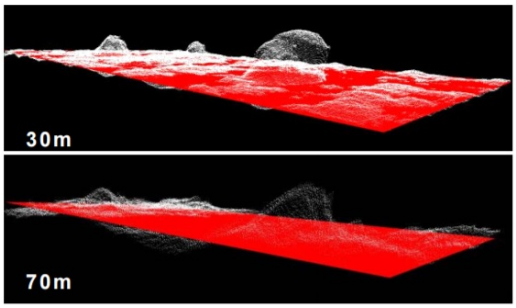

Figure 2.3: Robust plane fitting applied to 3D stereo range data for two different altitudes [5]

Stereo-based rock detection is achieved by computing the deviation from the mean plane and averaging the 25 highest range points in each region. [5] The latter step enables to reduce noise in the estimates. To detect rock hazards at much higher altitudes than stereo-based estimation, the shadows they cast has been employed to detect their presence. This is achieved by fitting the "best-ellipse" in shadow regions larger than five 5 pixels 2.4

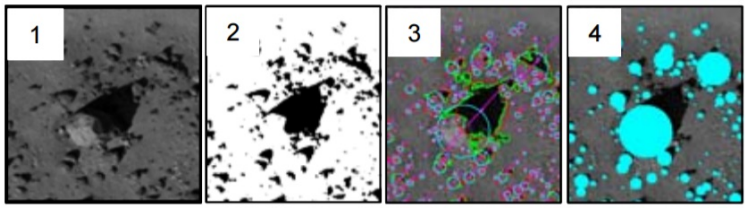

Figure 2.4: Shadow-based rock detection [5]

2.4 ALHAT

The development of technology that enables safe precision landings has been identified as a priority for future EDL missions also by the National Research Council (NRC) [2]. With the Autonomous precision Landing and Hazard Avoidance Technology (ALHAT) project, NASA has addressed this priority aiming to make LS selection scientifically driven rather than capability limited.

Tasked in 2006 [40], the project has developed and matured to a Technology Readiness Level (TRL) 6 [12] an autonomous system combining GN&C with terrain sensing and recognition functions for crewed, cargo, and unmanned planetary landing vehicles. No Host Vehicle (HV) description has been made in the definition of the HD system (HDS), therefore allowing a large-independent design and development path.

ALHAT design was driven by real-time landing operations requirements, primarily on HV navigation performance, HD pointing accuracy, and software communication interface. The process of safe site determination is demanded to occur within a 10 s interval [21], starting from a 1 km slant range along a 30° approach angle trajectory. During this time interval, the HD system successfully executes the following 4 tasks

- Terrain imaging of a 1 hectare region
- DEM construction
- HD processing
- Safe sites identification.

Aiming to be employed by vehicles with different hazard tolerances, ALHAT team established that the system should detect LS that are free of 30 cm roughness hazards and with surface slopes not to exceed 5° [21]. This roughness tolerance requires DEM generation with a 10cm Ground Sample Distance (GSD), using range images with a 5 cm (1σ) range precision. To generate such large DEMs under any lighting conditions, a flash-LIDAR technology for real-time terrain imagery was selected. The sensor [22], based on 3D imaging camera technology developed by ASC, employs a 128×128 px detector array camera allowing a 900 m operational range at 30° approach angle (Figure 2.5). The receiver Field-Of-View (FOV) was chosen to be 1.0° to provide the necessary 10 cm GSD accuracy. [21] The small FOV requires the overlapping of multiple flash-LIDAR images of the surface terrain for generation of a 1000×1000 element DEM (1 hectare at 10cm GSD). Therefore a twin-axis gimbal slews the flash-LIDAR to acquire the mosaic of images that are processed to generate the desired DEM.

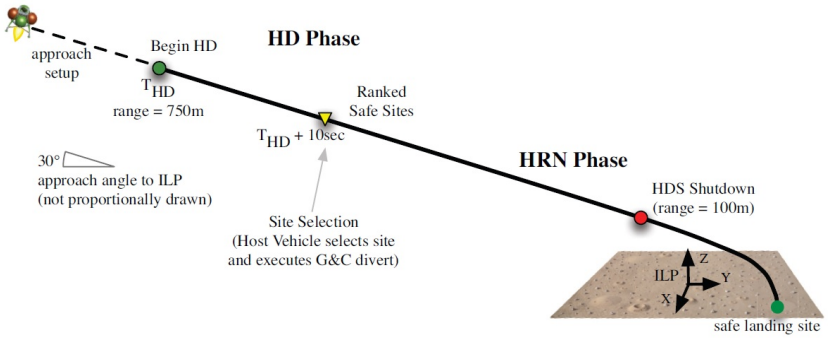

Figure 2.5: ALHAT HD trajectory profile for safe and precision landing [21]

2.4.1 Concept of Operations

Figure 2.5 also highlights that two operational phases occur during the HD trajectory profile: 1) the HD phase and 2) the Hazard Relative Navigation (HRN) phase. During the former, the HDS generates onboard a DEM of the surface terrain and processes it to assess safe LS within a 1 hectare region centered on the nominal landing site (ILP). The latter generates additional descent images and correlates them with the previously computed DEM, in order to match terrain features and to calculate lateral position offsets for LS-relative state updates.

To construct the DEM, the HDS a collection of range images is seamlessly assembled. As previously state, the entire 1 hectare region cannot be imaged to 10cm GSD from a single 1° FOV image. Therefore, the HDS plans a mosaic path (Figure 2.6) along the surface that is tracked by the gimbaled sensor during image collection. The slew maneuver takes approximately half of the 10 s required for the whole HD phase [21]. The HRN phase starts automatically upon HD completion, nevertheless it is not discussed since it is not meaningful for this project.

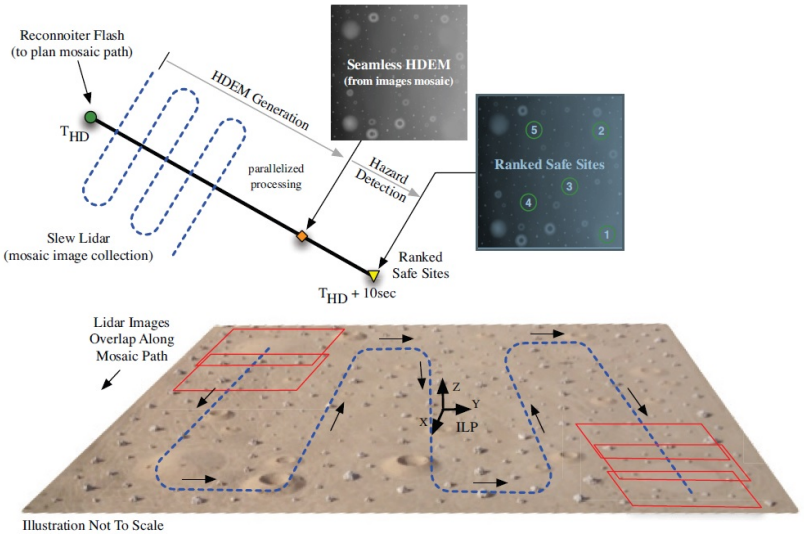

Figure 2.6: Detailed operations of ALHAT HD phase [21]

2.5 NASA Mars Exploration Program Advanced Technologies

As its name suggests, the NASA Mars Exploration Program Advanced Technologies was started to solicit technology developments needed for future martian missions, to be launched during or after the 2009 launch window. "Pin-point landing" (within tens of meters to 1 km of a target site), and advanced methods for detecting, avoiding, or tolerating landing hazards were identified as crucial requirements for future exploration missions [9].

Within this project Huertas et al. [4] developed a passive "multi-cue" HD system. Rather than computing a true hazard map, their proposed method selectively applies different algorithms to eliminate hazard areas and to identify only promising areas for further processing. The HD system involves four different phases. During the *survey stage*, occurring at early descent, only large craters are quickly located: comparison of their intensity gradient against an ideal crater model allows to calibrate shape-from-shading (SfS) and intensity to angle functions. The latter is employed at the last part of the parachute descent, namely *regional hazard detection stage*, to identify pixels on steep slopes.

Furthermore, through a Homography-based Slope Estimation (HSE) the flatness of the potential LS is assessed. As output, the best landing region (typically larger than the LS) is selected and the spacecraft is steered there. During the last part of the descent the *local hazard detection stage* starts, allowing the detection of small craters and rocks. The present mode employs texture analysis, SfS maps, and again the HSE due to the availability of higher resolution images. Finally, in the *site selection stage* the algorithm choose the "safest" site among all the potential candidates.

2.6 Depth Maps from Descent Images

Xiong et al. [43] at JPL have developed a method which computes depth maps, useful for navigation, from adjacent descent images taken at half altitude from one to the other. Their proposed approach has never flown on-board and experiments have validated it. A perfect motion forward along the camera pointing direction (Figure 2.7) is here considered.

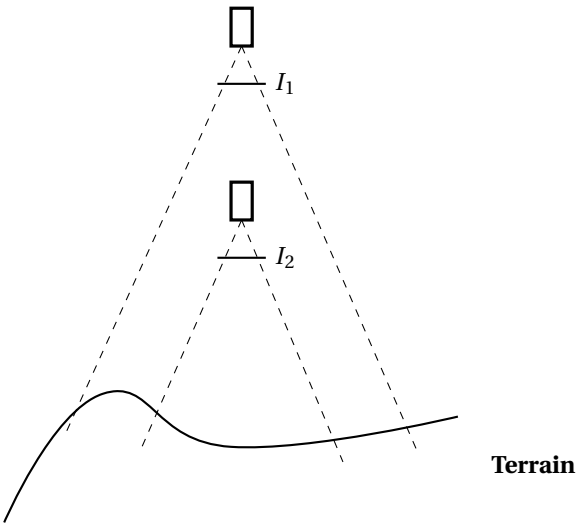

Figure 2.7: Descent motion along the camera principal axis

Under such assumption, dense depth recovery is achieved through a set of virtual parallel planes [43]. These slice the terrain inducing a homography $\boldsymbol{H_i}$ such that the image I_2, taken at camera 2nd pose, can be warped back to the first image plane. Therefore, the two images are made resolution equivalent and aligned. To better understand what this means, let us consider Figure 2.8. Given a point A, which lies on both the terrain and plane π_i, let a_2 be its image belonging to I_2. Therefore, $a_1 = \boldsymbol{H_i} a_2$ represents the projection of A in image I_1. For the virtual parallel planes π_{i-1} and π_{i+1} the induced homography matrices are $\boldsymbol{H_{i-1}}$ and $\boldsymbol{H_{i+1}}$, respectively. Both $\boldsymbol{H_{i-1}} a_2$ and $\boldsymbol{H_{i+1}} a_2$ are image points which belongs to I_1, nevertheless they are projection of the terrain points P and Q. Therefore, only when the 3-D point lies on both the terrain and the virtual plane its projections satisfy the homography induced by the considered virtual plane [11]. Since the lighting conditions vary little in the two images, the corresponding points are highly correlated. Therefore, the estimated depth value at each pixel is the depth of the plane for which the highest correlation is achieved.

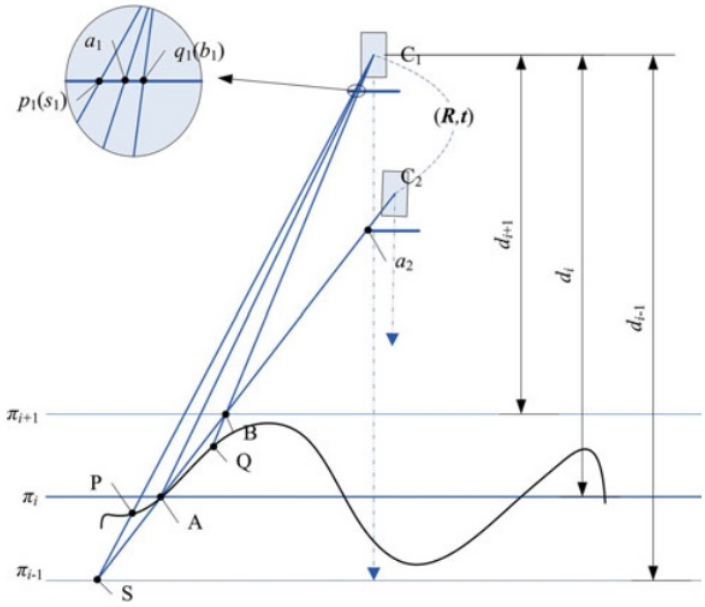

Figure 2.8: Scheme of the homography-based depth recovery [11]

2.7 Research at TU Delft

The Astrodynamics and Space Missions group at the Delft University of Technology is highly involved in planetary re-entry and descent studies, and in guidance and control system design. For instance, Ir. Svenja Woicke is currently investigating the use of simultaneous localization and mapping (SLAM) methods for HDA and terrain relative navigation (TRN). Her research also lead to the development of a stereo-vision based hazard detection algorithm for future planetary landers [37].

Her work [38] [39] [37] is not only a valuable reference for the algorithm system requirements, but also for the optimal methods to compute hazards within the framework of planetary landings. As a matter of fact, considering the slope computation, she has shown that methods such *stepest descent*, *third order finite difference*, and *partial quadratic equation* produce very large errors, overestimating the slope by more than 200% [38] in presence of noise and boulders. In [38] she concluded that despite *intelligent mean plane* algorithms better estimate slope, the gain obtained does not justify an execution time increased by at least a factor 4 respect to *linear regression*. Furthermore, in [39] she also provides an accurate comparison of 3 different shape recovery methods (stereo-vision, structure-from-motion, and shape-from-shading) within the framework of HD.

2.8 System Requirements

The previous sections have established the context of this research within the framework of actual space missions. This choice allows us to preset System Level 0 requirements (Table 2.1) for the hazard detection software. Most of them can be explicitly derived from the reference missions/projects formerly tackled. Despite Chang'e-3 considered 20 cm roughness features as hazardous [35], it has been decided to be bound to the less stringent requirement (30 cm roughness) demanded by ALHAT [21]. As matter of fact, in the attempt to answer more ambitious scientific goals, the next generation of landers is expected to increase their hazard tolerances. In accordance with recent studies [38], when considering the hazard posed by the terrain slope, it has been agreed that a 15° slope is expected to cause a hazard for future landers, while ALHAT and Chang'e-3 identify and avoid LS characterized by 5° and 8° slope respectively.

The requirement concerning the computational time of the HD algorithm often results "hard-nosed". Clearly, the HD system has to be computationally light enough to operate in real-time on space qualified hardware. In literature [21] it can be found that the sophisticated ALHAT demands approximately 4 seconds to get DEM readings, while other methods [18] have been tested to be much faster. Naively, we may ask what is the "conversion factor" to come up with a number for the computational requirement on our personal computer. Let us cut to the chase: it does not exist. Some tests have compared current personal computer with space qualified hardware, nevertheless the numbers found are only valid for those specific machines considered. Based on further studies [37] and common sense, it can be considered that if an

HD evaluation takes up to 2 seconds in an ordinary personal computers, this will be fast enough when operating on-board an older and slower [27] space qualified hardware. The requirements on the landing sites are related to the performance assessment. Comparing the LS found in the computed hazard map to the sites belonging to the ground-truth, the LS can in general be classified as

- *True Positive* (TP), if correctly identified as safe
- *True Negative* (TN), when it is correctly identified as unsafe.
- *False Positive* (FP), if an unsafe site is erroneously considered safe
- *False Negative* (FN), when an actually safe site is erroneously considered unsafe

For a robust and reliable HD algorithm, a strong correct assessment of the landing sites is obviously desired with undetected hazards (FP) consequently bounded to a small percentage (≤ 1%) [37].

Table 2.1: Hazard Detection functional requirements

Autonomy	fully autonomous
Hazards	
Shadows	
Roughness	≥ 30cm
Slope	≥ 15°
Landing Site (LS)	
Safe LS classification (TP+TN)	≥ 99%
Unsafe LS labelled as safe (FP)	≤ 1%
Real-time performances	
Computational time (on personal computer)	≤ 2s

2.8.1 Mission Scenarios

Similarly to the research paper [43], a scenario assuming a perfect vertical motion forward along the camera pointing direction is initially considered. With the camera height decreasing from 1000 m to 50 m above the surface, the altitude range is not only representative of the HDA assessments, but also it allows the investigation of the algorithm operational envelope.

Only the very last phase of a planetary descent is characterized by a vertical motion, therefore an additional scenario has also been constructed from literature to better simulate the lander's approaching phase. Based on Chang'e-3 powered flight (Section 2.1), a 45° angle dropping trajectory has been considered with the input images captured starting from 2400 m above the ground. The lander's pitch angle is initially set to 9°, while both the yaw and roll angle are kept at 0°. The sensor line-of-site is intentionally offset to 40° to ensure the camera is free of the effect of main engine plume and and its 30° FOV is directed towards the landing area. For both scenarios, a perfect knowledge of the lander's position and orientation has been initially assumed.

Given that the system employed is simply a camera, in accordance with other studies [30] [37] no reference vehicle has been described. A pinhole camera model (see Section 3.2 for description), which assumes a linear relation between image point position and the direction of the associated camera ray, and 8-bit gay scale input images have been considered. Color, indeed, does not add greater insights since most of the information contained in an image comes from the intensity maps. Furthermore, in gray-color the red, green, and blue components have all equal intensity in RGB space: therefore, it is only necessary to specify a single intensity value for each pixel. This can be stored in only 8-bit, giving 256 shades of gray. Concerning the size of the input images, 512×512 pixels size have been found a good compromise between the level of the terrain surface details and the demanded computational time for processing it. [37]

Table 2.2: Parameters of the camera system

Camera Model	pinhole
FOV	30°
Image resolution	512×512 px
Color	8-bit gay scale

2.9 Research Objectives

It is clear that planetary EDL sets challenges [43] [11] that traditional SfM approaches can not overcome. For instance, images taken at different instant of time during the lander descent are affected by large resolution differences, therefore making correlation more difficult. Furthermore, it is likely that the epipoles (points of intersection of the line joining the camera centres with the image planes) (Appendix A.3) are located inside the image boundaries, causing the failure of conventional rectification approaches: indeed, no linear relation that can map the epipole to infinity exists.

Nevertheless, Section 2.6 has illustrated a method that can robustly compute depth maps from descent images, for navigational purposes. Therefore, this approach is employed as a starting point for the development of a SfM based hazard detection algorithm. Already in the Introduction (Chapter 1), the main research goal of the project was stated as *numerically investigate the limitations of a SfM algorithm for planetary landings, given traditional sensor (e.g. IMU and altimeter) measurements to assess the lander's location.* Since a SfM approach has been now selected, to better structure the research effort the following sub-goals (SB) have been identified:

SB1: *Solutions to tackle depth recovery in the vicinity of the epipole*

SB2: *Algorithm robustness to errors in both camera position and orientation*

SB3: *Assess whether dense reconstruction, near the image center, is possible with adjacent image pairs such that one is taken at less than half altitude (e.g. one third) of the other image*

It is obvious that only after the software has been designed and verified, it is possible to proceed with the implementation of the required simulations to assess all the SBs.

Part II
Methods and Software

Chapter 3

Modelling

As an introduction to structure-from-motion, the issue of generating terrain maps from a sequence of descent images is tackled as follows: first, we present the reference frames and camera model selected to address the measurements taken from images in Section 3.1 and 3.2, respectively. The current state-of-the-art to compute depth from descent imagery is extensively discussed in Section 3.3. Finally, a description of the virtual descent images that will be employed in this research is given in Section 3.4.

3.1 Reference Frames

Any description and access to image locations requires the definition of a reference frame. Since digital gray-scale images are stored as two-dimensional arrays, the most convenient method for expressing locations is to employ pixel indices: therefore images are treated as a grid of discrete elements. Similarly to *MATLAB* and *OpenCV* [23], the reference system for the *image plane* is centred on the top left corner of the image, with pixels ordered from left to right and top to bottom. For 3D scene reconstruction, the *camera frame* has been set accordingly as illustrated in Figure 3.1, with the camera principal axis pointing along the Z_C axis.

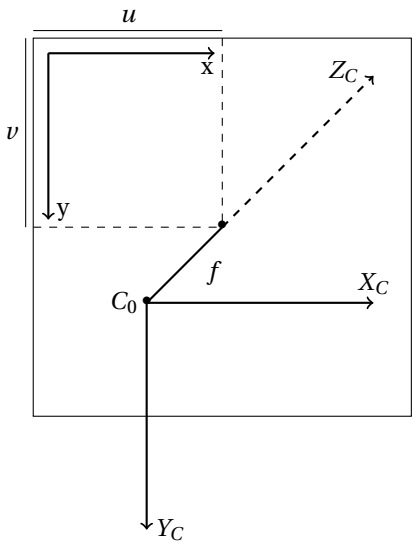

Figure 3.1: Camera (X_C, Y_C, Z_C) and image plane (x, y) reference frames

3.2 Camera Model

To address measurements taken from images it is necessary to select a model which coherently describes the camera. First and foremost, we define *camera* a mapping between the 3D world (object space) and a 2D image. [32] The model here considered is the basic *pinhole camera* which can be simply described thanks to projective geometry and expressed through a matrix representation. Under this model

a point **X**, belonging to the object space, is mapped to the point where the line joining **X** and the camera centre **C** meets the image plane (Figure 3.2).

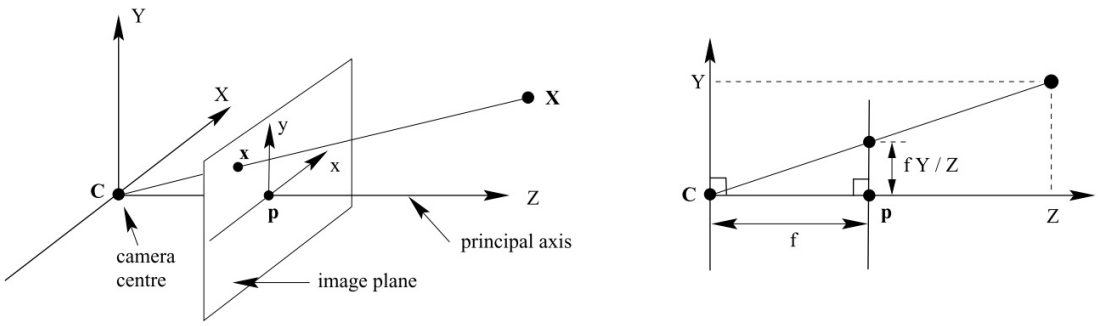

Figure 3.2: *Pinhole Camera Model Geometry*. Note the image plane is placed in front of the camera centre **C** [32]

Therefore, there exists a linear mapping between the position of the image point and the direction of the associated camera ray. Considering the camera frame, previously defined in Section 3.1, a 3D point is mapped on the image plane as:

$$(X,Y,Z)^T \rightarrow \left(f\frac{X}{Z}+u, f\frac{Y}{Z}+v\right)^T \tag{3.1}$$

with f the focal length of the camera, while u and v the coordinates of the principal point (namely **p** in Figure 3.2) relative to the image plane origin. Exploiting the matrix representation the following can be written

$$\begin{pmatrix} fX+u \\ fY+v \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & u \\ 0 & f & v \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \boldsymbol{KX} \tag{3.2}$$

where **K** is called *intrinsic camera matrix* (or *calibration matrix*) and transforms 3D camera coordinates to 2D homogeneous image coordinates. As the dominating imaging model in computer vision [16] besides, in agreement with other studies employing cameras for space application (e.g. optical navigation [23]), the pinhole camera model has been chosen.

3.3 Depth Recovery

A qualitative description on how depth can be recovered from two adjacent descent images was previously outlined in Section 2.6. To move a step forward, the shape recovery steps are here presented in detail. In the following, $\boldsymbol{C_1}$ and $\boldsymbol{M_1}$ represent the camera centre and camera projection matrix respectively. Instead, $\vec{C}_2$ and $\boldsymbol{M_2}$ are the same parameters at camera 2nd pose. For any pixel of this latter image, its world location has to lie on the 3D ray defined by

$$\boldsymbol{X} = s\boldsymbol{M_2^{-1}} \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} + \boldsymbol{C_2} \tag{3.3}$$

where r_2, c_2 are the row and column location of the pixel, while s is a positive scale factor. The k-th virtual planar surface π_k is given by

$$\boldsymbol{N^T X} + z_k = 0 \tag{3.4}$$

where $\boldsymbol{N} = (0,0,-1)^T$ is the normal of π_k, and z_k represents the distance between $\boldsymbol{C_1}$ and plane π_k. The minus sign in $\boldsymbol{N}$ is because the normal of the virtual parallel planes is pointing in opposite direction of the camera frame Z_C axis. Substituting Equation 3.3 into the previous constraint, the scale factor s can be retrieved as follows:

$$s\boldsymbol{N^T M_2^{-1}} \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} + \boldsymbol{N^T C_2} + z_k = 0$$

$$s = -\frac{\boldsymbol{N^T C_2} + z_k}{\boldsymbol{N^T M_2^{-1}} \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}} \tag{3.5}$$

The back projected 3D ray (Equation 3.3) can now be determined. Through the camera projection matrix $\boldsymbol{M_1}$, the 3D world point $\vec{X}$ of the object space is mapped to the point onto the first image, taken at a higher altitude. Referring to the centre of projection $\boldsymbol{C_1}$, it is possible to write

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}) = \boldsymbol{M_1}(\boldsymbol{X} - \boldsymbol{C_1}) \tag{3.6}$$

Between the two images there exist a projective homography $\boldsymbol{H_k}$, in other words, a 3×3 matrix that linearly relates points of the image pairs.

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \boldsymbol{H_k} \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} \tag{3.7}$$

where the matrix specifying the projective homography (see Appendix B for derivation) is attained from

$$\boldsymbol{H_k} = \boldsymbol{M_1}(\boldsymbol{C_2}) - \boldsymbol{C_1})\,\boldsymbol{N^T M_2}^{-1} - \left(\boldsymbol{N^T C_2} - z_k\right)\boldsymbol{M_1 M_2^{-1}} \tag{3.8}$$

Assuming the camera frame at 1st pose for the 3D reconstruction $\boldsymbol{C_1} = 0$ and $\boldsymbol{C_2} = -\boldsymbol{R^{-1}T}$, where $\boldsymbol{R}$ and $\boldsymbol{T}$ describe respectively the relative orientation and location of the camera at 2nd pose. Similarly, $\boldsymbol{M_1} = \boldsymbol{K}$ and $\boldsymbol{M_2} = \boldsymbol{KR}$. Under the assumptions of a pure translation motion along the camera's principal axis $\boldsymbol{R} = \boldsymbol{I_{3\times3}}$, the identity matrix, thus $\boldsymbol{M_1} = \boldsymbol{M_2} = \boldsymbol{K}$. With $\boldsymbol{H_k}$, I_2 is warped back to the first image plane. Furthermore, through this approach images are made resolution equivalent and aligned, therefore ready for correlation. The requirement for the input images, to be taken at half altitude from one to the other, arises to achieve dense reconstruction maps near the image center [11]. Let t_0 be the time at which the first image is captured moreover, let assume the camera frame at this instant of time to coincide with the world frame. Therefore, an arbitrary terrain point $\boldsymbol{X} = (X, Y, Z)$ is imaged into

$$\boldsymbol{x_{t_0}} = \frac{1}{Z}\boldsymbol{KX} \tag{3.9}$$

where **K** is the camera calibration matrix and the imaged point is characterized by an homogeneous representation. Similarly, when the second image is captured at t_1

$$\boldsymbol{x}_{t_1} = \frac{1}{Z-d}\boldsymbol{K}(\boldsymbol{X} - \boldsymbol{D}) \tag{3.10}$$

assuming the lander has travelled the distance $\boldsymbol{D} = (0,0,d)^T$, along the optical axis.
By denoting with $\boldsymbol{v} = \left(p_x, p_y, 1\right)^T$ the coordinates of the principal point in terms of pixel dimensions in the $\boldsymbol{K}$ matrix, it is possible to write

$$Z\left(\boldsymbol{x_{t_1}} - \boldsymbol{x_{t_0}}\right) = \frac{1}{Z-d}\left(d\boldsymbol{KX} - Z\boldsymbol{KD}\right) = \frac{1}{Z-d}\left(d\boldsymbol{KX} - Zd\boldsymbol{v}\right)$$

Adding and subtracting $\frac{d}{Z-d}\,\boldsymbol{KD}$ yields

$$\begin{aligned} Z\left(\boldsymbol{x_{t_1}} - \boldsymbol{x_{t_0}}\right) &= d\left[\frac{1}{Z-d}\left(\boldsymbol{KX} - \boldsymbol{KD}\right) + \frac{1}{Z-d}\left(\boldsymbol{K}D - Z\boldsymbol{v}\right)\right] \\ &= d\left[\boldsymbol{x_{t_1}} + \frac{1}{Z-d}\left(d\boldsymbol{v} - Z\boldsymbol{v}\right)\right] \end{aligned}$$

The previous expression can be simplified leading to the following relation

$$Z\left(\boldsymbol{x_{t_1}} - \boldsymbol{x_{t_0}}\right) = d\left(\boldsymbol{x_{t_1}} - \boldsymbol{v}\right) \tag{3.11}$$

Let $\bar{\boldsymbol{x}}_{\boldsymbol{t_1}} = \boldsymbol{x_{t_1}} - \boldsymbol{v}$ denote the coordinate of $\boldsymbol{x_{t_1}}$ relative to the image centre (principal point) and $\Delta\boldsymbol{x} = \boldsymbol{x_{t_1}} - \boldsymbol{x_{t_0}}$ the disparity of the images of $\boldsymbol{X}$ between t_0 and t_1, then

$$\Delta\boldsymbol{x} = \frac{d}{Z}\bar{\boldsymbol{x}}_{\boldsymbol{t_1}} \tag{3.12}$$

Therefore, the point disparity $\Delta\boldsymbol{x}$ is directly proportional to its distance from the image center and the ratio of descent height to elevation d/Z. If $\boldsymbol{x_{t_1}}$ lies in the image center then $\bar{\boldsymbol{x}}_{\boldsymbol{t_1}}$=0, thus the disparity is always zero. For points satisfying $\bar{\boldsymbol{x}}_{\boldsymbol{t_1}} = 1$ pixel, to have a 0.5 pixel disparity a ratio $d/Z = 0.5$ is required. This means that the heights of two adjacent images should be halved to detect such disparity around the image centre.

3.3.1 Image Correlation

The correlation operation is vital to the recovery problem. The Sum of Squared Differences (SSD) and the Sum of Absolute Differences (SAD) are the most commonly employed statistical methods for obtaining correlation

$$C_{k_{SSD}}(x,y) = \sum_{m=x-W}^{x+W}\sum_{n=y-W}^{y+W}\left(I_1(m,n) - I_2^k(m,n)\right)^2 \tag{3.13}$$

$$C_{k_{SAD}}(x,y) = \sum_{m=x-W}^{x+W}\sum_{n=y-W}^{y+W}\left|I_1(m,n) - I_2^k(m,n)\right| \tag{3.14}$$

where $2W+1$ is the size of the correlation window. The correlation is computed on the area shared by the descent images, border excluded (half-width of the correlation window), for every virtual planar surface. Therefore the estimated depth value at each pixel is the depth z_k of the plane whose corresponding image pixel $C_k(x,y)$ is the smallest. To further refine the depth values, once the smallest $C_k(x,y)$ is identified, the two adjacent values $C_{k-1}(x,y)$ and $C_{k+1}(x,y)$ (left and right side of $C_k(x,y)$, respectively) are employed together with $C_k(x,y)$ to approximate the correlation curve by a second-order polynomial. Therefore, the depth value with the "subpixel" adjustment can be computed [44] as:

$$z\left(x,y\right) = z_k + \frac{\delta z\left(C_{k+1}\left(x,y\right) - C_{k-1}\left(x,y\right)\right)}{2\left(C_{k+1}\left(x,y\right) + C_{k-1}\left(x,y\right) - 2C_k\left(x,y\right)\right)} \tag{3.15}$$

where δz is the depth increment between adjacent planar surfaces.

3.4 Descent Images

As previously mentioned, the input image pair is characterized by a large resolution difference. Indeed the surface area of 1 pixel in image I_2 (2nd pose) represents exactly a quarter of the area covered by 1 pixel belonging to I_1 (1st pose). Synthetic surface images, rendered in the PANGU planetary scene generator [41], have been considered. This software tool has been developed at the Space Technological Centre at the University of Dundee, to simulate the surface of various planetary bodies. Given position and orientation of a spacecraft, PANGU generates realistic surface images of heavily cratered planetary bodies such as the Moon. An example of input image pair is shown in Figure 3.3. Initially, a surface model is created (which may be entirely synthetic or it may be based on existing low resolution DEMs) only then features like craters and boulders are applied onto such model.

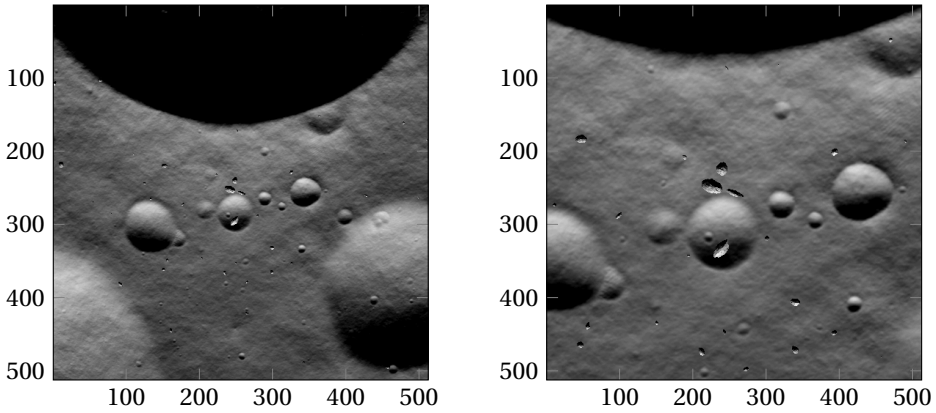

Figure 3.3: Example of a PANGU input image pair. Image I_1 taken at 1200 m above the surface (left) and I_2 taken when the altitude is halved (right)

Different scenes have been generated, all corresponding to a 30° FOV projective camera with 512×512 pixel 8-bits sensor (Table 2.2). The aforementioned sensor size has been found a good compromise between the level of terrain surface details and the demanded computational time for processing it [37]. Under this assumptions the intrinsic camera parameters have been set as follows:

$$\boldsymbol{K} = \begin{bmatrix} f & 0 & u \\ 0 & f & v \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 955.4050 & 0 & 256 \\ 0 & 955.4050 & 256 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{with} \quad \boldsymbol{K} = \text{px} \tag{3.16}$$

Chapter 4

Software

Through the modelling presented in the foregoing chapter, an HD algorithm has been implemented and tested. The software developed is written in *MATLAB* language and is able to compute hazard maps from two adjacent descent images. A modular structure has been chosen, allowing easier verification and a more efficient integration and extension. The current chapter details the software architecture, with all modules outlined in a coherent structure in Section 4.1. In addition, Section 4.2 discusses the challenges faced during depth recovery, as well as the choices made to overcome these. Finally, the methods selected for computing shadow, slope, and roughness maps within the framework of planetary HD are highlighted in Section 4.4

4.1 Software Architecture

To better structure the architecture of the software (Figure 4.2), 2 high-level sections have been identified: *DEM Computation* and *Hazard Detection*. Besides, the entire system is embodied by only eight main elements (blocks portrayed in light red), namely *Rock Detection*, *Image Warping*, *Correlation*, *Shadow Detection*, *Slope Computation*, *Roughness Computation*, *Mapping*, and *Georeferencing*. A detailed description per block will be given in the following sections.

- **Rock Detection** applies Harris Corner measure to determine the presence of rocks, boulders, and their immediate nearby pixels. The function handles the input image at camera 1st pose and delivers as output its related boolean rock map. A pixel value of 1 is associated with the presence of rocks, instead 0 for no detection.
  Input:
  - I1, image at camera 1st pose

  Output:
  - I1_rockMap, rock boolean map of the image at camera 1st pose [/]

- **Image Warping** warps the image taken at camera 2nd pose back to the first image plane. Through this block, I_2 is made resolution equivalent and aligned to the image I_1, therefore ready for correlation.
  Input:
  - I2, image at camera 2nd pose
  - invP_k, inverse of the homography induced by the k-th virtual parallel plane

  Output:
  - I2_k, warped version of the image taken at camera 2nd pose

- **Correlation** represents the last stage of the *DEM Computation*. The Sum of Absolute Differences (SAD) is employed as correlation operator.
  Input:
  - I1, image at camera 1st pose
  - I2_k, warped version of the image taken at camera 2nd pose
  - I1_rockMap, rock boolean map of the image at camera 1st pose [/]

  - Ax, Ay, Bx, Cy, vertices limits for correlation [px]
  - W, width correlation window from its center [px]
  - C_k, array to store correlation coefficients
  - n_slice, number indicating the current slicing surface

  Output:

  - C_k, array to store correlation coefficients

- **Georeferencing** is employed to assign real-world coordinates to each pixel of the raster. In other words, it represents the process of projecting the results from the image plane to the world coordinate frame.
  Input:
  - Z_e, computed depth map [m]
  - h1, altitude camera at 1st pose [m]
  - cameraTilted, logic variable for the imaging sensor offset
  - Oworld, origin world frame (X, Y) [m]
  - theta, line-of-site total offset [rad]
  - FOV, field-of-view [rad]
  - imgSize, image size [px]
  - W, width correlation window from its center [px]
  - nPX, number of pixel from the border to be neglected [px]
  - Ax,Ay,Bx,Cy, vertices limits [px]

  Output:
  - Xgeo, x-real-world-coordinates [m]
  - Ygeo, y-real-world-coordinates [m]

- **Shadow Detection** extracts shadow hazards through *thresholding*, which consists in comparing each image pixel with a selected threshold value.
  Input:
  - I1, image at camera 1st pose
  - cameraTilted, logic variable for the imaging sensor offset
  - Z_e, computed depth map [m]
  - K1,K2, indices of trapezoid lateral sides [px]

  Output:
  - shadowMap, shadow boolean map [/]

- **Slope Computation** adopts the well-known linear regression method to assess the slope of the surface terrain. Based on the safety constraints previously defined in Section 2.8, the slope values are re-scaled from 0 to 1.
  Input:
  - Z_e, computed depth map [m]
  - Xgeo, x-real-world-coordinates [m]
  - Ygeo, y-real-world-coordinates [m]

  Output:
  - z_mean, mean plane depth values [m]
  - slopeMap, slope map [rad]
  - hMap_slp, hazard slope map [/]
  - Wx,Wy, width slope window from its center x,y-direction [px]

- **Roughness Computation** identifies additional terrain features that differ from the local mean plane. Similarly to the slope computation block, roughness values are re-scaled from 0 to 1.
  Input:
  - Z_e, computed depth map [m]
  - z_mean, mean plane depth values [m]

  - Wx,Wy, width slope window from its center x,y-direction [px]

  Output:

  - rghMap_DEM, roughness map [m]
  - hMap_rgh_DEM, hazard roughness map [/]

- **Mapping** allows to combine the shadow, slope, and roughness hazard maps into a single map indicating the hazardousness of the input subspace, on a scale from 0 to 1. Therefore, for each pixel, it is a hazard value increasing from 0 to 1 (unsafe).
  Input:

  - Z_e, computed depth map [m]
  - shadowMap, shadow boolean map [/]
  - hMap_slp, hazard slope map [/]
  - hMap_rgh_DEM, hazard roughness [/]

  Output:

  - hazardMap, map assessing the hazardousness of the landing region [/]

For the sake of simplicity, Figure 4.2 only depicts a schematic of the software architecture: indeed, it can be clearly seen that only the main inputs/outputs (I/O) are displayed. Nevertheless, the flow of information and data shown is relevant to the flow within the developed software. As previously stated, a detailed description per block is given in the following sections, therefore allowing to better understand of the software architecture too.

The legend for the software block diagram is presented as follows (Figure 4.1). It is important to notice that all inputs and outputs (I/O) are characterized by a trapezoidal form. The different color is to highlight blocks that can be seen either as input and output (azure) from the true initial software inputs. For the latter, a further distinction is made to separate input describing camera parameters (light blue) from input depicting perturbations (light green).

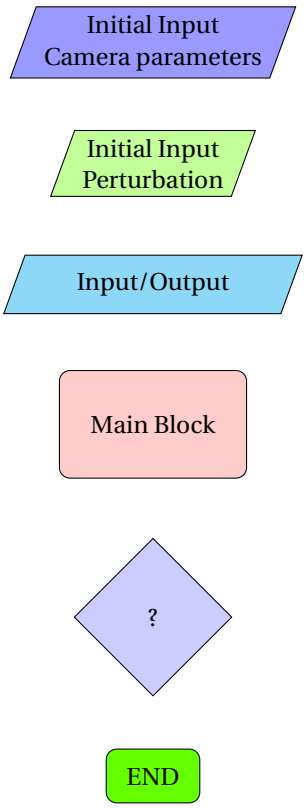

Figure 4.1: Legend for the Software Block Diagram

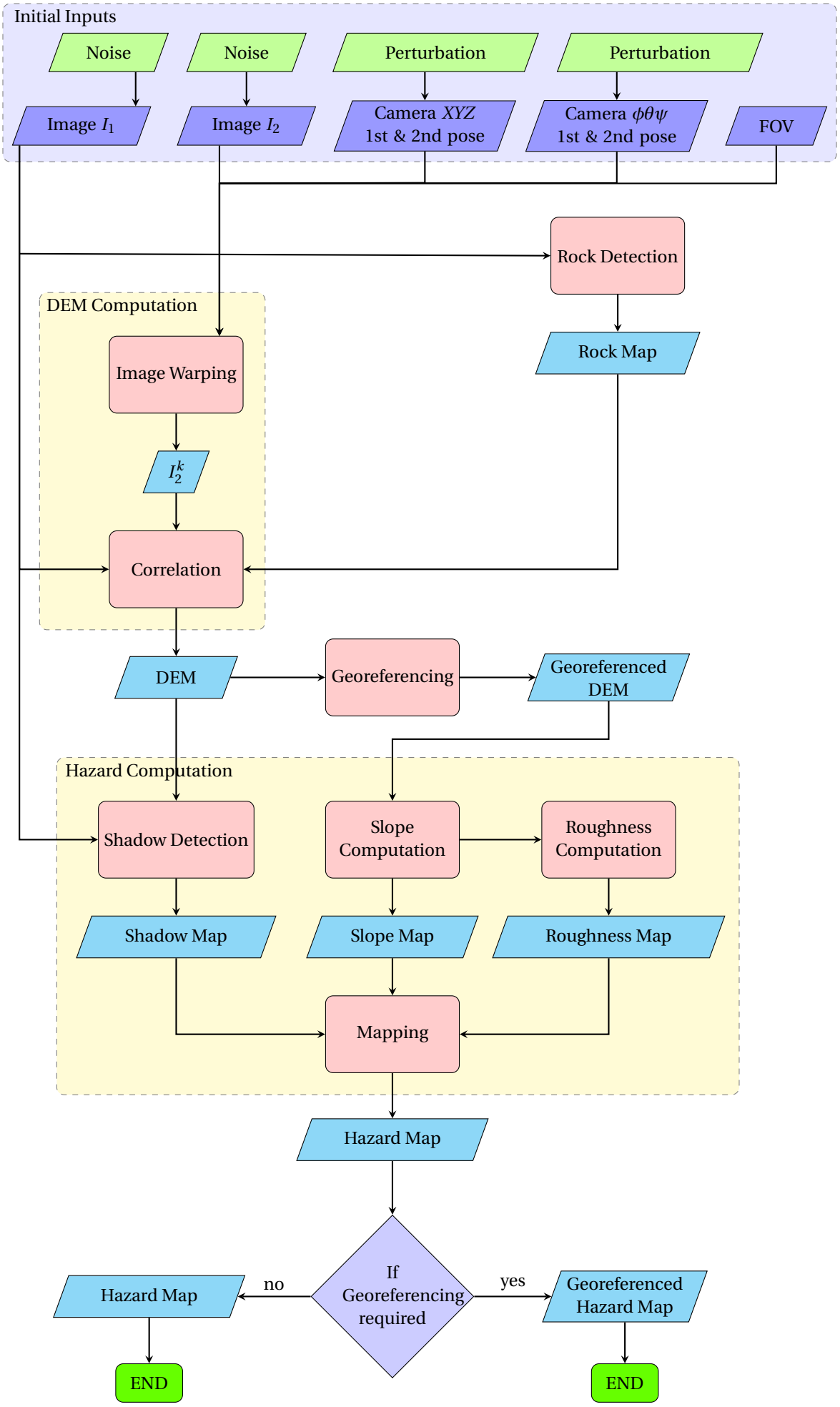

Figure 4.2: SfM HD Software Architecture

4.2 Challenges in Shape Recovery

Several challenges, related to the *DEM Computation* part, have been faced. Being one of these and furthermore a crucial step for the correct depth assessment, *image warping* is here extensively analyzed. First and foremost, image warping represents a geometric transformation between two images, a source and a target image. [34] The geometric transformation defines the relationship among the source and the target pixels. Hurdles have arisen due to lack of information in [43] concerning its implementation: both from a strategy-wise point of view (direct/reverse mapping) and in how to handle pixel locations expressed through real numbers, due to the homography computation. The strategy of reverse mapping (Figure 5.2) has been chosen, where the target pixels in raster order are inversely mapped to the source image and sampled accordingly. [34]

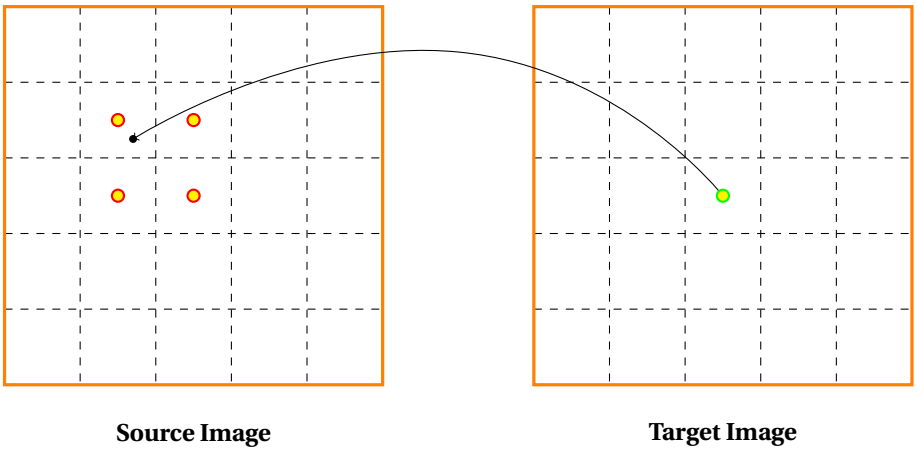

Figure 4.3: Scheme of reverse warping

For every z_k, belonging to the space of the depth search, the homography $\boldsymbol{H_k}$ (Equation 3.8) is computed together with its inverse $\boldsymbol{H_k^{-1}}$. A matrix of the same size of I_2 is pre-allocated for its warped version, namely I_2^k: therefore, for every location $(u, v) \in I_2^k$ the corresponding pixel $(u_k, v_k) \in I_2$ is determined as follows

$$[u_k, v_k, 1]^T = \boldsymbol{H_k^{-1}} [u, v, 1]^T \tag{4.1}$$

$$[u_k, v_k] = \left[\frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + h_{33}}, \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + h_{33}} \right]^T \tag{4.2}$$

where h_{ij} are simply the elements of the matrix $\boldsymbol{H_k^{-1}}$, associated with the k-th parallel planar surface.

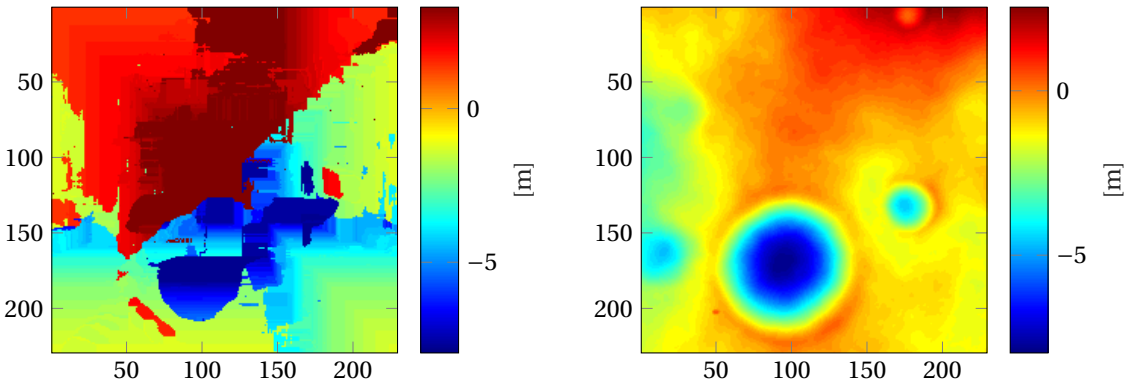

Figure 4.4: Computed DEM through warping accomplished by rounding coordinates to the nearest integer (left) ground-truth (right). Altitude 300 m

It must be highlighted that u_k, v_k may not be integers, therefore a way to express pixel locations needs to be found. Rounding the coordinates to their nearest integers led to wrong 3D reconstructions (Figure 4.4) since not correct gray-scale intensity were associated with the pixels belonging to I_2^k. On the other hand, bi-cubic interpolation was proved too computationally expensive (tens of minutes for a single DEM computation) for the tight real-time constraints involved (Section 2.8) and therefore it was discarded already at an early stage of the software implementation. Bilinear interpolation with the 4 nearest pixel values has been found optimal, therefore it is employed for the intensity assessment

$$I_2^k(u, v) = I_2(u_k, v_k) \tag{4.3}$$

Figure 4.5 shows an input image I_2 and its warped version I_2^k, computed for a specific depth z_k. During the first software performance assessment on depth recovery, an unusual squared pattern has emerged on the computed DEMs (Figure 4.6). Further study has shown that such figures arise in presence of small features (e.g. rocks and boulders), characterized by high differences in their gray-scale intensities from pixel to pixel.

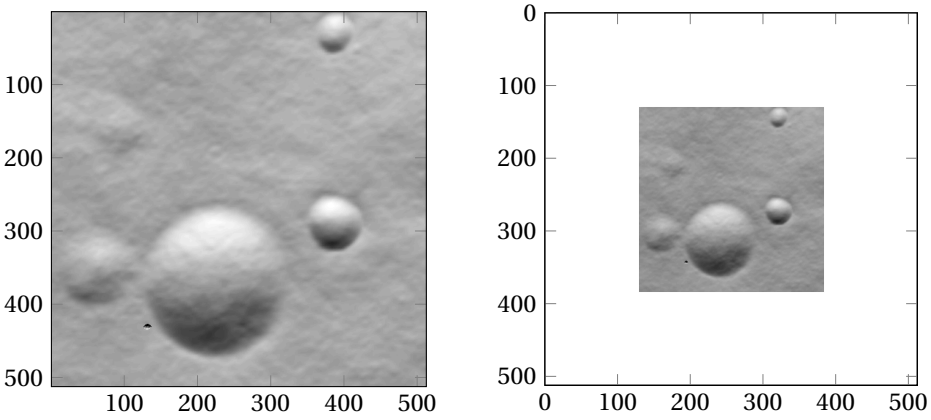

Figure 4.5: Input image I_2 (left) and its warped version I_2^k computed for z_k = 600 m (right)

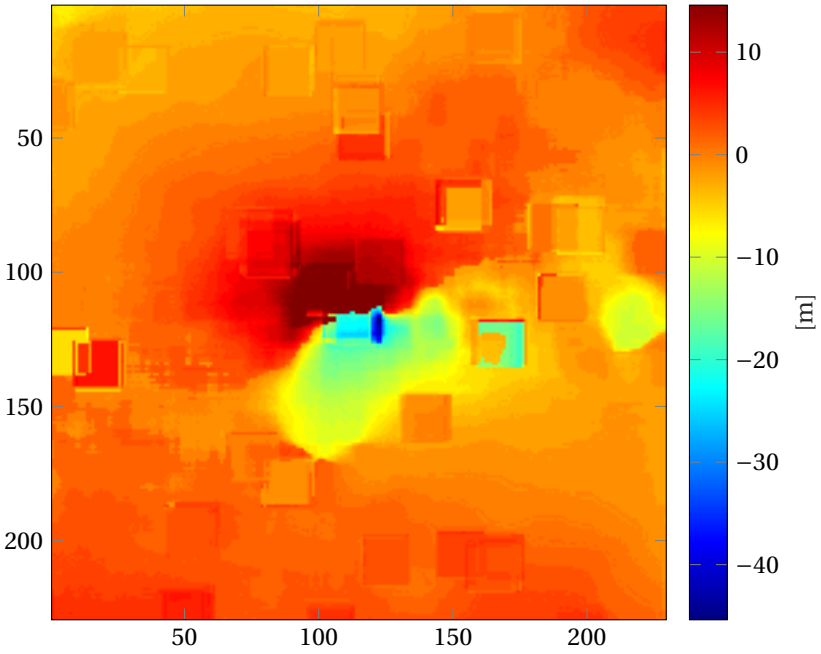

Figure 4.6: Squared pattern on a computed DEM with SSD. The length of the squares sides equal the size correlation window (17 pixels)

Furthermore, the length of the square side has been shown to perfectly match the size of the correlation window employed. The reason behind the phenomenon is associated with the limitation of projective warping implemented through bilinear interpolation. This can be clearly observed in Figure 4.7, where a zoom for a specific boulder is shown for both the image taken at camera 1st pose and a warped version of the image taken when the altitude is halved. It can be seen that the pixel highlighted on the right image has a gray-scale intensity of 50, instead the corresponding pixel on the left image (highlighted in red) has a gray-scale intensity of 0. Therefore, it can be concluded that the warping does not perform well since corresponding pixels are characterized by large differences in the intensity values. Figure 4.8 shows how such differences are responsible for high correlation costs (order of $\sim 10^4$). The match between the length of the square side and size of the correlation window also finds an explanation: for every correlation window, including the boulder, such high differences in pixel intensities drive the algorithm in the selection of the same boulder's depth value for their central pixel. As result, a levelling of all depth values is achieved within an area (located around the boulder) that equals the size of the correlation window employed.

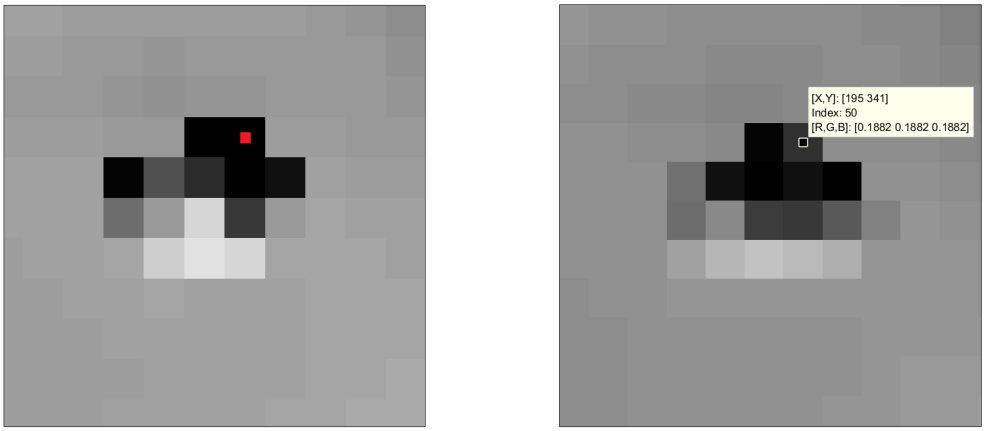

Figure 4.7: Zoom in for gray-scale pixel intensities depicting a specific boulder. Image taken at camera 1st pose (left) and a warped version of the image at camera 2nd pose, computed for $z_k = 600$ m (right)

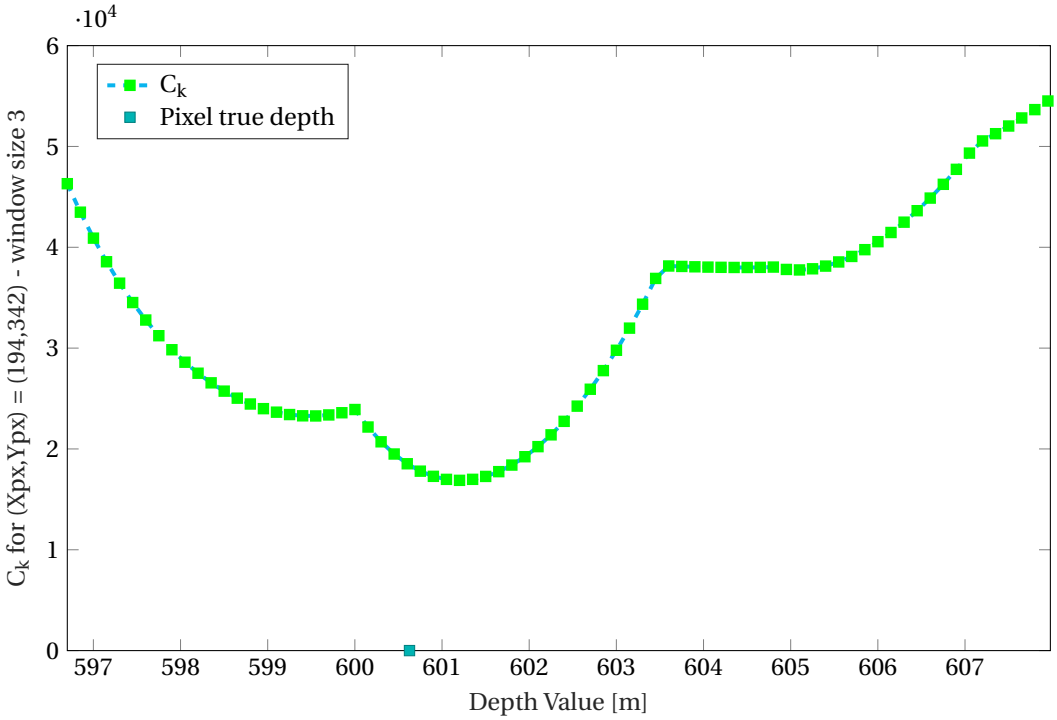

Figure 4.8: SSD cost for boulder pixel (194,342) versus Depth expressed in the camera frame. A 3×3 correlation window was employed

To tackle the problem, pixels depicting boulders and their immediate proximity are excluded from correlation (they provide a null contribution) and Not-a-Number (NaN) values are instead associated with their elevations. As outcome, we lose the opportunity to assess rock and boulder depth values and we directly consider such locations as hazardous. This will be further explained in Section 4.4.5. In addition, the SSD has been replaced with the SAD operator: by taking the absolute difference of the pixel intensities, rather than their squares, the impact of further potential outliers is indeed minimized (Figure 4.10).
To achieve this, it is mandatory to detect boulders directly at an early stage. Therefore, a *Rock Detection* block has been created to deliver the boolean rock maps of the images captured at 1st camera pose (Figure 4.2). Since pixels depicting rocks and boulders are characterized by large intensity differences from pixel to pixel, the Harris corner detector has been selected and implemented. The related theory has been extensively discussed during the literature research of the project [6] and is presented again in Appendix C for sake of completeness. In plain words, its working principle consists in computing the *second moment matrix M*

$$M = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \tag{4.4}$$

In order to detect features, the eigenvalues of M should be both large. Rather than enforcing a minimal value for the smallest eigenvalue, Harris and Stephens [10] suggested to compute the following *corner response function*

$$R = \det(M) - \alpha \mathrm{Tr}(M) \tag{4.5}$$

where Tr represents the trace of the matrix (e.g. the sum of its diagonal terms), while α a constant whose value ranges from 0.04 to 0.06. Features are then identified as points with large corner response (R > threshold) and a related boolean map is thus obtained. Concerning its implementation, the image derivatives have been computed using a small convolution filter of size 3×3.

$$\frac{\partial I}{\partial x} = dx * \mathrm{img} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * \mathrm{img} \tag{4.6}$$

$$\frac{\partial I}{\partial y} = dy * \mathrm{img} = dx' * \mathrm{img} \tag{4.7}$$

where $*$ denotes the 2 dimensional convolution operation, while *img* represents the input image. Concerning the threshold for the corner response function, it is obvious that different images are characterized by different values of Equation 4.5. This makes the selection of a threshold suitable for the descent images difficult. Inspecting the corner response function for different images has led the selection of the following threshold value

$$\mathrm{threshold} = k\mu \qquad \text{with} \quad k = 11 \tag{4.8}$$

where μ denotes the mean of the corner response function values. Therefore, there is not a single threshold for all the descent images but rather an optimal threshold is computed each time. An example of rock map is shown in Figure 4.9.

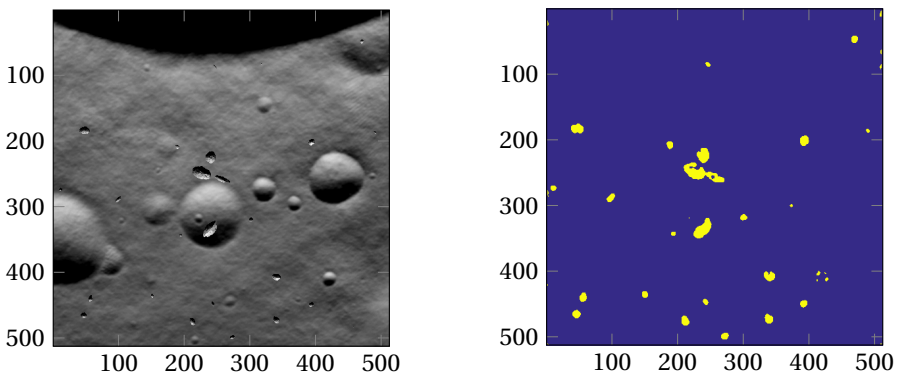

Figure 4.9: Input image (left) and its computed boolean rock map (right). Pixels depicting boulders and their immediate proximity are highlighted in yellow

The developed approach succeeds in solving the issue arisen. Figure 4.10 indeed highlights that the square features of the previously computed DEM (Figure 4.6) are now no longer present. On the contrary, NaN values have been associated with the boulder elevations which are depicting in white color.

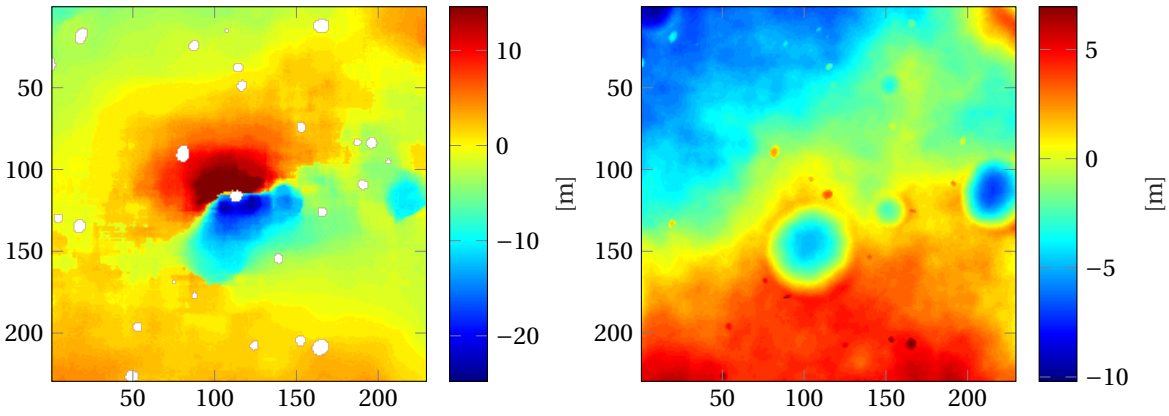

Figure 4.10: Computed DEM (left) and ground-truth (right). Altitude 500 m

4.3 Parametric Analysis

In Section 3.3.1, it was marked how correlation is vital to the recovery problem. Therefore, it is important to investigate which correlation window size allows a better reconstruction. To this end, window sizes from 3 to 49 pixels (values referred to the length of the squared window side) were considered while the depth step was initially chosen equal to half of the System Level 0 requirement on roughness (Δz = 15 cm). The analysis has been assessed by comparing the reconstructed DEM of the imaged scenes with respect to its ground-truth. PANGU, indeed, provides a 3D representation of the terrain free from errors and noise. Two different input image pairs, representative of the variability of the landing terrain besides of the altitude range for precise HDA assessment, have been considered for the analysis (Appendix D):

- *Scene 1* contains multiple craters and boulders of different size and shape. The input image pair has been taken at relatively low altitudes: 150m elevation for camera 1st pose and 75m elevation for camera 2nd pose, respectively.
- *Scene 2* presents two large craters on an otherwise flat terrain. The lower the spacecraft goes the more complex the scene appears since boulders are displayed with increasing resolution. The image at 1st camera is taken at 600m altitude.

For the reconstruction quality assessment the mean value μ (based on absolute errors), median (based on the errors taken with their own sign), and the median absolute deviation (MAD) have been selected. The choice of these statistical parameters is further strengthened when inspecting that the DEM error distributions are clearly not normal (Figures 4.11 and 4.12): indeed, skewness and kurtosis are always different from zero for both scenes (Tables 4.1 and 4.2).

Table 4.1: Scene1: DEM error skewness and kurtosis

Window Size px^2	**Skewness**	**Kurtosis**
17×17	0.96329	-2.0367
19×19	0.6321	-2.3679
21×21	0.59238	-2.4076
23×23	0.60575	-2.3943

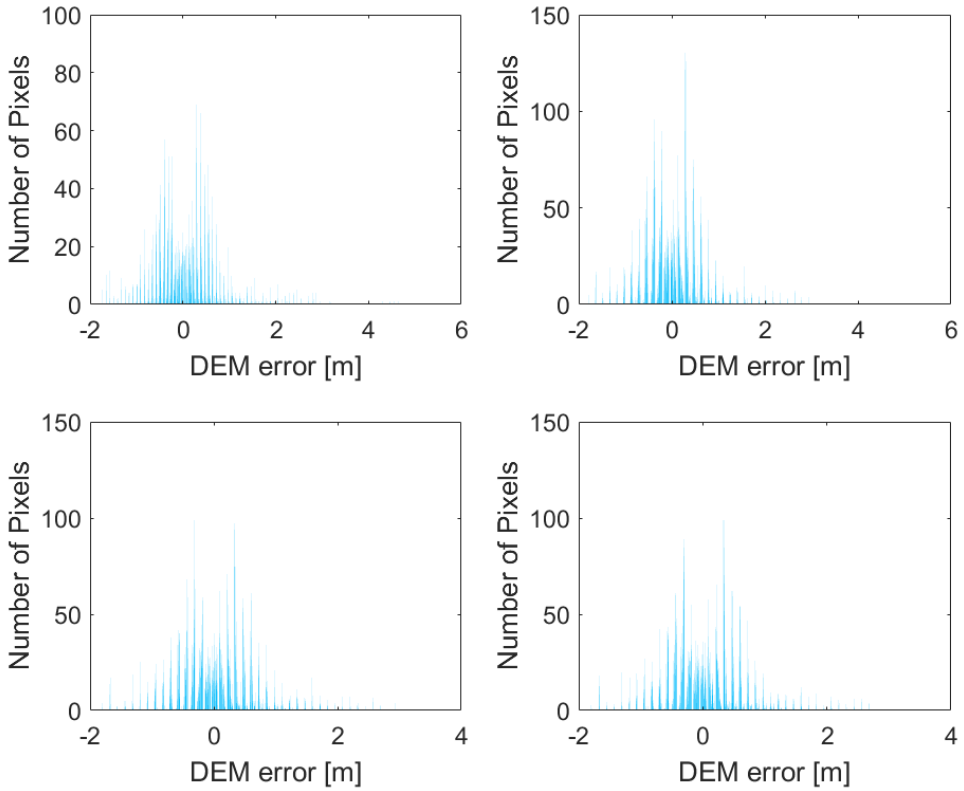

Figure 4.11: Scene 1 DEM error distributions. Results for correlation window sizes W = 17×17 px^2 (top-left), W = 19×19 px^2 (top-right), W = 21×21 px^2 (bottom-left), W = 23×23 px^2 (bottom-right)

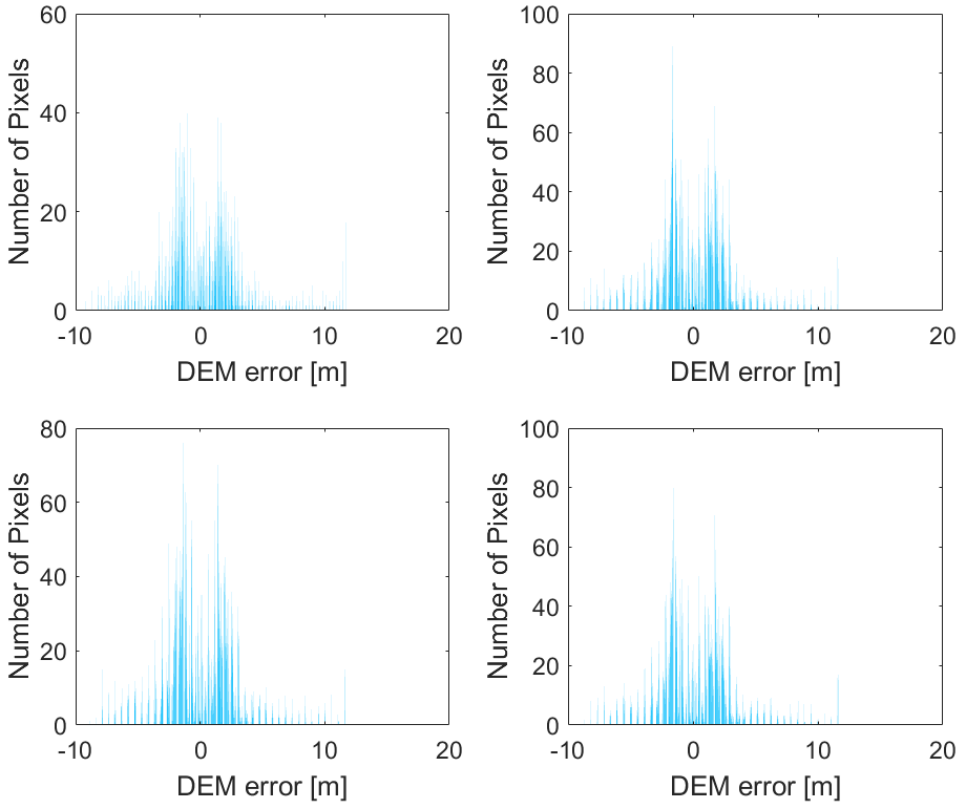

Figure 4.12: Scene 2 DEM error distributions. Results for correlation window sizes W = 11×11 px^2 (top-left), W = 13×13 px^2 (top-right), W = 15×15 px^2 (bottom-left), W = 17×17 px^2 (bottom-right)

Table 4.2: Scene2: DEM error skewness and kurtosis

Window Size px^2	Skewness	Kurtosis
11×11	0.53946	-2.4605
13×13	0.55127	-2.4487
15×15	0.55103	-2.449
17×17	0.54362	-2.4564

As expected, the two scenes lead to different optimal values for the size of the correlation window: 19 px for scene 1, in contrast to 13 px for the other scene (Figure 4.13). Notwithstanding, a better inspection of the DEM mean error μ_{DEM} shows that the optimal window size values are bounded within the interval 17-23 px for scene 1, and 11-17 px for scene 2. Clearly, the 17×17 px^2 correlation window stands out as a good candidate. Moreover, within both intervals, the MAD (Figure 4.14) reveals that the error-data values are less dispersed for decreasing number of pixels. As outcome of the analysis, the 17×17 px^2 correlation window is considered optimal.

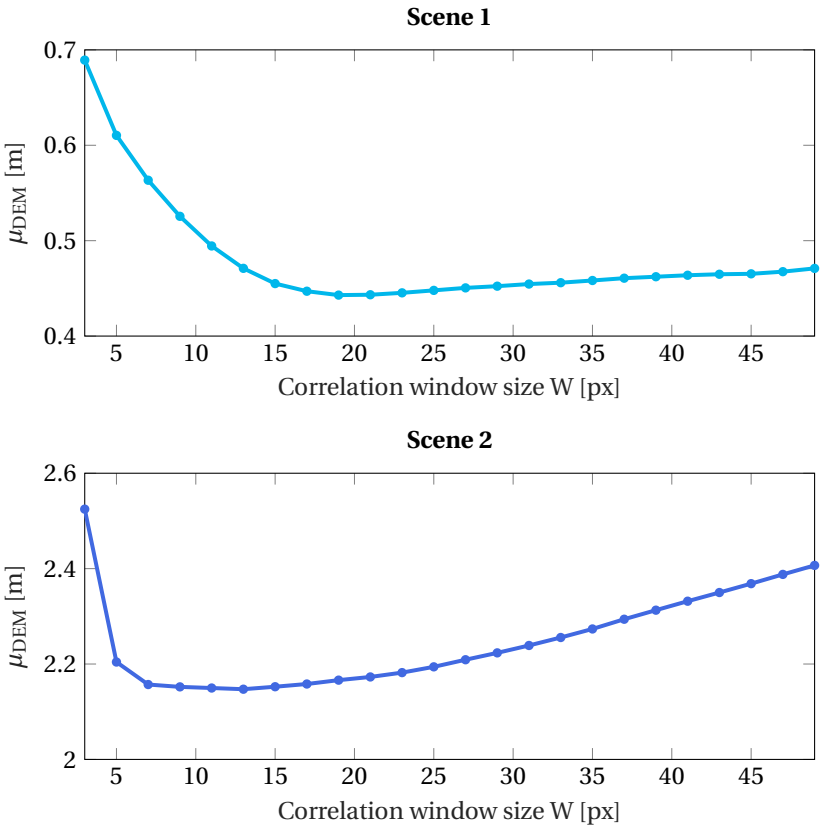

Figure 4.13: Parametric Analysis on correlation window size. DEM Mean error μ for 2 different PANGU input image pairs

It must be noticed that, so far, the variable time has not been taken into account. Clearly, increasing the size of the correlation window leads to a greater number of calculations and thus to a higher computational time. As a matter of fact, the function in charge of the correlation requires approximately 7 s for the 3×3 px^2 window, on the contrary, approximately 90 s for the 49×49 px^2 window. It is important to bear in mind that the software is written in *MATLAB* language. Furthermore, the implementation chosen by the author for correlation is probably not the fastest and a better coded function is believed to reduce the required computational time. Nevertheless, the candidate 17×17 px^2 window is rather small, leading to a total time of ~12 s for correlation. Therefore, this function will be employed in every other simulation.

The previous result is in contrast with the outcome of [43], though, where fairly large windows of 25 to 45 pixels were common in their experiments *(Clark F. Olson, personal communication)*. This can be explained through illumination differences and reflections which commonly occur when synthetic input images are replaced by real images, causing larger correlation windows to yield better recovery [15]. The personal communication, concerning the correlation window size employed in the experiments at JPL [43], is extremely valuable since allows to conclude that also larger window size can meet real-time

requirements and so the 17×17 px^2 window chosen.

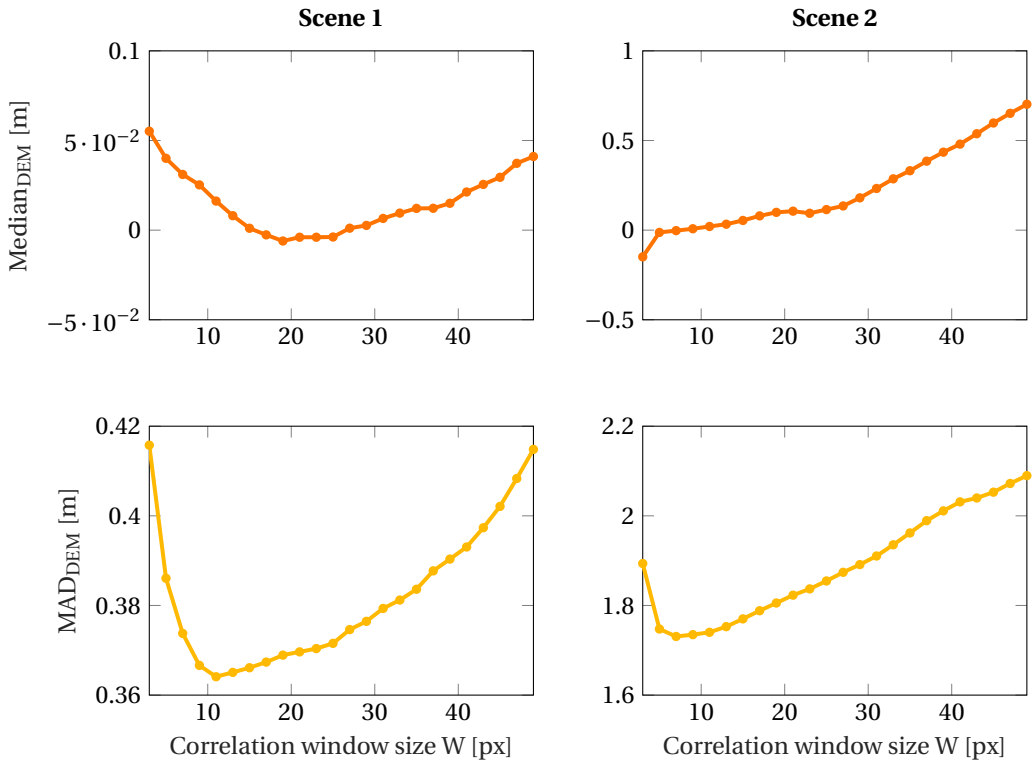

Figure 4.14: Parametric Analysis on correlation window size. DEM Median error and MAD for 2 different PANGU input image pairs

Obviously, the quality of the reconstructed DEM also depends on the depth step Δz: a small step size is expected to yield better results nevertheless at the expense of a higher computational load, which is in contrast with the tight real-time constraints. Therefore, it is mandatory to assess what elevation step is optimal. Considering the correlation window size (17 × 17 pixels) previously identified, the DEM mean, median, and MAD error have been computed for Δz starting from 2.5 cm till 50 cm. For the sake of conciseness, only the results for Scene 1 are here shown (Figure 4.16). The DEM mean error μ_{DEM} gradually lowers for decreasing depth steps. Nevertheless, for $\Delta z < 15$ cm the error reduction is not such to justify the inevitably higher computational time (Figure 4.15). Moreover, at $\Delta z = 15$ cm the median absolute deviation MAD$_{DEM}$ indicates that the error deviation is minimum. Therefore, a depth step $\Delta z = 15$ cm (half of the System Level 0 requirement on roughness) has been selected as optimal.

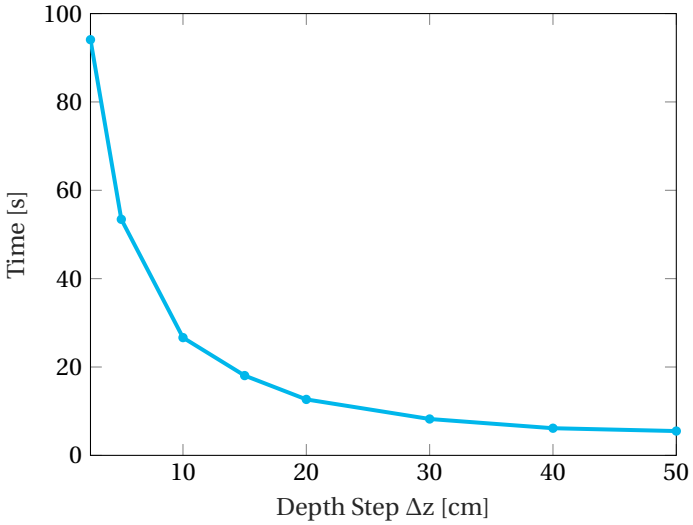

Figure 4.15: Effect of step size Δz on the shape recovery computational time. Time required only by the function in charge of the 3D reconstruction.

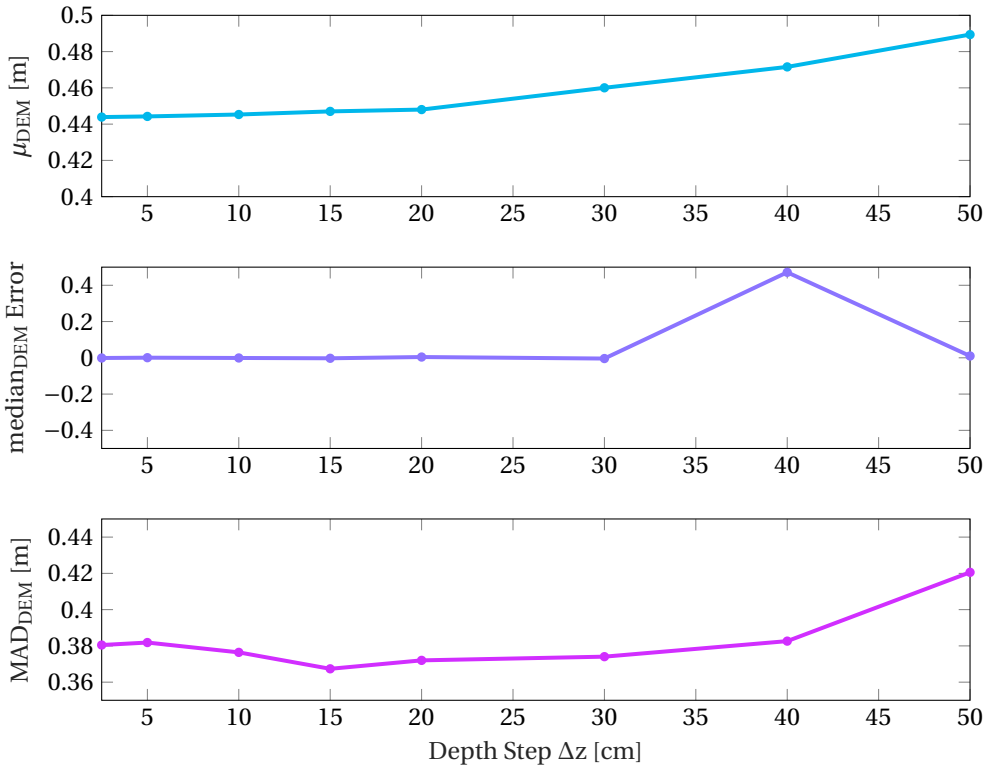

Figure 4.16: Parametric Analysis on Δz size. DEM mean error μ (top) median (center) and MAD (bottom) for PANGU Scene 1 input image pair

4.4 Hazard Computation

The *Hazard Computation* class (Figure 4.2) consists of four main tasks:

- Shadow Detection
- Slope Computation
- Roughness Computation
- Mapping

for each block different methods are available and a first selection was carried out during the literature research of the project. In the following, the ultimate trade-offs and the selected approaches are presented with insights on their implementation. To draw the attention only on how hazards are detected, the ground-truths rather than the computed DEMs have been here employed.

4.4.1 Shadow Detection

Shadow hazards can be simply extracted through *thresholding* which consists of comparing each pixel intensity with a selected threshold value. Since images taken at higher altitudes are generally darker than images captured at a lower elevation above the surface, an optimal intensity threshold needs to be selected. Analyses of pixel intensity distribution for several data set have led the selection of the following threshold value

$$\text{threshold} = \frac{\mu + \sigma}{2} \tag{4.9}$$

where μ and σ denote the mean and the standard deviation of the image intensity values, respectively. Therefore, there is not a single threshold for all the descent images but rather an optimal threshold is computed each time. The boolean shadow map is created by setting white, pixels whose intensity is lower than the threshold. Viceversa, pixels with higher intensities are depicted in black (Figure 4.17).

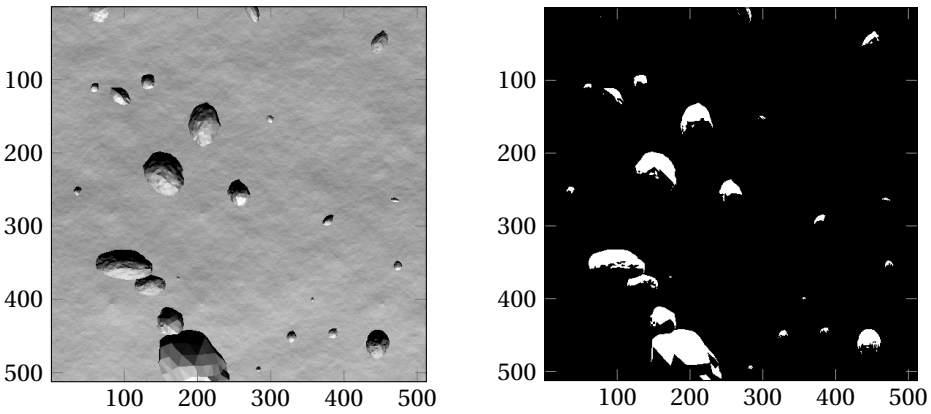

Figure 4.17: Input image (left) and its computed boolean shadow map (right). Pixels in shadow are highlighted in white

4.4.2 Slope Computation

Several approaches [33] are available to compute the slope. These have been considered in the literature research of the project [6], which has pointed out that methods such as *steepest descent, third order finite difference,* and *partial quadratic equation* produce very large errors, overestimating the slope by more than 200% [38] in presence of noise and boulders. In [38] it was also shown that despite *intelligent mean plane* algorithms better estimate slope, the gain obtained does not justify an execution time increased by at least a factor 4 respect to *linear regression.* Based on this analysis, linear regression was selected to compute the slope of the terrain. The local surface is approximated through a plane, which equation is presented below for sake of completeness

$$aX + bY + cZ + d = 0 \tag{4.10}$$

where Z denotes the "depth" of a point on the plane, while X and Y are the x- and y-coordinate of the plane, respectively. The previous expression can be re-written by solving for the Z variable as follows:

$$Z = pX + qY + r \tag{4.11}$$

Equation 4.11 can be expressed in matrix form $\boldsymbol{Ax} = \boldsymbol{b}$

$$\begin{bmatrix} X_1 & Y_1 & 1 \\ X_2 & Y_2 & 1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & 1 \end{bmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} Z_1 \\ z_2 \\ \vdots \\ Z_n \end{pmatrix} \tag{4.12}$$

where the least-square (LSQ) solution for $\boldsymbol{x} = \left[p, q, r\right]^T$ is commonly computed through Singular Value Decomposition (SVD) [32] (see Appendix E). The slope S, of the considered patch, is simply obtained from

$$S = \arccos\left(\frac{\left|\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{Z}}\right|}{\left|\hat{\boldsymbol{n}}\right|\left|\hat{\boldsymbol{Z}}\right|}\right) \tag{4.13}$$

where $\hat{\boldsymbol{n}} = \left[p, q, -1\right]^T$ represents the unit vector normal to the surface, while $\hat{\boldsymbol{Z}}$ is the unit vector of the Z axis. The number of points considered and thus the size of the plane is determined by the lander footprint: in accordance with other studies [38], a 3 m footprint has been assumed. The slope of the resulting plane is then the slope at the central pixel of the window. This method represents a good compromise between accuracy and computational speed, an example of computation is shown in Figure 4.18. Based on the 15° safety constraints, the slope hazard map is computed: with values increasing from 0 (safest) to 1 (most unsafe, ≥15°)

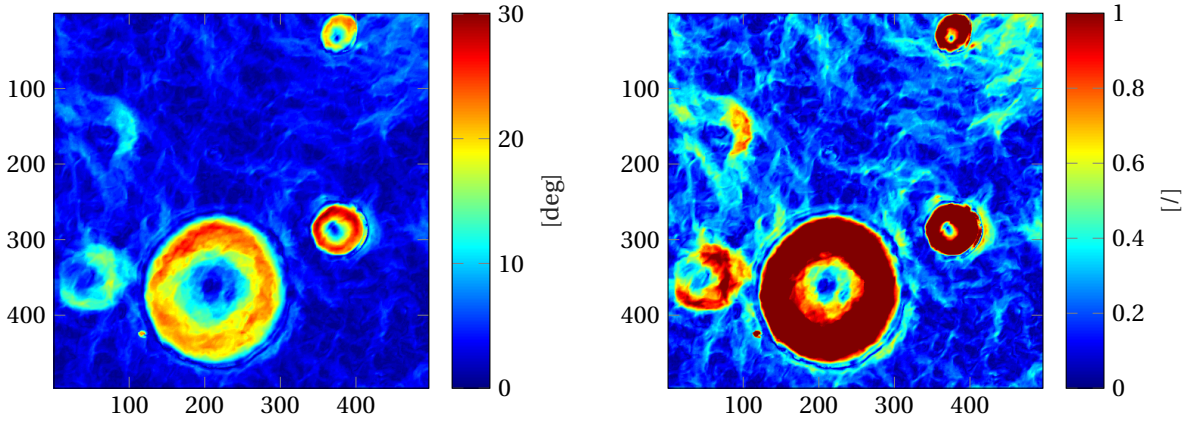

Figure 4.18: Ground-truth slope map computed through linear regression method (left) and map re-scaled using a grading scheme from 0 to 1 (right), based on the 15° slope constraint

4.4.3 Georeferencing

To compute the slope, real-world coordinates need to be assigned to each pixel of the image. This is achieved through the *georeferencing* block (Figure 4.2) where the computed depth map is projected from the image plane to the world coordinate frame, therefore obtaining a georeferenced DEM (Figure 4.21). Considering Figure 4.19 we want to compute the real-world coordinates X_P, Y_P of the image point P. Through simple trigonometry it can be easily proved the following

$$X_P = \frac{h}{\cos(\theta + \alpha)} \tan(\beta) \tag{4.14}$$

$$Y_P = h \tan(\theta + \alpha) \tag{4.15}$$

with h the camera elevation above the surface, θ the imaging sensor line-of-sight offset (for a camera pointing along the nadir direction $\theta = 0°$), α and β the camera FOV along the x-,y-direction, respectively. According to Table 2.2, $\alpha = \beta$. For each set of descent images, the center of the first image of the sequence is taken as the origin of the world frame unless stated otherwise. Nevertheless, the two equations formerly outlined are incomplete. Figure 4.20 shows how the terrain elevation values affect the ground coordinates: as a matter of fact, when in presence of a crater the true ground coordinates increase, vice-versa decrease when in presence of a rock or boulder.

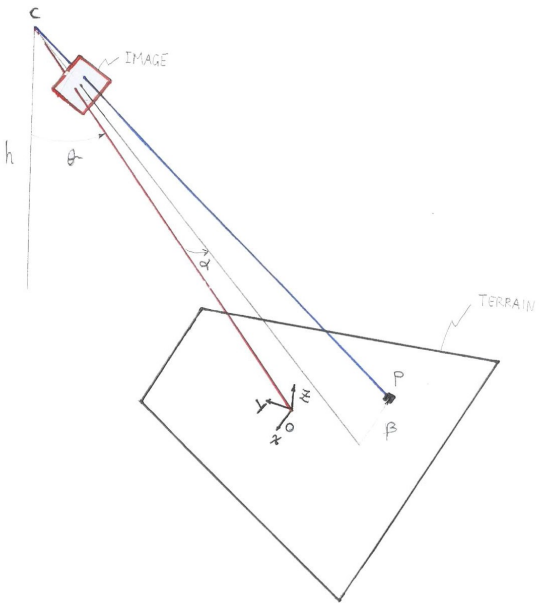

Figure 4.19: Georeferencing: schematic representation

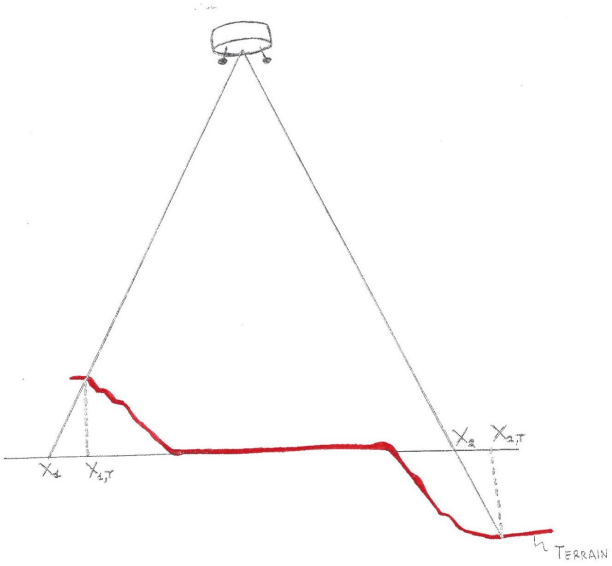

Figure 4.20: Georeferencing: depth values affect the ground coordinates

Therefore Equation 4.16 and 4.17 are completed introducing the correction terms *dX*,*dY* which accounts for the effect of the depth values.

$$X_P = \frac{h}{\cos(\theta+\alpha)}\tan(\beta) + dX = \frac{h}{\cos(\theta+\alpha)}\tan(\beta) - \frac{Z_P}{\cos(\theta+\alpha)}\tan(\beta) \tag{4.16}$$

$$Y_P = h\tan(\theta+\alpha) + dY = h\tan(\theta+\alpha) - Z_P\tan(\theta+\alpha) \tag{4.17}$$

where Z_P is the depth value of point P considered with its own sign.

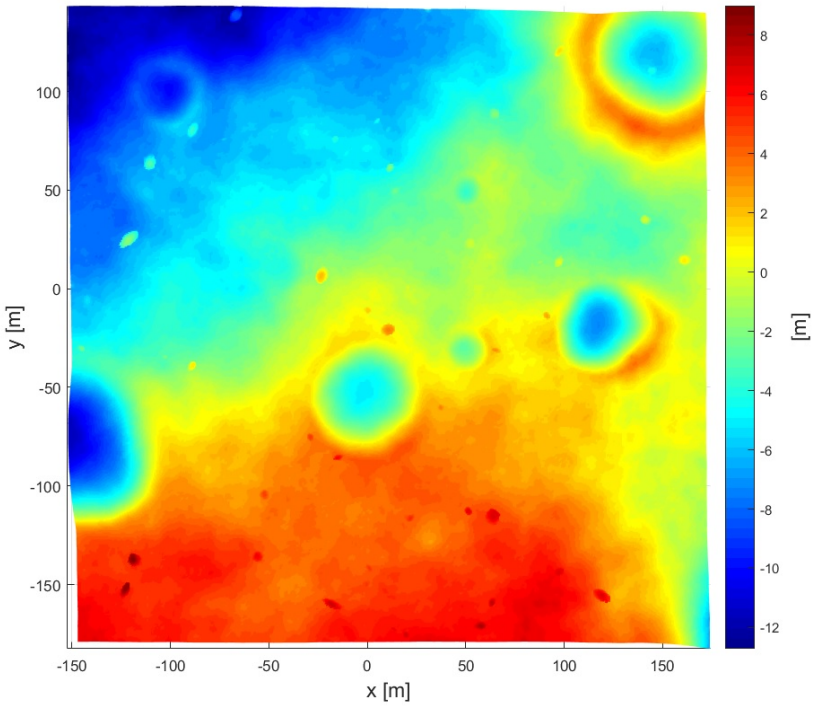

Figure 4.21: Example of a georeferenced DEM. Coordinates are expressed in the world coordinate frame

4.4.4 Roughness Computation

Hazards are not only caused by shadow and slope, but also by roughness which originates from the presence of rocks, boulders, and small craters. Its computation is far more complex than the slope assessment, since roughness can be mistaken with noise in the input data [38]. Roughness features can be determined directly from the DEM or by employing the information available from the input image pair, such as texture and brightness. The different methods have been extensively analyzed during the literature research of the project, concluding that *deviation from the mean plane* (Equation 4.18) and *histogram-based variance* (Equation 4.19) are optimal for the framework of planetary HD. The equation are reported below for the sake of completeness

$$R = \text{abs}\,(Z_i - Z_{mean}) \tag{4.18}$$

where Z represents the DEM elevation and Z_{mean} the elevation from the mean plane

$$\sigma^2 = \frac{\sum_{i,j}\left(img(i,j) - \mu\right)^2}{N-1} \tag{4.19}$$

with img the input image while N the total number of pixels of the considered window.
Nevertheless, to solve the squared pattern problem (Section 4.2), rocks and boulders are identified at an early stage through the Harris Corner detector, losing the opportunity the assess their depth values. These pixels are then directly considered as hazardous in the mapping phase, which is discussed in next section. To avoid undetected roughness features such as small craters, the deviation from the mean plane was also implemented. Similarly to the slope computation, this approach leads to a map whose values needs to be re-scaled to the interval [0 1] according to the 30 cm constraint on roughness (Table 2.1).

4.4.5 Mapping

The final hazard map is obtained after combining shadow, slope, roughness and boulder information. Their merging is achieved by simply associating for each pixel the maximum value among the three maps, besides highlighting as hazardous pixels identified as depicting rocks and boulders. Therefore, the resulting hazard map has values increasing from 0 to 1 (unsafe). Hence, landing sites are not just grouped with the pure binary representation "safe/unsafe" but different values of safety are also provided (Figure 4.22). As matter of fact, not all the true positive landing sites are equal and such information needs to be taken into account for further landing site selection. This outcome is collected by the so-called *Piloting function*, which will create other information such as a reachability map, based on the required propellant and the guidance cost to reach each LS. The development of such function has been considered out of scope for this master thesis project and it is only mentioned for sake of completeness. As highlighted in Figure 4.2 the hazard map can be expressed either in the world coordinate frame or as projection in the image plane.

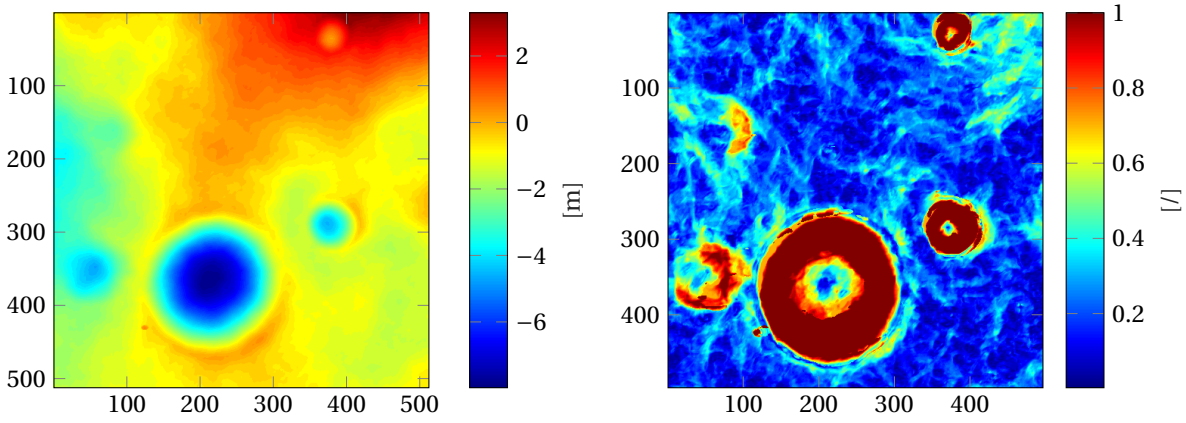

Figure 4.22: Ground-truth DEM (left) and its resulting hazard map (right)

Part III
Verification and Validation

Chapter 5

Verification

Verification and validation ($V\&V$) activity occurs for all the products we daily use. From the most simple to the most complex system, extensive $V\&V$ has indeed occurred to ensure they meet their intended purpose. Space systems are different in the sense that they are usually one-of-a-kind products [24]. Therefore $V\&V$, for both space software and space hardware components, must be extremely rigorous.

In this chapter, we turn our attention to the description of all the tests developed to prove the compliance of the software with its design solution specifications and descriptive documents [43]. Eight functional blocks were identified in Chapter 4. Thus, at least one unit test has to be performed for each block to assess it is providing the correct outputs from the given inputs. Nevertheless, it should be pointed out that also tests at a lower level (e.g. per function) must be performed. All tests carried out were successful and are summarized in Table 5.2. In Section 5.1, the results for some of the unit tests concerning the *DEM Computation* part are presented, while Section 5.2 focuses on tests related to the *Hazard Computation* part.

5.1 Unit Tests DEM Computation

Once running, the software opens a folder selection dialog box allowing to choose the mission scenario. Thereupon, it asks to select the altitude of the camera at 2nd pose, among those available. Based on this user-prompt command, it searches in the directory for the matching input images (e.g. alt_userValue_Imghigh.ppm for the image at 1st camera pose and $textalt$_userValue_Imglow.ppm for the image at 2nd camera pose) and ground-truths files (e.g. $textalt$_userValue_groundTruth.txt). Clearly, it is desired that the right files matching the user-prompt command are selected. For this reason, test INT-1 (Table 5.2) has been performed to assess that, for every altitude of the whole data-set, the right files are picked. The test outcome has proved that all the correct files matching the user-prompt command are uploaded.

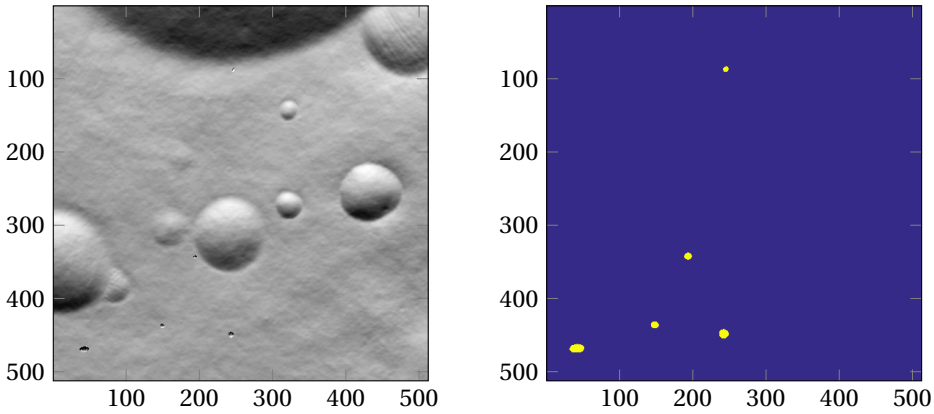

Figure 5.1: Input image at 600 m altitude (left) and its computed boolean rock map (right). Pixels depicting boulders and their immediate proximity are highlighted in yellow

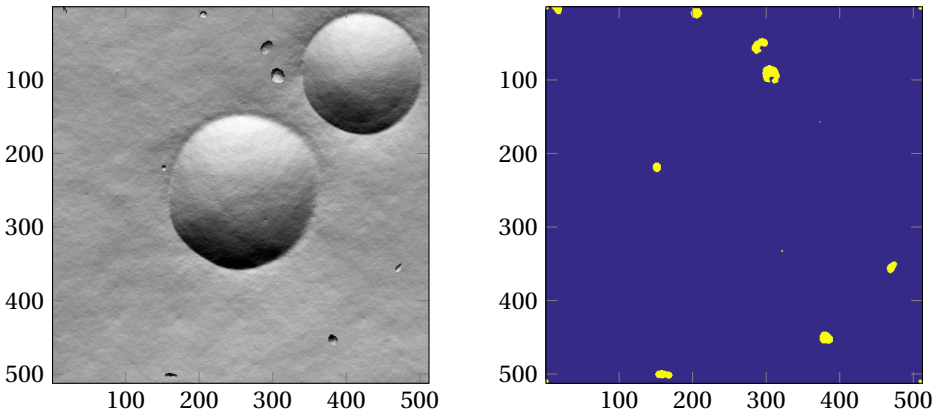

Figure 5.2: Input image at 1200 m altitude (left) and its computed boolean rock map (right). Pixels depicting boulders and their immediate proximity are highlighted in yellow

Test RDT-1 has been conducted to assess that rock detection, through a Harris Corner measure, is rightly accomplished. Several decent images have been considered, showing that the selected threshold for the corner measure allows to identify pixels depicting rocks, boulders, and their immediate proximity. Figures 5.1 and 5.2 shows an example.

Concerning reverse warping, Table 5.2 highlights that two different tests (tests RWT-1 and RWT-2) have been carried out. Test RWT-1 aims to verify the correctness of the warping process as a whole. As a matter of fact, inspecting Figure 5.3, already by visual inspection it can be seen that image I_2 is correctly warped back to the 1st image plane and made resolution equivalent and aligned to I_1.

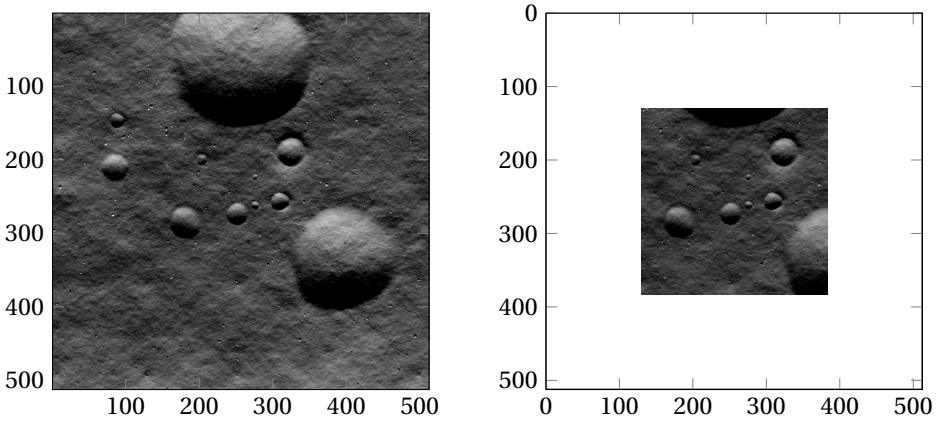

Figure 5.3: Input image I_1 (left) and a warped version of image I_2 computed for a specific z_k (right)

To further ensure that, test RWT-2 has verified that bilinear interpolation has been correctly implemented for the gray-scale intensity assessment, at non-integer image locations. For this purpose, bilinear interpolation has been compared with *interp2* at any image location (Figure 5.4). The latter is a *MATLAB* embedded function which allows interpolation for 2-D gridded data in meshgrid format. Due to its low computational speed, a better tailored function for the tight real-time constraints has been implemented, namely *imbinterp.m*. Pixelwise comparison has demonstrated that the images obtained are equal at ϵ-machine precision, therefore allowing to state that image warping is successfully accomplished. In test CT-1 (Table 5.2) to verify the implementation of the SSD and SAD operator 3×3 matrices, rather than actual images, were feeded to the function. The computations were proved manually to be correct.

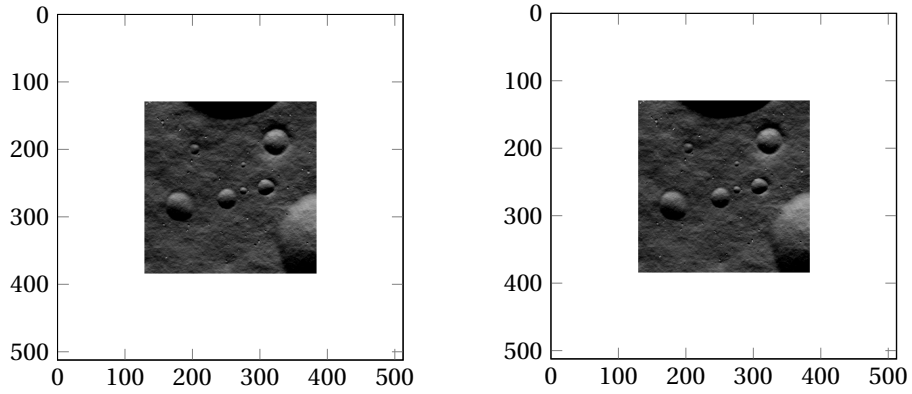

Figure 5.4: Warped versions of the input image I_2, computed with the tailored *imbinterp* (left) and with *MATLAB* built-in *interp2* function (right)

So far, all the blocks within the *DEM Computation* part have been proved to work properly, providing the correct outputs from the given inputs. Nevertheless, it is now mandatory to assess whether the integration of all these blocks allows to correctly compute DEMs. For this purpose, PANGU tool set comes to aid. As a matter of fact, PANGU also provides the ground-truth DEMs of the scenes being imaged: in other words, the 3D representation of the terrain free from errors and noise. It must be highlighted that the reconstructed DEMs are not expected to be equal to their ground-truths. Nevertheless, characteristics features of the scene (e.g craters), should be clearly identified. Moreover, the resulting errors should be within acceptable limits.

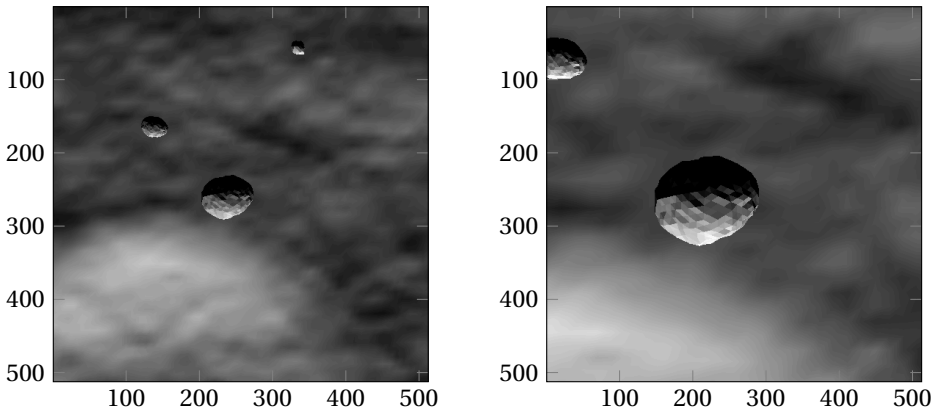

Figure 5.5: Image I_1 at 100m elevation (left) and I_2 (right) taken when the altitude is halved

For the following test a PANGU image-pair (Figure 5.5) has been generated for a camera set-up at 100 m and 50 m, respectively. The mean (μ) and the resulting median absolute deviation (MAD) are:

$$\mu = 0.3836\text{m} \quad MAD = 0.2971\text{m}$$

The mean is based on absolute errors while the MAD on the errors taken with their own signs, otherwise no meaningful conclusions could be drawn. These errors are clearly within acceptable limits. Figure 5.6 shows the resulting DEM and error map. As expected, the higher errors are located near the image center since depth recovery is numerically unstable in the vicinity of the epipole. The comparison with PANGU ground-truth identifies that the main features of the scene can be reckoned: the crater rim located on the bottom-left and the surface dip on the top-right. Therefore, it can be concluded that shape recovery has been successfully accomplished. Further discussion concerning the 3D reconstruction and its limits is extensively tackled in *Part IV Simulation and Results*.

In agreement to the software block diagram (Figure 4.2), the *Georeferencing* block is now considered before moving to the unit tests for the *Hazard Computation* part (Section 5.2). Once again, PANGU

ground-truths have been considered to verify this software block. Table 5.2 highlights that tests with both DEMs captured through a camera pointing along the nadir-direction (GRT-1) and with a line-of-site imaging sensor offset have been considered (GRT-2). To better understand the test assessments, let us consider the PANGU ground-truths. These represent a set of 3 matrices *X,Y,Z* identifying the coordinates of the planetary surface respect to the world reference frame (georeferenced DEM). An example of a georeferenced DEM was already shown in Figure 4.21. If only the ground-truth depth values *Z* are plotted, the result obtained are the elevation values shown as a projection into the image plane. Through georeferencing we want to assign to every pixel of the raster its x-,y-coordinate with respect to the world reference frame: in other words, we want to retrieve the *X,Y* matrices. Figure 5.7 shows an example of DEM employed within test GRT-2. Already by visual inspection, it can be concluded that the DEMs have been correctly georeferenced, with only a small difference.

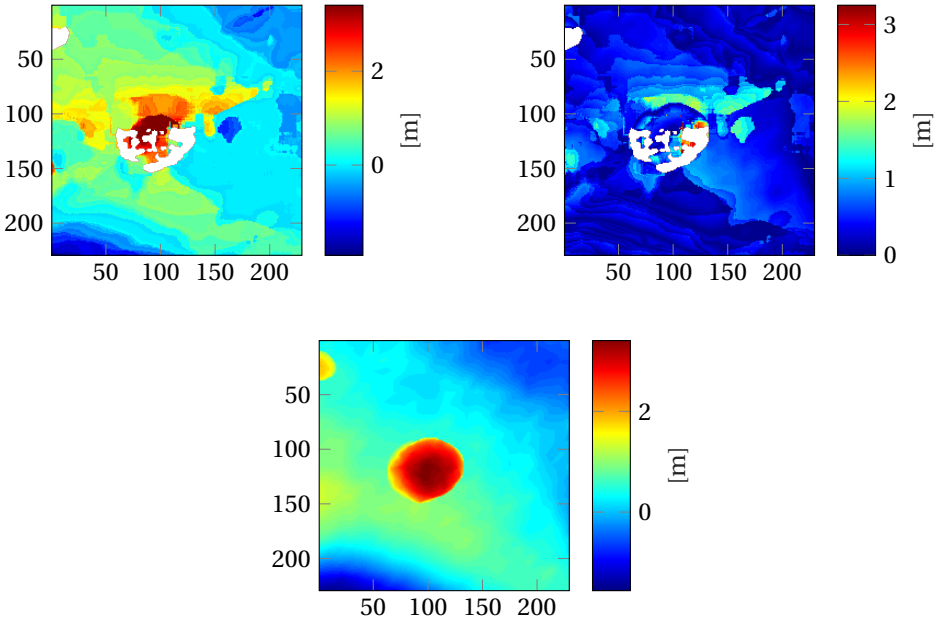

Figure 5.6: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 50m

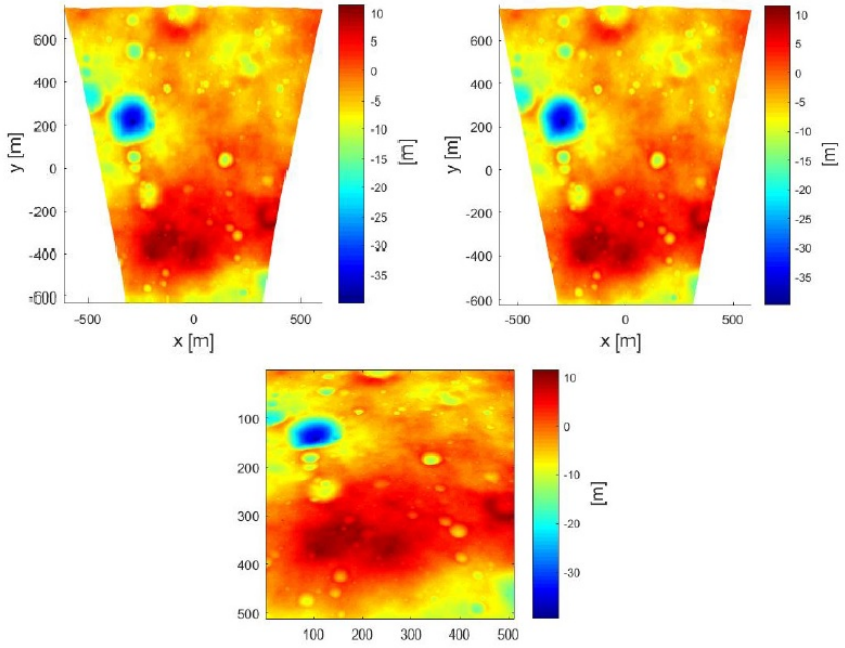

Figure 5.7: Test GRT-2. Ground-truth depth values as a projection into the image plane (left) Georeferenced DEM (center) Ground-truth DEM (right). Camera pointing along nadir direction, altitude h = 500 m

5.2 Unit Tests Hazard Computation

In this section, tests related to the *Hazard Computation* part are considered by starting with the assessment of the shadow mapping. Given PANGU input images, the computation of shadow boolean maps was already shown beforehand in Figure 4.17. Nevertheless, Table 5.2 illustrates that tests have also been performed independently of these images. For this purpose, in test SDT-1 chessboard images have been generated and fed as input to the *Shadow Detection* block (Test SDT-1). Figure 5.8 shows an example: as expected, the computed boolean shadow map displays pixels in shadow depicted as white, while pixels in light depicted as black. The following test has been performed for images with different number and size of the chessboard squares. All tests were successful, meaning also that the threshold selected is optimal and it can deal with different lighting conditions.

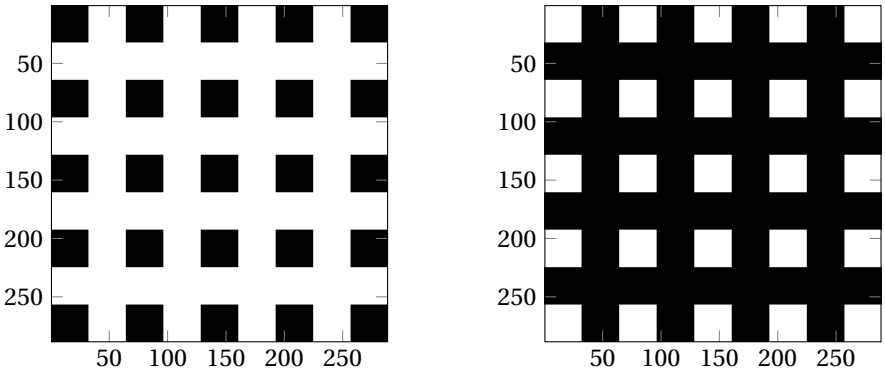

Figure 5.8: Input test image (left) and resulting output from the *Shadow Detection* block (right)

To test the *Slope Computation* block, both PANGU ground-truths (Test SCT-2) and DEMs generated by the author (Test SCT-1) have been considered. For the latter, Figure 5.9 shows an example, where the terrain elevation model resemble a "roof" turned upside down. Its key parameters are highlighted in the Table 5.1. Observing the figures, it can be clearly seen that both slopes have been correctly detected. A better inspection of the results reveals the presence of a 1 pixel-wide transition region when moving from left to right slope (Figure 5.10). This is within the accuracy envelope of the algorithm, therefore it can be concluded that the slope computation block is properly verified.

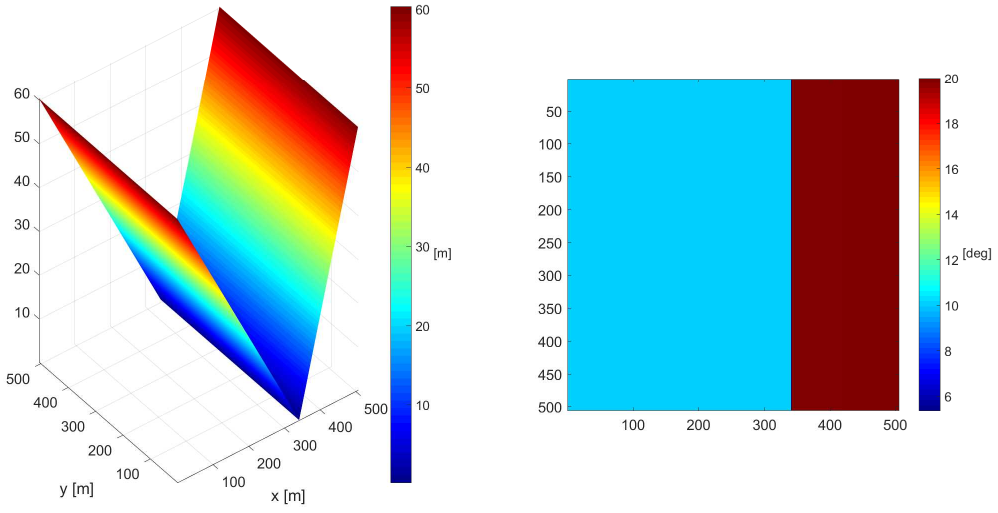

Figure 5.9: Input DEM for slope computation (left) and resulting output from the *Slope Computation* block (right). Slope computed through the linear regression method

Table 5.1: Input DEM parameters

$slope_{left}$	10°
$slope_{right}$	20°
resolution	1 m

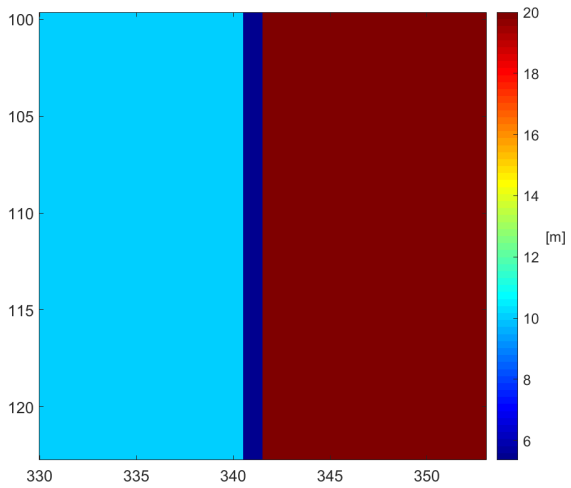

Figure 5.10: Zoom-in for the slope transition region

Through the previous tests, it has been assessed that shape recovery is successfully accomplished and that the functions to compute the surface hazards have been verified. Obviously, a global test is now needed to address the integration of all the software blocks: in other words, it is desired to ensure that the software, as a whole, correctly provides hazards maps that will be employed for landing site selection. For this purpose, in test HT-2, the hazard map from a computed DEM has been compared to the hazard map directly obtained from the PANGU DEM (Figure 5.11).

In agreement with the expectations, the result from the estimation does not match the true values of the corresponding ground-truth. Nevertheless, inspecting both maps it appears clear that main hazard such as the boulders, and the four main craters in the scene have been correctly identified as hazardous. Therefore, it is possible to state that the system has been successfully verified. Further discussion concerning the detection probabilities, and number of false alarms/undetected hazards will be extensively tackled in *Part IV Simulation and Results*.

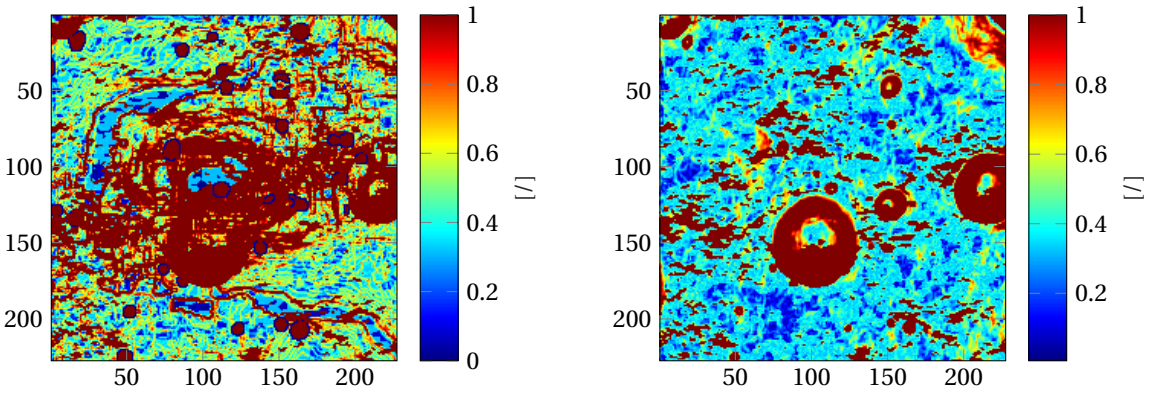

Figure 5.11: Computed scaled hazard map (left) and ground-truth scaled hazard map (right). Altitude h = 500 m

All the tests beforehand discussed have been summarized in Table 5.2. It can be clearly seen that further tests have also been executed, to prove the software compliance. At least one test for each software block has been developed, nevertheless additional tests were conducted where necessary. It is further stressed that all these tests were successful.

Table 5.2: Software Tests Summary

Test Name	Function	Function Description	Block/Part	Test Input
INT-1	getFile	The function looks up for files matching a user-prompt input, returning the descent adjacent images, the respective altitude of the camera poses, and the LS ground truth depth values	Input	• Altitude camera 2nd pose
RDT-1	RockDetection	Recognition of rocks, boulders, and their immediate nearby pixels through Harris Corner measure. Computation of related boolean map	Rock Detection	• PANGU images
RWT-1	reverseWarping	It warps the image taken at camera 2nd pose back to the first image plane	Image Warping	• PANGU image pairs
RWT-2	imbinterp	Bi-linear interpolation employing the 4 nearest pixels around the target location	Image Warping	• PANGU images 2nd pose
CT-1	corr_operator	Implemention of SSD and SAD	Correlation	• 3×3 matrix of intensity values
GRT-1	georeferencing	Assign real-world coordinates to each pixel of the raster	Georeferencing	• Ground-truth DEMs (no tilted camera)
GRT-2	georeferencing	Assign real-world coordinates to each pixel of the raster	Georeferencing	• Ground-truth DEMs (tilted camera)
SDT-1	imshadow	Computation of a shadow boolean map	Shadow Detection	• chessboard images
SDT-2	imshadow	Computation of a shadow boolean map	Shadow Detection	• PANGU images
SCT-1	slopeDEM	Computation slope map through linear regression	Slope Computation	• "Roof"-like DEMs
SCT-2	slopeDEM	Computation slope map through linear regression	Slope Computation	• Ground-truth DEMs
RT-1	roughnessDEM	Computation of a roughness map as deviation from mean plane	Roughness Computation	• artificial map
HT-1	hazardDEM	Computation of the hazardousness of the landing region	Mapping	• Artificial maps
HT-2	hazardDEM	Computation of the hazardousness of the landing region	Mapping	• PANGU image pairs • ground-truth DEM

Part IV
Simulation and Results

Chapter 6

Performance Assessment

At this point, the simulation software is complete. Its performance and accuracy are hereafter objectively quantified. For this purpose, two different mission scenarios have been identified: 1) a perfect vertical motion forward along the camera pointing direction (Section 6.1) 2) a 45° angle dropping trajectory to simulate a realistic approaching descent phase (Section 6.2).

For the former scenario, the set of images is presented in Appendix F. As the camera height decreases from 1000 m to 50 m above the ground, it can be noticed that the nominal landing site is dwelled by a big boulder, therefore re-targeting is necessary. The scenes at high altitude contain multiple craters and boulders on a rather sloped and complex terrain, with a maximum elevation difference of approximately 50 m. The latter scenario simulates Chang'e-3 powered flight (Section 2.1) and the descent sequence (Appendix I) is captured starting from 2400 m above the ground. During this phase the pitch angle is set to 9°, while both the yaw and roll angle are kept at 0°, and the lander approaches the nominal site through a 45° angle dropping trajectory. The sensor line-of-site is intentionally offset to 40° to ensure the camera is free of the effect of main engine plume and its 30° FOV is directed towards the landing area. For the input image pairs at 300-150 m and 250-125 m, instead, the attitude of the lander is aligned with the normal direction of the surface, preparing for the vertical approach phase. Similarly to the previous scenario, the different scenes are representative of the lunar surface with multiple craters and boulders.

Finally, in Section 6.3, a sensitivity analysis is carried out to determine the limitations of the developed algorithm under ordinary operative conditions. For this purpose, errors in the knowledge of the lander's position and orientation are introduced based on the accuracy that can be achieved through traditional onboard sensors (e.g. altimeter, IMU) during an actual landing.

6.1 Mission Scenario 1

As the lander is steered towards the surface, in its vertical descending motion, the scenes captured have different hazard-levels (Table 6.1). As a matter of fact, the number of safe landing locations progressively decreases within the 400-100 m altitude range, with more than 55% of the scene imaged at 100 m altitude labelled as hazardous.

Table 6.1: Reference Mission Parameters

Altitude [m]	50	75	100	150	200	300	400	500
Hazards in scene [%]	37.22	53.91	55.88	46.29	40.61	26.73	20.62	21.34
Size DEM patch [m×m]	24×24	36×36	48×48	71×71	95×95	143×143	190×190	238×238

The full set of DEM, slope (*slp* subscript), and roughness (*rgh* subscript) errors are presented in Table 6.2. Given that rocks and boulders are directly detected from the input images through a Harris Corner measure, it must be highlighted that the roughness errors are not really meaningful. As carefully explained beforehand (Section 4.2), the pixels depicting rocks and boulders are indeed omitted from the shape recovery quality assessment, being directly identified as hazardous. Therefore, roughness errors are here only representative of secondary features which are not detected by the rock detection block (e.g small craters). For this reason, the attention is focused foremost on the slope and DEM that, on the contrary, are quantitatively assessed.

Table 6.2: Reference mission: DEM, slope, and roughness errors

Altitude [m]	μ_{DEM} [m]	median_{DEM} [m]	MAD_{DEM} [m]	μ_{slp} [deg]	median_{slp} [deg]	MAD_{slp} [deg]	μ_{rgh} [m]	median_{rgh} [m]	MAD_{rgh} [m]
50	0.38364	-0.025445	0.29711	6.3561	2.6167	4.4023	0.13029	0.068651	0.067006
75	0.59525	-0.072331	0.53531	8.0967	2.622	5.4328	0.14939	0.078775	0.075244
100	0.77523	0.027263	0.67226	8.3897	3.2301	4.9601	0.14298	0.079738	0.066827
150	1.2054	0.1133	1.0004	8.5611	2.9505	5.0463	0.12764	0.068526	0.060017
200	1.6117	0.322	1.3765	8.5738	3.2806	5.1678	0.1223	0.067949	0.059155
300	2.3598	0.57111	1.9841	8.333	2.5906	4.9204	0.13295	0.061375	0.058871
400	3.103	0.54828	2.5299	8.1927	2.6774	4.6725	0.10276	0.055558	0.051066
500	3.8769	0.52091	3.1105	7.8275	2.2812	4.2974	0.11622	0.057676	0.052951

Similarly to the reconstruction quality assessments of Sections 4.3 and 5.1, the mean value μ (based on absolute errors) and median (based on the errors taken with their own sign) have been employed. Likewise, the median absolute deviation (MAD) has been considered to assess the dispersion of the data values. The choice of these statistical parameters is further strengthened when inspecting that the DEM error distributions are clearly not normal (Figure 6.1): indeed, both skewness and kurtosis are always different from zero for the whole descent sequence (Table 6.3). This holds true for slope errors as well (Figure 6.2).

For decreasing altitudes, the DEM mean errors μ_{DEM} appear to lower steadily: from approximately 3.87 m at 500 m (red cell in Table 6.2) to only ~0.38 m at 50 m above the surface (blue cell in Table 6.2). Nevertheless, the reader's attention should be drawn to the median absolute deviation MAD_{DEM}, which appear to match the respective μ_{DEM}: this highlights that the errors fluctuate a lot, being highly dispersed. As a consequence, the relevance of the information provided only by the mean parameter is not fully reliable. Similarly for the slope errors: despite the mean μ_{slp} is on average half of the value that would indicate a hazard (<15°), the median absolute deviation MAD_{slp} is 4 to 5 meters!

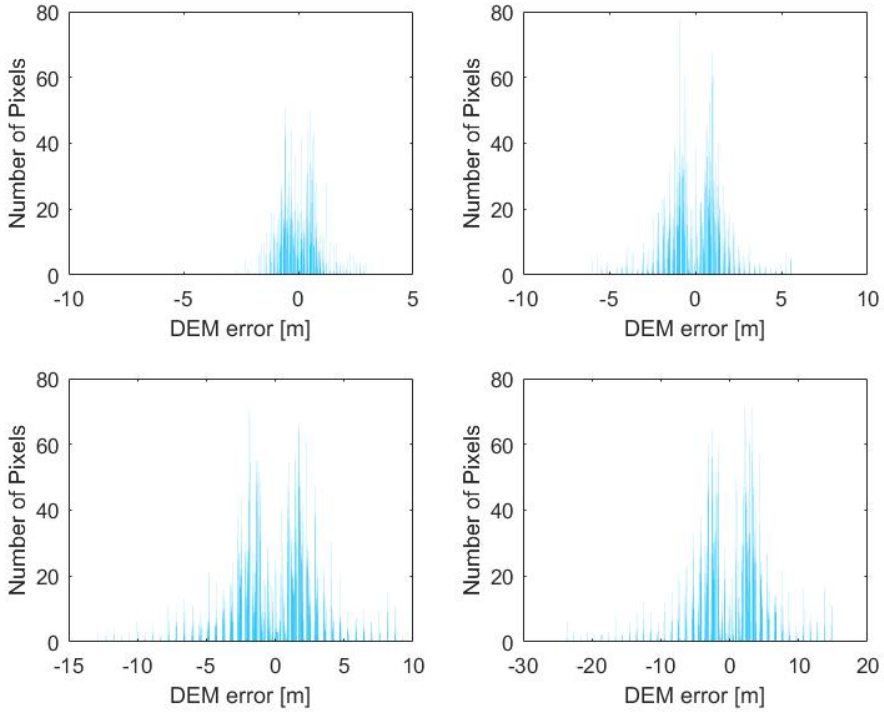

Figure 6.1: DEM error distributions. Results for altitude h = 75 m (top-left), h = 150 m (top-right), h = 300 m (bottom-left), h = 500 m (bottom-right)

Table 6.3: Reference mission: DEM error skewness and kurtosis

Altitude [m]	**Skewness**	**Kurtosis**
50	0.29711	-2.9691
75	0.44251	-2.5575
100	0.066929	-2.9331
150	0.012546	-2.9875
200	-0.13797	-3.138
300	-0.23239	-3.2324
400	-0.36041	-3.3604
500	-0.42495	-3.4249

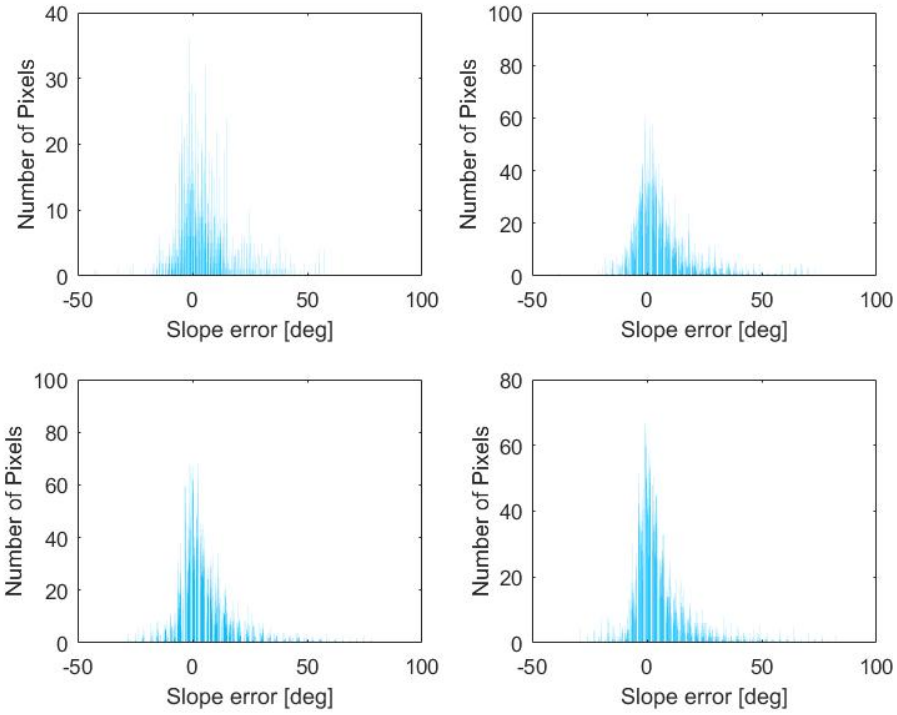

Figure 6.2: Slope error distributions. Results for altitude h = 75 m (top-left), h = 150 m (top-right), h = 300 m (bottom-left), h = 500 m (bottom-right)

These large errors are clearly linked to the quality of the reconstructed terrain maps. Appendix G shows the computed DEMs together with the related error maps and ground-truths. It can be seen that all the obtained DEMs are highly noisy, with the elevation either under-estimated or over-estimated also outside the epipole region, where shape recovery is numerically unstable. While the crater rims appear reconstructed rather well, the depth in their immediate inner region can be under-estimated by almost 7 m. Furthermore, the rather flat regions at the top-left and bottom-right of the maps appear to be affected by a bias error of 2 to 3 m. Investigating Figures 6.3 and 6.7 it can be seen that such DEM errors are propagated to the slope, with the error map almost matching the actual computed map. Nevertheless, no new errors are introduced. Due to the jagged shape recovery, slope and roughness are overestimated. It should be pointed out, however, that overestimation is in general preferable to underestimation, as it can be seen as a safety factor.

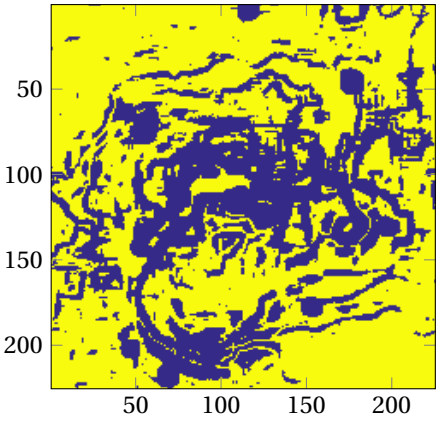

Figure 6.3: Slope Boolean Error Map. Slope errors below half of the slope value that would indicate a hazard (15°) are shown in yellow, altitude h = 300 m

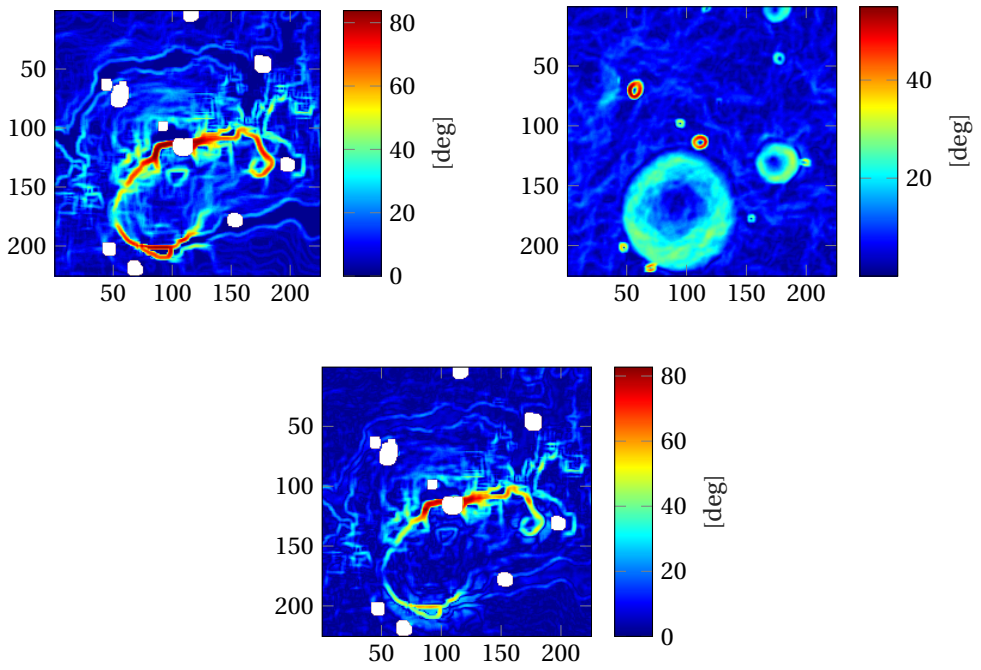

Figure 6.4: Slope computed DEM (top-left) slope ground-truth (top-right) slope error (bottom). Altitude h = 300 m

Xiong [43] only discloses the root-mean-square (RMS) DEM errors for two maps at 6 m and 12 m respectively, obtained from a set of synthetic images. Furthermore, when true descent images are employed, with the height of the camera decreasing from 1085 m to 8 m above the ground, no quantitative information is provided. To assess whether the computed DEMs can be employed for HD, the final hazard maps (Appendix H) obtained after combining slope, roughness, and shadow information are now considered. Here, the red regions are those that are hazardous based on the safety constraints beforehand specified (slope $\geq 15°$ and roughness ≥ 0.3 m). Appendix H also includes the related hazard-error maps indicating the location of the undetected hazards (dark-gray regions) and false alarms (light-gray regions). Clearly, the undetected hazards (FP) are those we want to minimize. Taking these into account, the detection probabilities FP, FN, TP, TN are presented in Table 6.4.

In contrast to what is expected, it can be observed that the FP percentage grows for decreasing altitudes. Further analysis reveals how such behaviour is primarily linked to the presence of a big boulder at the image center. As a matter of fact, the image resolution increases as the lander is gradually steered towards the surface. Therefore, the intensity difference from pixel to pixel depicting the boulder becomes gradually smaller. As a consequence, the Harris corner measure is not successful in identifying the whole boulder (Figure 6.5). Clearly, the issue does not arise at higher altitudes allowing to achieve better results (1.36% at 500 m above the surface). Nevertheless, the percentage of the undetected hazards is never below the allowable maximum 1%. Furthermore, inspecting the hazard maps, it can be seen that all FP errors are always adjacent to a detected hazard or even surrounded by detected hazards.

Table 6.4: Reference mission: detection probabilities

Altitude [m]	**FP [%]**	**FN [%]**	**TP [%]**	**TN [%]**
50	7.27	11.70	51.08	29.95
75	4.73	8.96	37.14	49.17
100	4.71	11.79	32.33	51.17
150	4.77	14.85	38.86	41.52
200	3.82	16.19	43.20	36.79
300	1.87	17.23	56.04	24.86
400	1.44	17.18	62.20	19.18
500	1.36	15.60	63.06	19.98

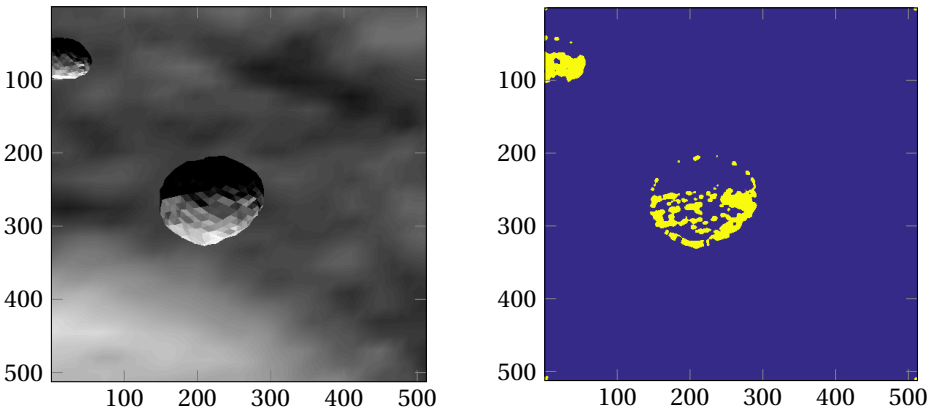

Figure 6.5: Input image at 50 m altitude (left) and its computed boolean rock map (right). Pixels depicting boulders and their immediate proximity are highlighted in yellow

Since the FP index is a measure on how many pixels represent undetected hazards out of all pixels in the scene, the percentage of hazards correctly detected should also be analyzed. The results are presented in Figure 6.6: from the graph it can be concluded that more than 90% of all hazards are correctly identified above 75 m altitude. The graph also shows how many safe LS out of all safe LS are correctly identified (TP%) and the percentage of total correct detection (CDI). Overall, TP% is always smaller than TN% index due to the large number of false alarms, linked to the overestimation of the slope.

For 50 m altitude a major drop in the TN% and TP% values can be observed, which is inconsistent with the pattern described by the data previously gathered. This is primarily linked to the failure of the rock-detection block in identifying the whole boulder at the image center. As a matter of fact, when the rock is excluded from the performance assessment through a 80×90 px^2 area, Table 6.5 shows that the detection probabilities and indices improve in agreement with the trend marked by the higher altitude values. The result clearly suggests that this data-set might not be optimal for the hazard performances assessment of the algorithm: the big boulder, located at the image center, together with the crater below appear to negatively influence the results below 200 altitude.

Concerning the implementation aspects, the rectangular region has been simply neglected after the DEM was computed. Nevertheless, to assess whether this area could potentially have a negative impact on the surrounding pixels, its exclusion before the image correlation was also considered. Table 6.6 highlights that the two approaches lead to the same results up to negligible differences. Therefore, for the sake of simplicity, the first approach was chosen for further investigations.

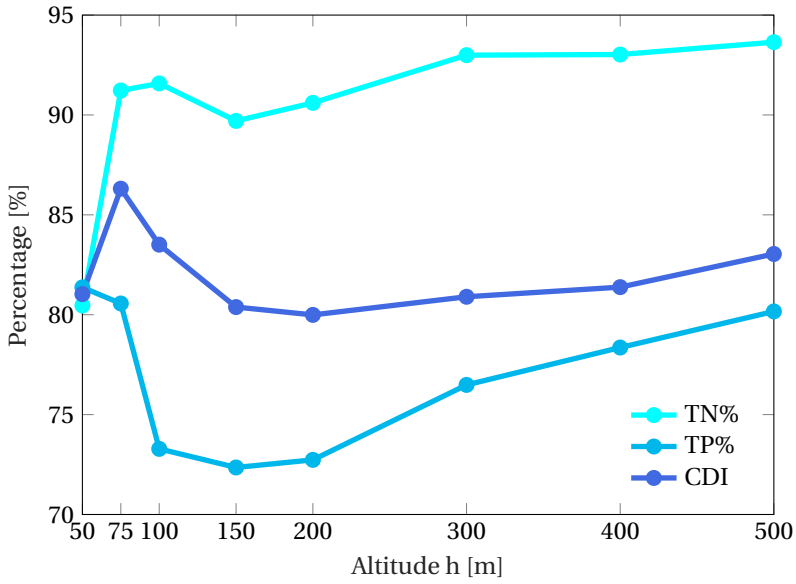

Figure 6.6: Correct detection TN%, TP%, CDI vs Altitude

Table 6.5: Approach 1: detection probabilities and indices excluding boulder at image center, altitude 50 m

FP [%]	**FN [%]**	**TP [%]**	**TN [%]**	**TP% [%]**	**TN% [%]**	**CDI [%]**
3.30	9.76	62.15	24.79	88.24	86.43	86.94

Table 6.6: Approach 2: detection probabilities and indices excluding boulder at image center, altitude 50 m

FP [%]	**FN [%]**	**TP [%]**	**TN [%]**	**TP% [%]**	**TN% [%]**	**CDI [%]**
3.30	10.03	61.87	24.79	88.24	86.05	86.67

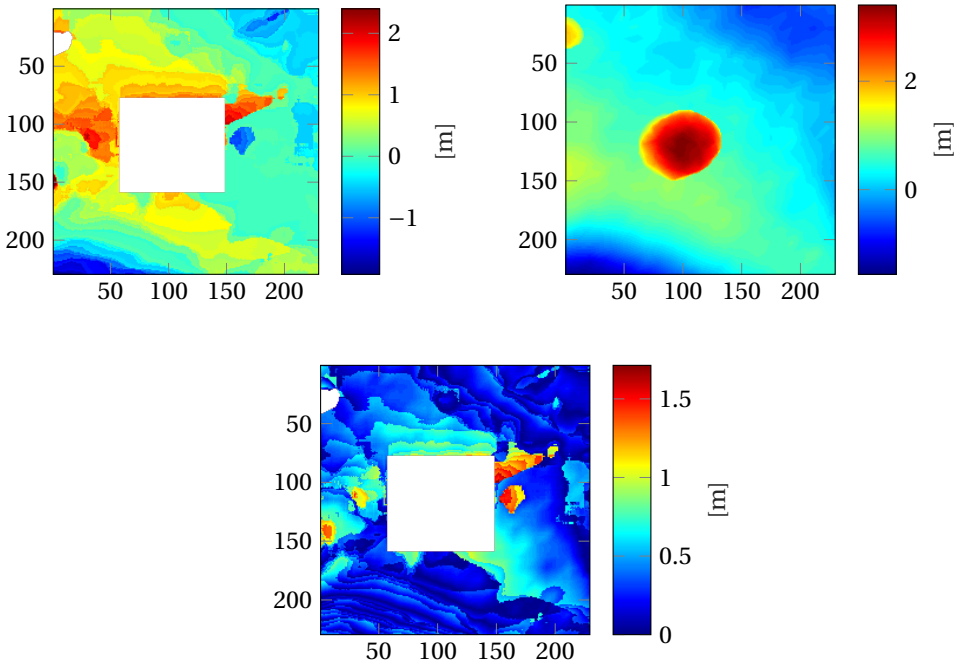

Figure 6.7: Shape Recovery excluding boulder. Computed DEM (top-left) ground-truth (top-right) and DEM error (bottom). Altitude h = 50 m

Clearly, the previous analysis raises questions whether removing completely the epipole region leads to significant improvements in the hazard assessment for higher altitudes. The unstable recovery due to the presence of the epipole can be enclosed approximately within a 80×80 px^2 squared area around the center. Table 6.7 highlights the results obtained for altitudes ≥200 m, where the boulder and the crater below seem to not affect the results. The comparison with the data of Table 6.4 shows a decrease in the percentage of false alarms, with a related growth of TP% and CDI indices. False alarms are indeed mainly located around the nominal landing site. Nevertheless, the percentage of the undetected hazards is still not below the allowable maximum 1%.

Whether removing the epipole region or not, it appears undeniable that the locations around the image center can not be correctly assessed. This represents a significant problem since these locations are the most convenient in terms of distance and guidance costs. As these sites are wrongly classified as unsafe, they result unaccessible. Therefore, it can be concluded that the overall quality of the recovered depth maps, obtained by correlating vertical image pairs, does not appear adequate enough for landing site

selection.

Table 6.7: Detection probabilities and indices excluding epipole region, altitude 300 m

Altitude [*m*]	**FP [%]**	**FN [%]**	**TP [%]**	**TN [%]**	**TP% [%]**	**TN% [%]**	**CDI [%]**
200	3.49	12.62	45.80	30.09	78.399	91.06	83.89
300	1.46	12.55	61.46	24.52	83.04	94.37	85.98
400	1.07	13.07	67.84	18.02	83.85	94.39	85.86
500	1.33	11.82	68.56	18.29	85.29	93.23	86.85

6.2 Mission Scenario 2

For the new descent sequence (Appendix I) the computed depth maps are presented in Appendix J. Figure 6.8 highlights the shape recovery results at 500 m above the surface. Two features immediately catch the eye: the shape of the terrain patch and the location of the epipole. For the latter, the coupling of selected descent trajectory and line-of-site offset ensure that the epipole is not anymore located at the image center, thus allowing a stable shape recovery around the nominal LS. During the approaching phase, when the nominal pitch angle is set to 9°, the epipole is always located below the image center. Instead, it moves upward when the lander attitude is aligned with the normal direction of the surface.

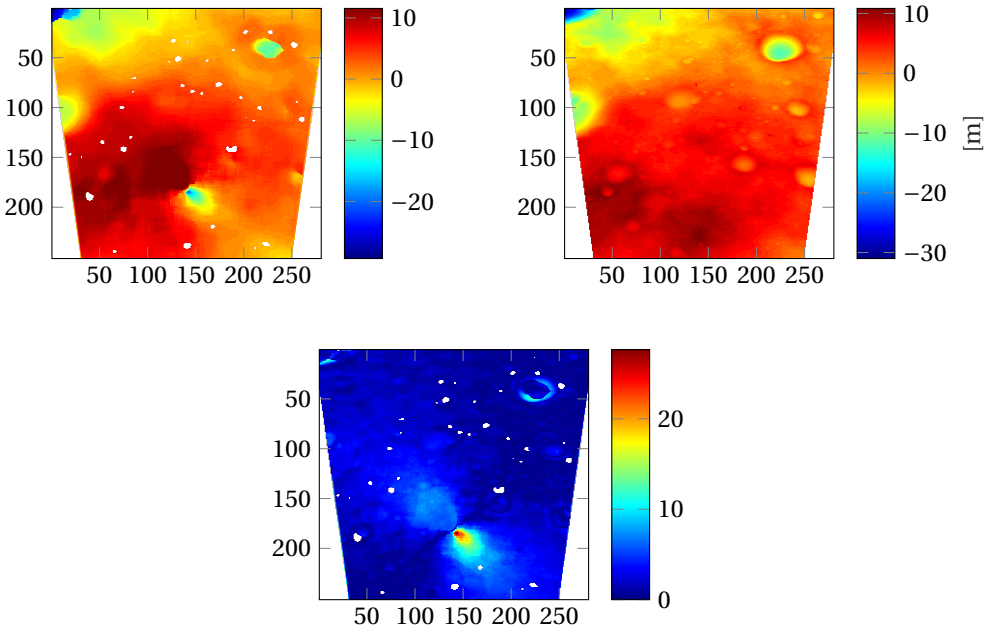

Figure 6.8: Computed DEM (top-left) ground-truth (top-right) and DEM error (bottom). Altitude h = 500 m

Concerning the shape of the terrain patch, it does not resemble a square anymore but rather a trapezoid. The reason lies behind the warping, back to the first image plane, of scenes captured with a tilted camera. For an intuitive explanation, let us now consider Figure 6.9 which outlines the current mission scenario. Points C_1 and C_2 represent the camera centers at 1st and 2nd pose, respectively. The trapezoids T_1 (ABCD), T_2 (EFGH) depict the surface caught by the squared imaging sensor at each pose. It is further stressed that the images reflect the sensor size, therefore they are squares of 512×512 pixels. Nevertheless, the terrain imaged resembles a trapezoid since a larger portion of the surface is captured while moving from the lower to the upper border of the image, due to the tilted camera axis. It can be easily proved

that $\overline{AB}$ is double the length of $\overline{EF}$. Similarly, this holds true for sides $\overline{CD}$ and $\overline{GH}$. Therefore, it would appear reasonable that the image at the lower altitude is warped to a square of 256×256 pixels, in order to match the resolution of the upper image. Nevertheless, it should be noticed that the lateral resolution is not constant throughout the whole image, as for the vertical image pairs, but rather it increases as moving from the lower to the upper image border. Let us now consider the sides $\overline{E'F'}$, $\overline{G'H'}$ obtained by extension of $\overline{EF}$,$\overline{GH}$ until touching the perimeter of the trapezoid T_1. Side $\overline{E'F'}$ is more than double the length of $\overline{EF}$. Therefore, when warped back to the first image plane, side $\overline{EF}$ is imaged into less than 256 px. Vice versa, the length of side $\overline{G'H'}$ is less than double the length of $\overline{GH}$. Thus, it is warped to more than 256 px to match the resolution of the image taken at 1st camera pose.

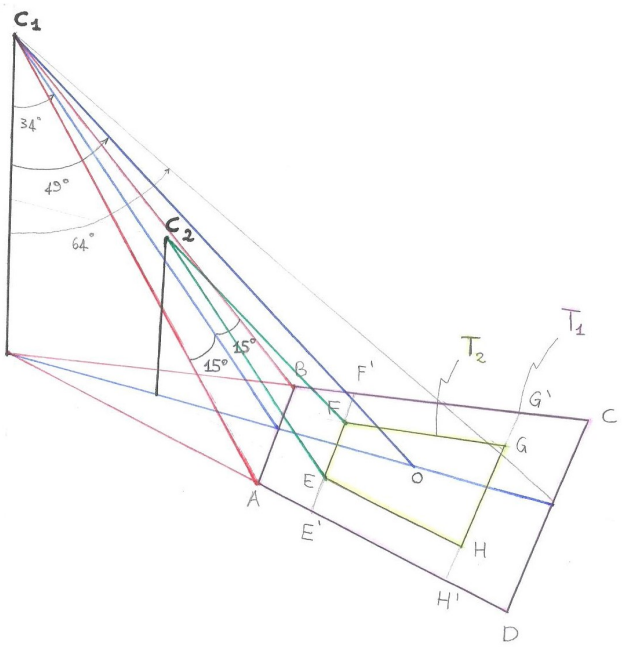

Figure 6.9: Schematic representation of the reference scenario

In contrast to the previous reference mission, the current descent sequence includes a lower number of hazards: Table 6.8 highlights how the hazard percentage decreases from ~30% at 1200 m altitude to only ~4% at 125 m, indicating that the designated landing area is rather safe. The full set of DEM, slope (*slp* subscript), and roughness (*rgh* subscript) errors are presented in Table 6.9. Similarly to the previous scenario, the MAD rather than the standard deviation has been considered to better described the dispersion of a not gaussian data set (Table 6.10).

Table 6.8: Reference Mission Parameters

Altitude [m]	125	150	250	300	500	600	1000	1200
Hazards in scene [%]	4.20	13.05	13.06	14.17	18.73	22.71	27.58	29.87

In agreement with the expectations, the DEM mean errors μ_{DEM} appear to lower steadily for decreasing altitudes: from approximately 4.29 m at 1200 m (red cell in Table 6.9) to only ~0.72 m at 125 m above the surface (azure cell in Table 6.9). The comparison of the former results against those outlined in Table 6.9 allows to conclude that a better shape recovery is achieved for the coupling of selected descent trajectory and line-of-site offsets. For instance, the DEM mean error μ_{DEM} and the median absolute deviation MAD_{DEM} at 500 m altitude, for the camera pointing upright, are greater than the respective errors at 1000-1200 m obtained with a tilted camera (orange cell in Tables 6.2 and 6.9). Furthermore, the slope mean error μ_{slp} and the median absolute deviation MAD_{slp} for the tilted camera are always half of the values achieved at lower altitudes with the camera pointing upright. The overall improved reconstruction is linked to the reduction of the epipole region, visible when inspecting the computed DEMs (Appendix J). The DEMs also reveal that the rock-detection block is able to identify hazardous small craters and not just rocks and boulders.

Table 6.9: Reference mission: DEM, slope, and roughness errors

Altitude [m]	μ_{DEM} [m]	median_{DEM} [m]	MAD_{DEM} [m]	μ_{slp} [deg]	median_{slp} [deg]	MAD_{slp} [deg]	μ_{rgh} [m]	median_{rgh} [m]	MAD_{rgh} [m]
125	0.72354	-0.23938	0.49552	4.1601	1.236	2.5799	0.058964	0.043056	0.039252
150	0.83139	-0.20839	0.65487	4.2042	0.41483	2.7537	0.061349	0.041466	0.041734
250	1.0979	0.34943	0.61064	4.3322	-0.10187	2.7768	0.058029	0.034208	0.039659
300	1.2463	0.38489	0.71013	4.0794	-0.47703	2.5652	0.06089	0.031729	0.041647
500	2.047	0.63435	1.134	4.2334	-0.70081	2.4075	0.083768	0.023364	0.052593
600	2.4032	0.75001	1.3972	4.596	-1.0041	2.546	0.1007	0.016546	0.062304
1000	3.7599	0.87281	2.2703	4.5371	-0.86167	2.5187	0.17218	0.017973	0.098402
1200	4.2886	0.86947	2.6501	4.5248	-0.72401	2.4977	0.2099	0.019997	0.11888

Table 6.10: Reference mission: DEM error skewness and kurtosis

Altitude [m]	Skewness	Kurtosis
125	-0.29254	-3.2925
150	-0.826	-3.826
250	-1.6174	-4.6174
300	-1.7908	-4.7908
500	-1.6319	-4.6319
600	-1.8701	-4.8701
1000	-1.9732	-4.9732
1200	-2.1156	-5.1156

Further inspection of the obtained depth maps reveals that the algorithm is not able to correctly resolve crater rims when the camera axis is not aligned with the normal direction of the surface. This is well highlighted in Figure 6.8 where it can be seen that the true rim of the crater located top-left is completely missing and it is instead identified as a rather flat region. Clearly, such errors are propagated to both slope and roughness (Figure 6.10 and 6.11) nevertheless, no additional error is introduced.

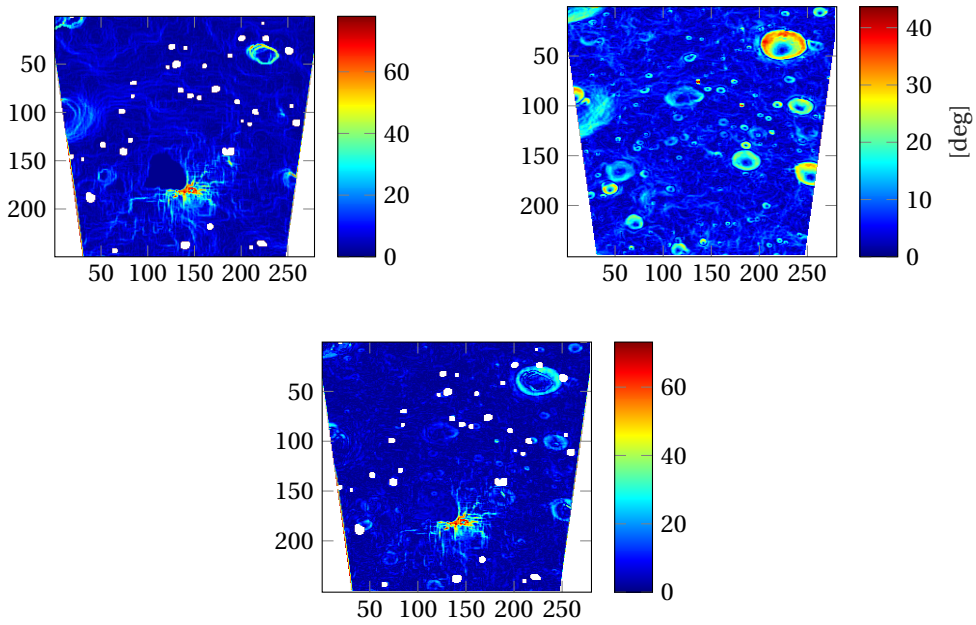

Figure 6.10: Slope computed DEM (top-left) slope ground-truth (top-right) and slope error (bottom). Altitude h = 500 m

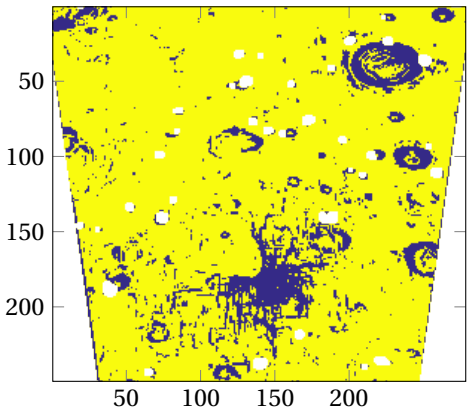

Figure 6.11: Slope Boolean Error Map. Slope errors below half of the slope value that would indicate a hazard (15°) are shown in yellow, altitude h = 500 m

To understand whether the overall quality of the computed DEMs is adequate enough for landing site selection, the obtained detection probabilities are now considered (Table 6.12). It should be noticed that the FP percentage is below the allowable maximum 1% at both 125 m and 150 m altitude. Moreover, for the maps obtained at 250 m and 300 m altitude, such limit is approached with a value of approximately 1.3%. Investigating the final hazard maps (Appendix H), similar conclusions to the previous scenario can be drawn regarding the location of the undetected hazard. This raises questions whether the lower FP percentages are due to the robustness of the algorithm or the fact that the scenes are inherently safe (Table 6.8). Unfortunately, tight timing and lack of additional data set did not allow further investigation.

The TN%, TP%, and CDI indices are presented in Figure 6.12. From the graph it can be seen that the number of hazards correctly detected increases as the lander approaches lower altitudes: precisely, the algorithm detects more than 90% of all hazard correctly up to 300 m, with total correct detection on average above 94%. This behaviour is in agreement with the expectations since, as the lander is steered towards the surface, higher resolution images are available. The result further support the idea that features in the data-set, employed for the vertical motion analysis, might have negatively affected the results for low altitudes.

Table 6.11: Reference mission: detection probabilities

Altitude [m]	**FP [%]**	**FN [%]**	**TP [%]**	**TN [%]**
125	0.10	6.24	89.56	4.10
150	0.75	5.11	81.84	12.30
250	1.32	5.34	81.60	11.74
300	1.34	4.48	81.35	12.83
500	2.26	4.62	76.65	16.47
600	3.34	4.83	72.47	19.36
1000	3.39	11.22	61.20	24.19
1200	5.67	14.46	55.67	24.20

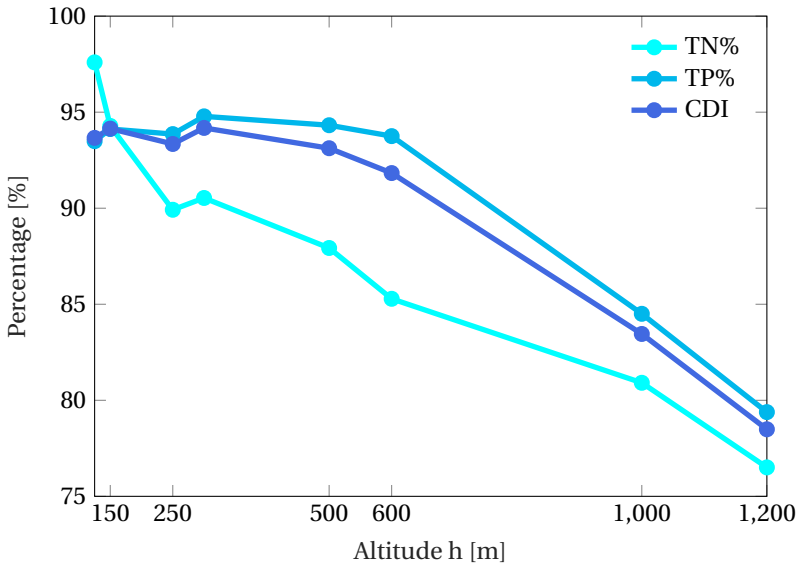

Figure 6.12: Correct detection TN%, TP%, CDI vs Altitude

Overall, the considered image descent sequence indicates that below 300 m the algorithm is a suitable candidate for HD to increase the autonomy and decrease the risk of a landing failure for future exploration missions. Furthermore, it has been proved that a solution to improve depth recovery, in the vicinity of the epipole, consists of selecting an *ad hoc* coupling of descent trajectory and line-of-site offsets to move the epipole away from the image center or potentially outside the image boundaries. For the sake of completeness, the full data-sets (slope and roughness from DEM) of these feasible scenarios are shown in Appendix L

6.3 Sensitivity Analysis

In the previous section, the experiments have resulted in depth maps that can be employed for HD, to further increase the autonomy and decrease the risk of a landing failure. Nevertheless, a perfect knowledge of the spacecraft's state has so far always been assumed. The dearth of navigation aids (e.g. GPS or radio-beacons) forces EDL systems to rely on onboard sensors for position and attitude estimation. Traditional sensors include, for instance, altimeter and IMU [29]. Given an estimate of the initial position and velocity, IMUs simply integrate acceleration measurements thus being in theory fully self-sufficient with no knowledge about the surroundings necessary. [14] Nevertheless, these are subject to unbounded error accumulation due to the associated integration of noise and errors (e.g. bias, axis misalignment, angle random walk to name but a few).

It is not the purpose of this MSc thesis to further dive into the description of these sensors nevertheless, it turns out essential to assess whether they provide knowledge of the lander's location accurate enough to perform HD with the developed algorithm. During an actual landing, a 1 m and 2° level of accuracy can be achieved for the spacecraft's altitude and orientation, respectively. [28] [43] Therefore, assuming to employ an inertial navigation system for position and attitude estimation, these levels of accuracy are now considered.

6.3.1 Errors in Camera Position

To assess the algorithm robustness to errors in camera position, a Monte Carlo simulation was performed. Thereupon, random uncertainties Δh_1, Δh_2, within the interval [-0.5 0.5] meters were taken into account for the altitude of each camera pose up to 20 samples. With reference to the previous analysis on shape recovery, it was concluded that a +15% increase of the DEM errors represents the maximum allowable to further employ the reconstructed maps for landing site selection. As matter of fact, higher deviations would lead to depth maps that are not adequate enough for hazard detection. In Tables 6.12, 6.13, 6.14, and 6.15 the DEM mean, median, and MAD errors obtained for the different uncertainties are presented together with their percentage deviation, relative to the errors computed for a perfect knowledge of the spacecraft's state at each camera pose. It can be clearly marked that for all the altitudes considered, the DEM errors are bounded to the maximum allowable 15% only for small deviations (approximately $\pm$10 cm), peaking to 250%-300% for the other values belonging to the specified interval.

To gain more insights on the role played by these uncertainties, Figures 6.13, 6.14, 6.15, and 6.16 illustrates the DEM mean relative percentage errors as function of the deviations on altitude ($\Delta h_1, \Delta h_2$) at both camera poses. As expected, the lower error percentages appear to be arranged along the right-to-left diagonal. As matter of fact, when both altitudes are affected by errors of similar magnitude and same sign ($\pm$) the ratio h_2/h_1 is really close to 1/2, which has been previously proved necessary to achieve dense reconstruction near the image center (Section 3.3). Furthermore, the admissible deviations along the diagonal increase gradually with altitude. This is linked to the fact that the weight of a deviation decreases for progressively increasing altitudes (e.g. a deviation of 10 cm represents 8 parts over 10000 for 125 m, on the contrary 3 parts over 10000 for 300 m). Nevertheless, traditional sensors obviously do not provide knowledge of the lander's altitude to such precise extent.

6.3.2 Errors in Camera Orientation

Concerning the robustness to errors in camera orientation, deviations of the camera pointing direction were considered only along the plane containing both the normal to the surface and the camera axis. With regards to the results of the sensitivity to camera position errors, the analysis was limited to only equal angular deviations for both camera poses thus allowing to exclude major error contributions. The results are alarming: already +0.05° difference at 250 and 300 m altitude is responsible for DEM errors exceeding the maximum allowable (Tables 6.18 and 6.19). Besides, slightly larger angular deviations contribute to an exorbitant increase of the DEM errors for all altitudes (Tables 6.16 and 6.17), proving useless to assess the performances for greater angular variations.

Overall, these results clearly indicate that not only traditional sensors do not provide knowledge of the lander's location and attitude accurate enough for structure-from-motion hazard detection but also, that developed structure-from-motion algorithm is not suitable for hazard detection as a stand-alone method. As matter of fact, even if visual based navigation is employed, it is unrealistic to expect such sensor to track the position and orientation of the lander to a sub-pixel accuracy.

Table 6.12: Sensitivity analysis on camera position: DEM errors, altitude 125 m

Altitude h_1 $[m]$	**Altitude h_2 $[m]$**	μ_{DEM} $[m]$	$\frac{\Delta\mu_{DEM}}{\mu_{DEM}}$ [%]	median_{DEM} $[m]$	$\frac{\Delta\text{median}_{DEM}}{\text{median}_{DEM}}$ [%]	MAD_{DEM} $[m]$	$\frac{\Delta\text{MAD}_{DEM}}{\text{MAD}_{DEM}}$ [%]
250.4172	124.6656	3.4898	382.32	1.5343	-740.95	2.3292	370.05
249.7858	125.102	2.4035	232.19	-0.98742	-312.49	1.8971	282.85
250.2572	124.763	2.7899	285.59	1.1312	-572.55	2.098	323.39
250.2537	25.1541	0.91687	26.72	-0.27313	-14.10	0.643	29.76
249.8804	125.1892	2.6859	271.22	-1.1295	-371.84	2.1548	334.86
250.0678	125.2482	2.439	237.09	-0.99656	-316.31	1.9255	288.58
249.5759	124.9505	2.0406	182.03	-0.78034	-225.98	1.5839	219.64
249.554	124.5838	1.4965	106.83	0.37966	-258.60	1.0287	107.60
250.0308	124.729	2.267	213.32	0.79495	-432.09	1.6648	235.97
250.2792	125.4133	2.8566	294.81	-1.1923	-398.08	2.3511	374.47
250.434	124.6524	3.5556	391.42	1.5258	-737.40	2.4098	386.32
249.6299	125.3258	4.2962	493.76	-2.2559	-842.39	4.137	734.88
250.0688	125.0383	0.74901	3.52	-0.24205	-1.15	0.51113	3.15
249.9694	125.4961	4.2955	493.68	-2.2509	-840.30	4.1436	736.21
250.0897	125.0613	0.83617	15.57	-0.25939	-8.36	0.59154	19.38
249.8371	124.9427	0.8957	23.79	-0.27849	-16.38	0.62691	26.55
249.6622	124.6067	1.7617	143.48	0.51175	-313.78	1.2382	149.88
250.2943	125.4619	3.1455	334.74	-1.3503	-464.08	2.6711	439.05
249.8112	124.5046	2.971	310.62	1.229	-613.41	2.1847	340.89
250.0285	125.2749	2.7657	282.25	-1.1517	-381.17	2.251	354.27

Table 6.13: Sensitivity analysis on camera position: DEM errors, altitude 150 m

Altitude h_1 $[m]$	**Altitude h_2 $[m]$**	μ_{DEM} $[m]$	$\frac{\Delta\mu_{DEM}}{\mu_{DEM}}$ [%]	median_{DEM} $[m]$	$\frac{\Delta\text{median}_{DEM}}{\text{median}_{DEM}}$ [%]	MAD_{DEM} $[m]$	$\frac{\Delta\text{MAD}_{DEM}}{\text{MAD}_{DEM}}$ [%]
299.9419	149.8383	0.92488	11.25	0.013544	-106.50	0.74416	13.63
300.4374	150.3548	1.7271	107.74	-0.35202	-68.92	1.4331	118.84
299.9652	149.6635	2.2012	164.76	0.44772	-314.85	1.333	103.55
299.9702	149.7825	1.384	66.47	0.19738	-194.72	1.0087	54.03
300.0534	150.3862	3.1195	275.22	-1.1113	-433.28	2.7445	319.09
299.8366	150.231	2.8535	243.22	-0.94085	-351.48	2.4561	275.05
300.2771	150.3266	2.0742	149.49	-0.49786	-138.91	1.7493	167.12
300.3696	149.7245	3.076	269.98	0.52143	-350.22	1.7673	169.87
299.8351	150.3522	3.5408	325.89	-1.3803	-562.36	3.1819	385.88
299.7633	149.6051	1.8956	128.01	0.35336	-269.57	1.2321	88.14
300.4158	150.0562	1.0407	25.18	0.069761	-133.48	0.828	26.44
299.5372	150.2246	3.647	338.66	-1.3995	-571.58	3.2933	402.89
299.6906	150.2296	3.2659	292.82	-1.2178	-484.39	2.8938	341.89
300.0616	149.5448	3.2296	288.46	0.55811	-367.82	1.8164	177.37
300.348	149.9672	1.4128	69.93	0.21204	-201.75	1.0281	56.99
300.1178	150.1895	1.6912	103.42	-0.34625	-66.15	1.4023	114.13
299.7044	149.8704	0.93185	12.08	-0.20652	-0.90	0.7362	12.42
300.3172	149.8843	1.8787	125.97	0.36046	-272.97	0.57661	11.95
300.4324	149.8135	2.7287	228.21	0.5242	-351.55	1.566	139.13
300.0105	150.0198	0.90767	9.18	-0.19304	-7.37	0.71585	9.31

Table 6.14: Sensitivity analysis on camera position: DEM errors, altitude 250 m

Altitude h_1 $[m]$	**Altitude h_2 $[m]$**	μ_{DEM} $[m]$	$\frac{\Delta\mu_{DEM}}{\mu_{DEM}}$ [%]	median_{DEM} $[m]$	$\frac{\Delta\text{median}_{DEM}}{\text{median}_{DEM}}$ [%]	MAD_{DEM} $[m]$	$\frac{\Delta\text{MAD}_{DEM}}{\text{MAD}_{DEM}}$ [%]
499.9016	250.1499	2.5608	133.25	1.6307	366.67	0.97398	59.50
500.3697	250.4782	3.3005	200.62	2.2225	536.04	1.18	93.24
499.9463	250.4972	4.874	343.94	3.6395	941.55	1.3706	124.45
499.7504	250.2076	3.6006	227.95	2.4792	609.50	1.2608	106.47
499.7573	249.6661	1.5237	38.78	-1.0293	394.57	0.72892	19.37
499.5331	249.9226	2.2184	102.06	1.3508	286.57	0.88764	45.36
500.3833	250.4385	2.9315	167.01	1.9242	450.67	1.0772	76.41
500.3394	249.5236	4.7968	336.91	-3.913	1219.82	1.7585	187.98
499.9348	249.5919	2.7293	148.59	-2.0958	699.78	1.0033	64.30
500.1774	249.5654	3.8431	250.04	-3.0819	981.98	1.4351	135.02
499.678	250.0547	2.689	144.92	1.7344	396.35	1.0053	64.63
499.6291	249.996	2.4216	120.57	1.5181	334.45	0.93408	52.97
500.105	250.0674	1.182	7.66	0.43819	25.40	0.62686	2.66
499.7172	250.3521	4.6898	327.16	3.4777	895.25	1.3885	127.38
499.8261	249.6013	2.2681	106.58	-1.683	581.64	0.87885	43.92
499.9545	249.6185	2.6076	137.51	-1.9867	668.55	0.97	58.85
499.8905	250.2168	3.1249	184.63	2.0773	494.48	1.1339	85.69
500.1305	250.2966	2.8107	156.01	1.8323	424.37	1.0467	71.41
499.9014	250.1785	2.7839	153.57	1.8083	417.50	1.0345	69.41
499.7967	250.0456	2.1483	95.67	1.2924	269.86	0.88002	44.11

Table 6.15: Sensitivity analysis on camera position: DEM errors, altitude 300 m

Altitude h_1 $[m]$	**Altitude h_2 $[m]$**	μ_{DEM} $[m]$	$\frac{\Delta\mu_{DEM}}{\mu_{DEM}}$ [%]	median_{DEM} $[m]$	$\frac{\Delta\text{median}_{DEM}}{\text{median}_{DEM}}$ [%]	MAD_{DEM} $[m]$	$\frac{\Delta\text{MAD}_{DEM}}{\text{MAD}_{DEM}}$ [%]
599.5521	300.2972	4.7806	283.58	3.5471	821.59	1.3336	87.80
599.6411	300.2778	4.3794	251.39	3.1864	727.87	1.2653	78.18
600.0346	299.9919	1.1474	7.94	0.24824	35.50	0.69981	1.45
600.4394	300.3524	2.0539	64.80	1.1616	201.80	0.94261	32.74
600.3972	299.9967	1.4493	16.29	-0.93044	341.74	0.8055	13.43
599.8448	300.348	4.1615	233.91	3	679.44	1.2528	76.42
600.3087	300.4174	2.9992	140.65	1.9823	415.03	1.1155	57.08
599.8701	300.1144	2.3878	91.59	1.4427	274.83	1.011	42.37
599.9951	300.0293	1.406	12.81	0.56211	46.04	0.73806	3.93
599.5915	300.1856	3.9057	213.38	2.7779	621.74	1.2356	73.99
599.5475	300.0879	3.3593	169.54	2.283	493.16	1.1925	67.93
600.2797	299.783	2.5337	103.30	-1.9606	609.39	1.0084	42.00
599.579	300.1203	3.4826	179.43	2.394	521.10	1.2116	70.62
600.3779	300.3168	2.0193	62.02	1.1327	194.29	0.93505	31.67
599.6263	300.4081	5.2072	317.81	3.9191	918.24	1.368	92.64
599.8559	300.4919	5.0349	303.99	3.7809	882.33	1.3569	91.08
600.226	300.1458	1.4127	13.35	0.57045	48.21	0.73917	4.10
600.0397	300.1864	2.2956	84.19	1.3647	254.57	0.99157	39.63
600.437	299.639	4.2364	239.92	-3.437	992.98	1.4518	104.44
599.9277	299.626	2.391	91.85	-1.8343	576.58	0.97613	37.46

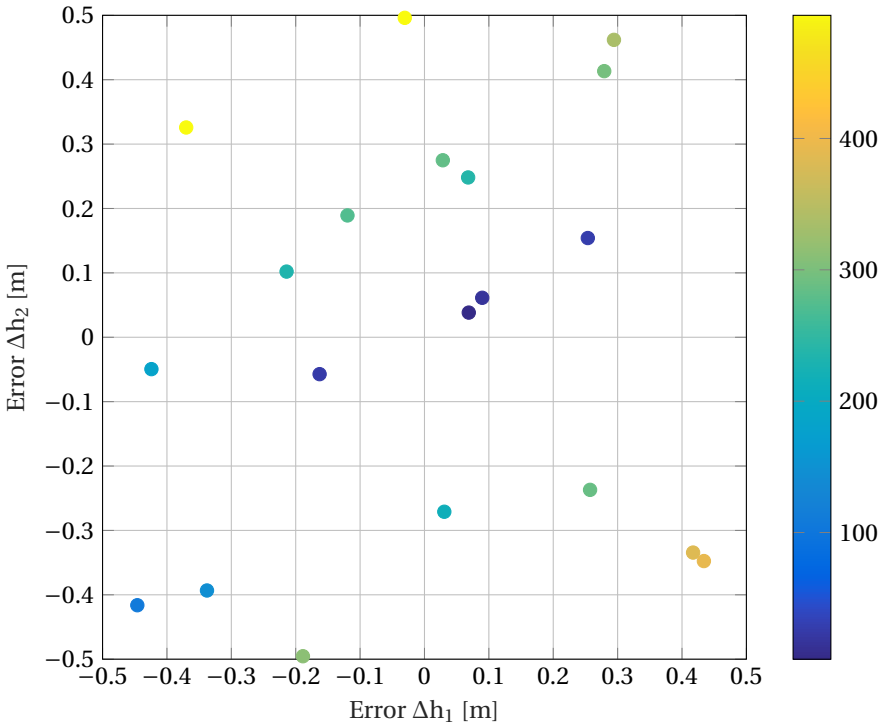

Figure 6.13: DEM mean relative percentage errors as function of the deviations on altitude $(\Delta h_1, \Delta h_2)$ at both camera poses. Altitude 125 m

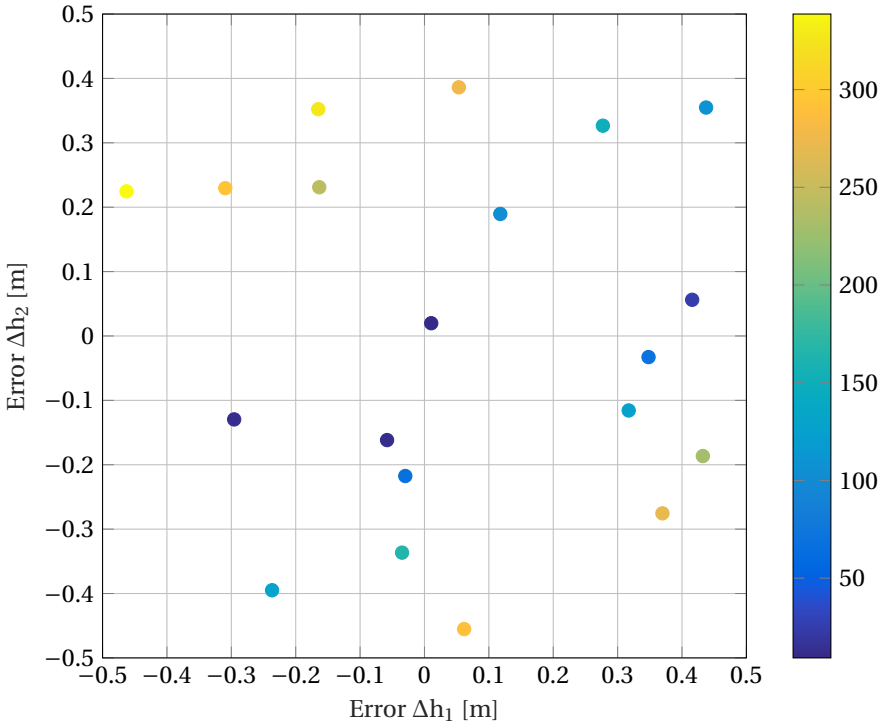

Figure 6.14: DEM mean relative percentage errors as function of the deviations on altitude $(\Delta h_1, \Delta h_2)$ at both camera poses. Altitude 150 m

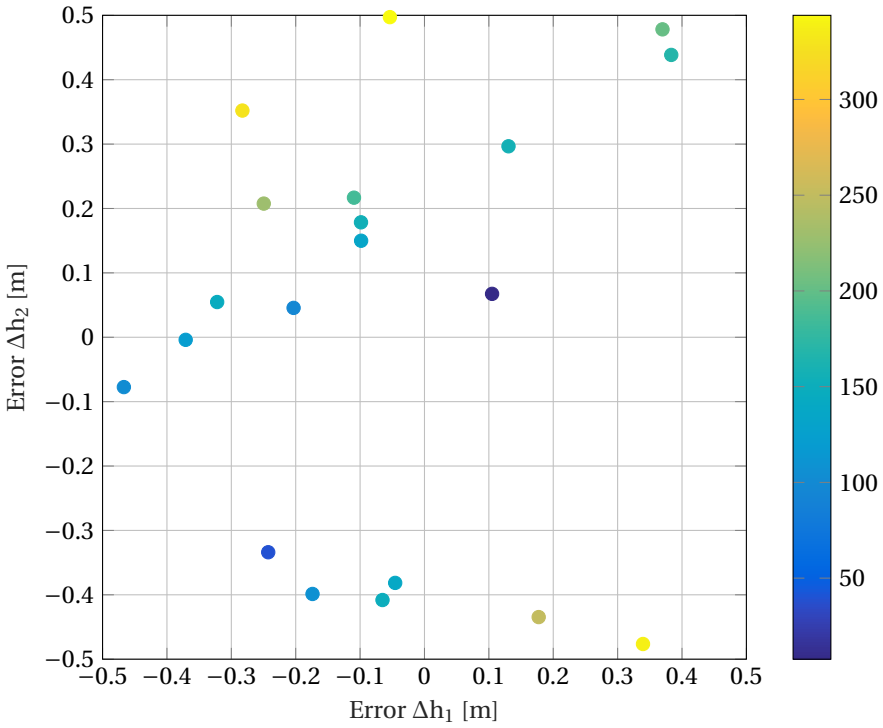

Figure 6.15: DEM mean relative percentage errors as function of the deviations on altitude $(\Delta h_1, \Delta h_2)$ at both camera poses. Altitude 250 m

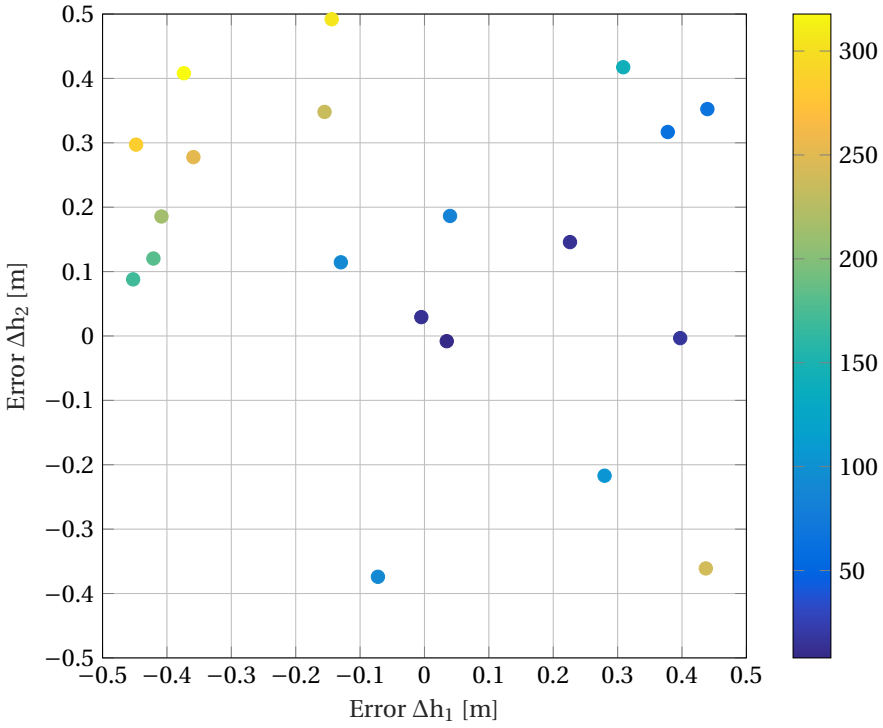

Figure 6.16: DEM mean relative percentage errors as function of the deviations on altitude $(\Delta h_1, \Delta h_2)$ at both camera poses. Altitude 300 m

Table 6.16: Sensitivity analysis on camera orientation: DEM errors, altitude 125 m

	Angular Deviations [deg]					
	-0.15	**-0.1**	**-0.05**	**0.05**	**0.1**	**0.15**
$\boldsymbol{\mu_{DEM}}$ **[m]**	3.319	2.5111	1.6181	0.79903	1.739	2.5358
$\frac{\Delta\mu_{DEM}}{\mu_{DEM}}$ **[%]**	358.72	247.06	123.64	10.43	140.35	250.47
median$_{DEM}$ **[m]**	-2.0058	-1.3981	-0.74589	0.26706	0.87236	1.4628
MAD$_{DEM}$ **[m]**	2.7268	1.9245	1.1736	0.56939	1.1587	1.8045

Table 6.17: Sensitivity analysis on camera orientation: DEM errors, altitude 150 m

	Angular Deviations [deg]					
	-0.15	**-0.1**	**-0.05**	**0.05**	**0.1**	**0.15**
$\boldsymbol{\mu_{DEM}}$ **[m]**	3.552	2.6768	1.7351	0.89822	1.7418	2.5245
$\frac{\Delta\mu_{DEM}}{\mu_{DEM}}$ **[%]**	327.24	221.97	108.70	8.038345	109.51	203.65
median$_{DEM}$ **[m]**	-1.9101	-1.2289	-0.5786	0.22821	0.67785	0.89848
MAD$_{DEM}$ **[m]**	-1.9101	-1.2289	-0.5786	0.22821	0.67785	0.89848

Table 6.18: Sensitivity analysis on camera orientation: DEM errors, altitude 250 m

	Angular Deviations [deg]					
	-0.15	**-0.1**	**-0.05**	**0.05**	**0.1**	**0.15**
$\boldsymbol{\mu_{DEM}}$ **[m]**	5.1488	3.8752	2.4324	1.3132	2.497	3.7694
$\frac{\Delta\mu_{DEM}}{\mu_{DEM}}$ **[%]**	368.97	252.96	121.55	19.61	127.43	243.33
median$_{DEM}$ **[m]**	2.7955	2.0401	1.1959	-0.50213	-1.3604	-2.2364
MAD$_{DEM}$ **[m]**	1.8853	1.5235	1.0168	0.76611	1.1896	1.787

Table 6.19: Sensitivity analysis on camera orientation: DEM errors, altitude 300 m

	Angular Deviations [deg]					
	-0.15	**-0.1**	**-0.05**	**0.05**	**0.1**	**0.15**
$\boldsymbol{\mu_{DEM}}$ **[m]**	5.689	4.2881	2.6691	1.4963	2.9773	4.5027
$\frac{\Delta\mu_{DEM}}{\mu_{DEM}}$ **[%]**	356.47	244.07	114.16	20.06	138.89	261.28
median$_{DEM}$ **[m]**	3.2432	2.3459	1.3228	-0.64804	-1.7048	-2.7886
MAD$_{DEM}$ **[m]**	1.7897	1.5362	1.1259	0.90475	1.412	2.137

6.3.3 Dense Reconstruction Image Center

The algorithm capability to achieve dense reconstruction near the image center, with adjacent descent images taken at less than half altitude from one to the other has also been tested. Figure 6.17 highlights the input images considered: for this analysis the image at 1st camera pose is taken at 375 m, while the image at 2nd camera pose is captured at 250 m above the surface. Therefore, the two adjacent descent images are taken at 1/3 altitude from one to the other. The computed DEM, together with the ground-truth and related error map are presented in Figure 6.18. Their inspection reveals how the elevation is highly underestimated for the hollow region located on the top-left corner, besides, as expected high errors (greater than 3 m) characterize the whole image center. Previously in Section 3.3, it was indeed proved how the point disparity is directly proportional to its distance from the image center and the ratio of descent height to elevation d/Z (Equation 3.12). The relation is brought again to the reader's attention below, for the sake of completeness

$$\Delta \boldsymbol{x} = \frac{d}{Z} \bar{\boldsymbol{x}}_{t_1} \tag{6.1}$$

where $\Delta \boldsymbol{x}$ represents the point disparity, d the distance travelled by the lander, and $\bar{\boldsymbol{x}}_{t_1}$ the coordinate of a point belonging to the second image relative to the principal point (image center).

For the present state $d/Z = 1/3$, therefore points satisfying $\bar{\boldsymbol{x}}_{t_1} = 1$ pixel are only affected by a ~0.3 disparity which is insufficient for a good recovery around the center. The DEM errors are presented in Table 6.20. Overall, the errors are >2 times higher than those obtained when the altitude pair 250-500 m is considered (Table 6.9).

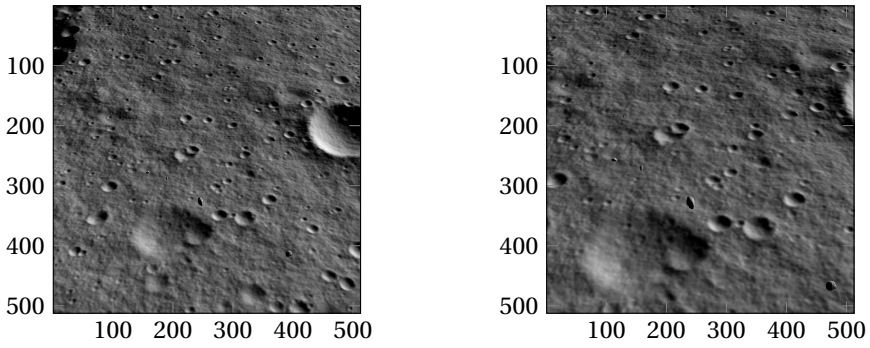

Figure 6.17: PANGU input image-pair. Image I_1 taken at 1st camera pose (h = 375 m) while image I_2 when the altitude drops to 250 m (right)

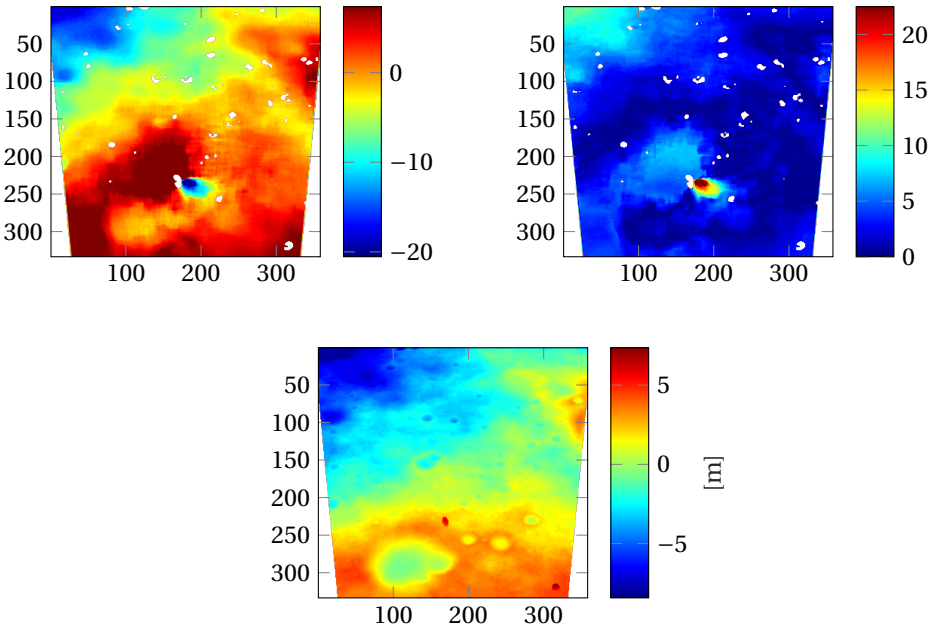

Figure 6.18: Computed DEM (top-left) ground-truth (top-right) and DEM error (bottom). Altitude 250 m

Table 6.20: Test 1 Dense Reconstruction: DEM errors

μ_{DEM} [m]	median$_{DEM}$ [m]	MAD$_{DEM}$ [m]
2.6179	-0.41971	1.8359

Shape recovery for images taken at 2/3 altitude from one to the other has also been tested. For this analysis, a 900-300 m altitude data set was available only for a vertical motion along the camera axis (Figure 6.19). The DEMs and the computed errors are shown in Figure 6.20 and Table 6.21, respectively. In contrast to what is expected, the results are worse than those computed for the 600-300 m data set (Table 6.2). As matter of fact, with a ratio of descent height to elevation $d/Z = 2/3$, points 1 pixel far from the image center have ~0.66 increased disparity. Nevertheless, it should be reminded that for a vertical motion the epipole lies in the image center, therefore improvements in the reconstruction around this region can not be observed. For more insights, data sets with an imaging sensor line-of-site offset should be considered. Due to time constraints, further investigation on this specific case is left as a recommendation for future work.

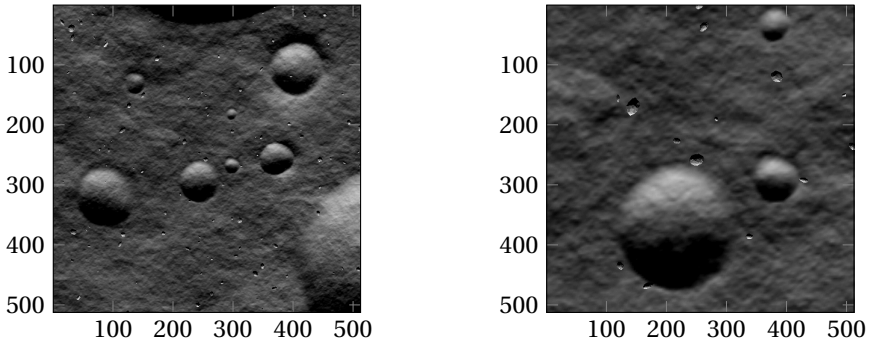

Figure 6.19: PANGU input image-pair. Image I_1 taken at 1st camera pose (h = 900 m) while image I_2 when the altitude drops to 300 m (right)

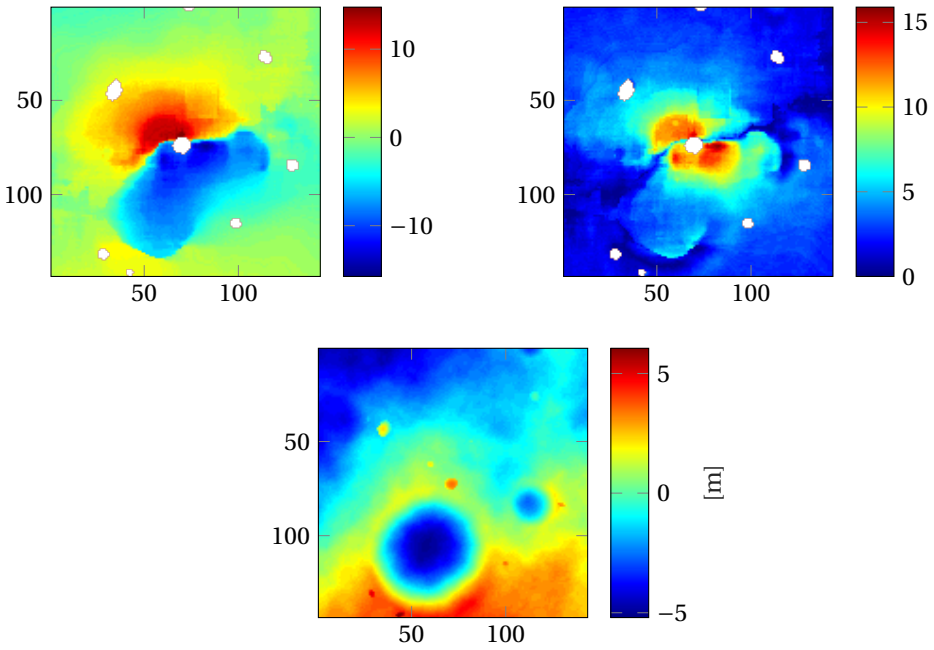

Figure 6.20: Computed DEM (top-left) ground-truth (top-right) and DEM error (bottom). Altitude 300 m

Table 6.21: Test 2 Dense Reconstruction: DEM errors

μ_{DEM} [m]	median_{DEM} [m]	MAD_{DEM} [m]
3.6091	1.0302	3.0791

6.4 Computational Time

Beforehand, in Section 4.3, the software computational time was considered for the assessment of the optimal size of the correlation window and depth step. Nevertheless, the analysis was only limited to the the functions in charge of the correlation and shape recovery. Therefore, the complete execution time profiling of the software is presented in Figure 6.21. For the sake of simplicity, the whole software profile has been divided into only 5 main tasks as follows:

- *load*, within this task the input image pair is loaded into the workspace besides, parameters necessary for the homography-based shape recovery are computed (e.g. camera projection matrices)
- *Rock Map*, rock detection assessment through the computation of a rock boolean map for the image taken at camera 1st pose
- *DEM*, 3D reconstruction of the landing region
- *Georeferencing*, task to assign real-world coordinates to each pixel of the raster
- *HMAP*, evaluation of the hazardousness of the landing region

Figure 6.21 highlights that the major contribution is due to the computation of the hazard map. Further analysis also reveals that most time is required for the slope assessment (approximately 30.67 s), which indeed is more computationally demanding. On the contrary, tasks like Rock Map and Georeferencing only requires only 0.0347 s and 0.0569 s, respectively, being hardly visible in the bar graph. It should be remembered that the simulation software is written in *MATLAB* language, therefore the execution time is always above the allowable maximum of 2 s, for all performed test cases. This was clear already by the inspecting the time required for the correlation in Section 4.3. Figure 6.21 shows that the current set-up allows to achieve a complete hazard map in less than 75 s. Therefore, it is expected that a better implementation and the employment of C/C++ software language will enable to accomplish the real-time requirements

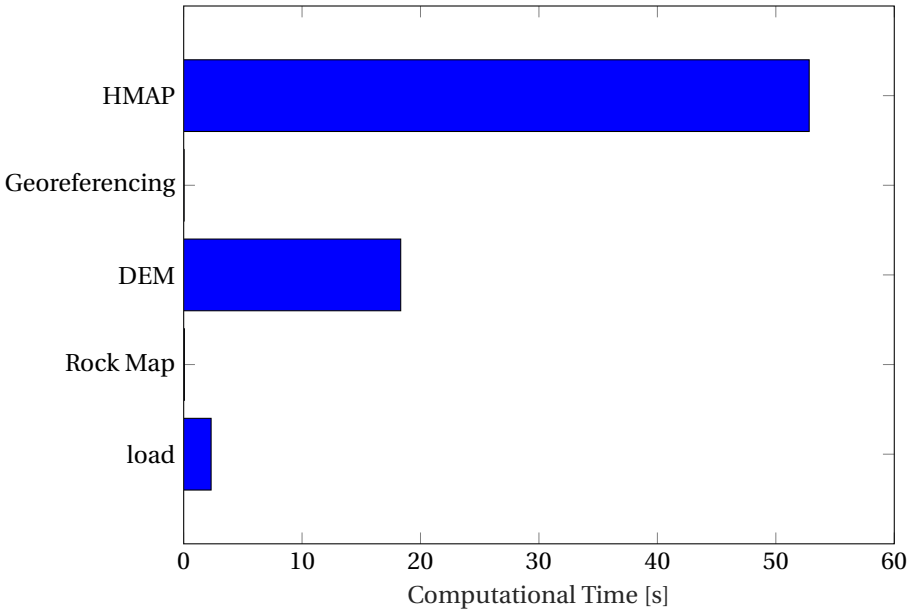

Figure 6.21: Software execution time profiling

Part V
Conclusion and Recommendation

Chapter 7

Conclusions

The main research goal of the project is *to numerically investigate the limitations of a SfM algorithm for planetary landings, given traditional sensor (e.g. IMU and altimeter) measurements to assess the lander's location.* The better structure the research effort the following SBs have been identified:

SB1: *Solutions to tackle depth recovery in the vicinity of the epipole*

SB2: *Algorithm robustness to errors in both camera position and orientation*

SB3: *Assess whether dense reconstruction, near the image center, is possible with adjacent image pairs such that one is taken at less than half altitude (e.g. one third) of the other image*

Concerning SB1, two different mission scenarios were considered: 1) a perfect vertical motion forward along the camera pointing direction and 2) a 45° angle dropping trajectory for a more realistic approaching descent phase, with a 40° imaging sensor line-of-site offset. A perfect knowledge of both lander's position and orientation was initially assumed.

For the former scenario, the epipole coincides with the nominal landing site (image center). The results show that its presence seriously affect depth recovery within an area of approximately 80×80 px^2 around the image center. This represents a significant problem since these locations are the most convenient in terms of distance and guidance costs. As these sites are wrongly classified as unsafe, they result unaccessible. Excluding the epipole region has been proved to have no significant impact rather than decreasing the number of false alarms. For the latter scenario, when the nominal pitch angle is set to 9° the epipole is always located below the image center. Instead it moves upward when the lander attitude is aligned with the normal direction of the surface. The results for this descent sequence indicates that below 300 m altitude the software is a suitable candidate for hazard detection, with total correct detection on average >94% and the percentage of undetected hazards below the allowable maximum 1%.

Thus, the *ad hoc* selection of descent trajectory and imaging sensor line-of-site offset (to move the epipole away from the image center or potentially outside the image boundaries) has been shown to shrink the epipole region and to improve the overall depth recovery. In addition, the analysis of the computed depth maps has highlighted that rocks and boulder are not well resolved from both high- and low-elevation image pairs, in contrast to the result NASA JPL [43]. The presence of rocks on the surface is accounted by employing an adapted version of the Harris Corner detector directly on the input images.

Addressing SB2, a Monte Carlo simulation was performed to assess the algorithm robustness to errors in camera position. Random uncertainties within the interval [-0.5 0.5] meters were taken into account for the altitude of both camera poses. As expected, the lower DEM errors are obtained when both altitudes are affected by small deviations of similar magnitude and same sign (approximately ±10 cm). Indeed, for this specific case the ratio h_2/h_1 of camera poses altitudes is really close to $1/2$, necessary to achieve dense reconstruction near the image center. The relative percentage errors peak above 100% everywhere else within the considered interval. Moreover, concerning the robustness to errors in camera orientation, deviations of the camera pointing direction were considered only along the plane containing both the normal to the surface and the camera axis. Already differences greater than ±0.05°, in the imaging sensor line-of-site, are responsible for exorbitant errors in the DEM for all altitudes. Overall, these results allow us to answer the lower-level thesis RQs reported below for the sake of completeness

RQ1.1 *Is Structure from Motion suitable for HD as a stand-alone method?*

RQ1.2 *Do traditional sensors (e.g. IMU and altimeter) provide knowledge of the lander's location accurate enough for SfM Hazard Detection or Visual based Navigation (VN) has to be added to GNC architecture?*

Clearly, the sensitivity analysis has highlighted that not only traditional sensors do not provide knowledge of the lander's location and attitude accurate enough for SfM hazard detection but also, that developed SfM algorithm (at current stage) is not suitable for hazard detection as a stand-alone method. As a matter of fact, even if visual based navigation is employed, it is unrealistic to expect such sensor to track the position and orientation of the lander to a sub-pixel accuracy.

With reference to SB3, the capability to achieve dense reconstruction near the image center with adjacent descent images taken at less than half altitude from one to the other has also been tested. For this analysis the image at 1st camera pose is taken at 375 m, while the image at 2nd camera pose is captured at 250 m above the surface. Therefore, the two adjacent descent images are taken at 1/3 altitude from one to the other. In agreement with the expectations, the computed errors are >2 times higher than those obtained when the altitude pair 250-500 m is considered: the lower pixel disparity, around the image center, is insufficient to achieve a dense recovery around the aforementioned region.

The high-level RQ set to answered for the project is shown below

RQ1 *Is it possible to increase autonomy for planetary landings using equipment readily available on space missions?*

For this reason, the framework of hazard detection has been considered. Through a single camera, therefore employing equipment readily available on space missions, it was shown that valuable information related to the hazardousness of the landing region can be obtained. This potentially allows to increase the autonomy and decrease the risk of a landing failure for future exploration missions. Nevertheless, several limitations of the developed algorithm have been marked, highlighting that further research is needed to properly address them and further mature this technology.

Chapter 8

Recommendations for Future Work

Having concluded the results of this MSc thesis, a number of recommendations for future work on the current topic are listed below.

- Concerning shape recovery, the limitation of projective warping implemented through bilinear interpolation have clearly emerged. Unfortunately, here, there are not many parameters to play with since higher order interpolations are way too computational expensive for the tight real-time constraints imposed by planetary landings. Different approaches to correlate input image pairs affected by high resolution differences (as those considered in this research) shall be explored. Since this involves re-thinking a core part of the algorithm, completely from scratch, it is author's personal opinion that a strong background in projective geometry and computer vision would be required.

- As stated in Section 4.2, the presence of rocks and boulders on the surface is accounted through an adapted version of the Harris Corner detector directly on the input images. For the scenario simulating the approaching phase of Chang'e-3 lunar landing, it was shown that also hazardous small craters can be identified. Therefore, further investigation whether to enhance the rock detection block and totally discard the computation of roughness, based on the resulting DEMs, should be made. Initially, a different approach was developed to solve the square-pattern problem: the SSD chosen for correlation was replaced with the SAD operator for pixels, within the correlation window, detected by the Harris Corner measure to depict boulders and their immediate proximity. By taking the absolute difference of the pixel intensities rather than their squares, their impact was indeed minimized. Nevertheless, this approach was verified to fail for the images belonging to the second mission scenario, which was only considered at a late stage of the project.

- Considering the perfect vertical motion along the camera axis, a different data-set without features that could negatively affect shape recovery for lower altitudes should be analyzed. This recommendation is to prove that better results can be achieved as the lander is steered towards the surface, since higher resolution images are available.

- All the scenes generated correspond to a 30° FOV projective camera with a 512×512 pixel 8-bits sensor. It may be valuable to assess whether employing a higher resolution sensor (for instance 16-bits) would lead to improve the overall accuracy of the depth maps.

- The previous three points highlight that it will be convenient for the student to obtain a license for the PANGU software, to be more independent for the creation of tailored data-sets. This is a process based on trials and errors, and it turns out difficult for the student to ask further in advance for data-set with features complying with specific requirements.

- For an improved and advanced version of the software, it will be interesting to further improve the sensitivity analysis by considering lateral deviations in the lander's position and orientation at each camera pose. In addition, it will be valuable to implement a flight version of the software (C/C++ implementation) to assess whether the algorithm can meet the tight real-time constraints and for validation in an experimental landing facility, gaining the highest possible confidence for the in-flight operations.

Appendices

Appendix A

Fundamentals of Computer Vision

This appendix presents the geometric concepts for computer vision which represents a necessary background for this thesis. Its drafting has been mostly based on [32], elsewhere stated otherwise. Section A.1 introduces homogeneous coordinates and notations of projective geometry, central for the analysis of two view geometry. Homographies, namely transformations that arise when a plane is imaged by a perspective camera, are outlined in Section A.2, while epipolar geometry is extensively covered in Section A.3.

A.1 Homogeneous Coordinates

As a basic concept of planar geometry, it is well known that a line is represented by an equation such as $ax + by + c = 0$. Of course, different choices of the coefficients a, b and c give rise to different lines in the plane. Therefore, it appears straightforward that a line can be defined by the vector $(a,b,c)^T$. Nevertheless, it must be highlighted that the correspondence between vectors and lines is not one-to-one: equations $ax + by + c = 0$ and $(ka)\,x + (kb)\,y + (kc) = 0$ are indeed the same, for any non-zero constant k. Thus, under such assumption, $(a,b,c)^T$ and $k\,(a,b,c)^T$ represent the same line. Vectors equal by an overall scaling are *equivalent* besides, a class of equivalent vectors is defined *homogeneous*. It is possible now to introduce a new structure, namely the *Projective Space* $\mathbb{P}^2$ which is described by the set of all classes of homogeneous vectors of $\mathbb{R}^3$ - $(0,0,0)^T$.
Let us now consider a point $\boldsymbol{x} = \left(x,y\right)^T$ in the plane. Such a point lies on the line $\boldsymbol{l} = (a,b,c)^T$ if and only if $ax + by + c = 0$. The previous equation can be expressed through a scalar product of vectors as follows

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \boldsymbol{l} = \boldsymbol{x}^T \boldsymbol{l} = 0 \tag{A.1}$$

where $\left(x,y\right)^T \in \mathbb{R}^2$ has been represented through a 3-vector by adding an additional coordinate equal to 1. Multiplying by a non-zero constant k, the previous expression becomes $\left(kx,ky,k\right)\boldsymbol{l} = 0$. Thus, it appears natural to consider the set of vectors $\left(kx,ky,k\right)^T$ as a representation of $\left(x,y\right)^T$ in the Projective Space. As result, points as homogeneous 3-vectors are also elements of $\mathbb{P}^2$. The key role played by homogeneous coordinates will be further highlighted in Section A.2, nevertheless in the following it will be shown how they allow a natural representation of points at infinity (or *ideal points*), without the necessity of taking limits. Consider a point $(x_1,x_2,0)^T$: now, if we attempt to find its inhomogeneous representation we would obtain $(x_1/0,x_2/0)^T$, which makes no sense. The previous expression, however, suggests that the point has infinitely large coordinates. Thus, the set of ideal points can be specified by a 3-vector whose last coordinate equals zero. In the previous paragraph, a result has been mentioned and it is further highlighted below

Result 3.1 *A point $\boldsymbol{x}$ lies on the line $\boldsymbol{l}$ if and only if $\boldsymbol{x}^T\boldsymbol{l} = 0$*

The former can be used to prove the following result

Result 3.2 *The intersection of two lines* $\boldsymbol{l}$ *and* $\boldsymbol{l}'$ *is the point* $\boldsymbol{x} = \boldsymbol{l} \times \boldsymbol{l}'$

Proof
Let us consider two lines $l = (a,b,c)^T$ and $l' = \left(a',b',c'\right)^T$, we wish to determine their intersection. Define the vector $\boldsymbol{x} = \boldsymbol{l} \times \boldsymbol{l}'$, where × symbolizes the cross product. From $\boldsymbol{l} \cdot \left(\boldsymbol{l} \times \boldsymbol{l}'\right) = 0$ and $\boldsymbol{l}' \cdot \left(\boldsymbol{l} \times \boldsymbol{l}'\right) = 0$, it is possible to deduce $\boldsymbol{l}^T \boldsymbol{x} = \boldsymbol{l}'^T \boldsymbol{x} = 0$. Therefore, $\boldsymbol{x}$ lies both on $\boldsymbol{l}$ and $\boldsymbol{l}'$ hence is the intersection of the two lines.

Similarly, the expression for the line passing through two points can be derived.

Result 3.3 *The line through two points* $\boldsymbol{x}$ *and* $\boldsymbol{x}'$ *is* $\boldsymbol{l} = \boldsymbol{x} \times \boldsymbol{x}'$

The purpose of highlighting the previous results is twofold: they introduce the notation applied throughout the entire chapter furthermore, they enable an handy and algebraic representation of basic geometric concepts (e.g. belonging and intersection) which are recurring in modelling the geometry of multiple views in computer vision.

A.2 Projective Geometry

In the previous section we have seen that homogeneous coordinates represent ideal points without the necessity of taking limits. Besides, homogeneous coordinates allow non-linear mappings (such as perspective projections) to be defined through linear matrix equations. In the following, projective transformations of 2-space will be introduced. These are the transformations that model the perspective geometric distortion which arises when a plane is imaged by a camera.

Definition 3.1

A *projectivity* (or *homography*) is an invertible mapping h from $\mathbb{P}^2$ to itself such that three points $\boldsymbol{x_1}$, $\boldsymbol{x_2}$ and $\boldsymbol{x_3}$ lie on the same line if and only if $h(\boldsymbol{x_1})$, $h(\boldsymbol{x_2})$ and $h(\boldsymbol{x_3})$ do

Therefore projectivities preserve *collinearity*, in other words they map lines into lines.
The previous definition has been expressed in terms of a coordinate-free viewpoint, nevertheless an algebraic definition of homography is also possible and stems from the following theorem

Theorem 3.1

A mapping h: $\mathbb{P}^2 \rightarrow \mathbb{P}^2$ is a projectivity if and only if there exists a non-singular 3×3 matrix $\boldsymbol{H}$ such that for any point in $\boldsymbol{P}^2$ represented by a vector $\boldsymbol{x}$ it is true that
$h(\boldsymbol{x}) = \boldsymbol{H}\boldsymbol{x}$

The theorem asserts that a planar projectivity is a linear transformation on homogeneous 3-vectors and conversely any such mapping is a projectivity.

$$\begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{A.2}$$

Note that $\boldsymbol{H}$ is a homogeneous matrix and it may be changed by a multiplication factor. Therefore, with only the ratio of its elements being significant, the transformation has 8 degrees of freedom (dof).

A.3 Epipolar Geometry

Epipolar geometry encodes the relation between two images of the same scene but taken from different viewpoints. Being independent of the scene structure, it is only affected by camera's internal parameters and relative pose. As it will appear clearer, this geometry is motivated by the search of corresponding points within the image pair. Consider a point $\boldsymbol{X}$ in 3-space which is imaged into $\boldsymbol{x}$ in the first view, while $\boldsymbol{x}'$ in the second view (Figure A.1). We wish to determine the existing relation between $\boldsymbol{x}$ and $\boldsymbol{x}'$.

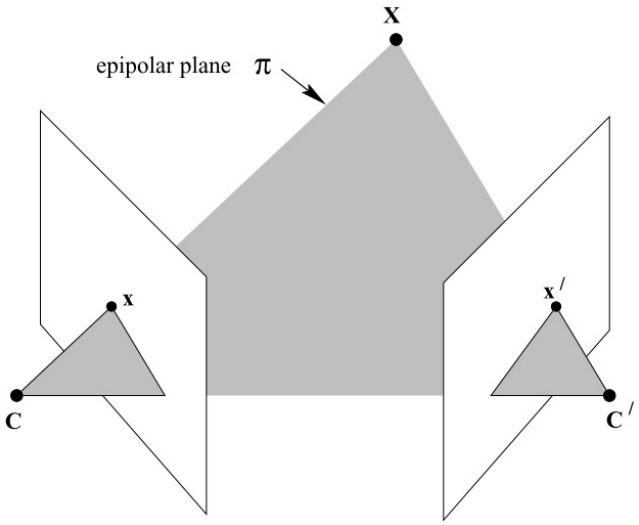

Figure A.1: The two cameras centres, 3D world point $\boldsymbol{X}$, and its 2D images lie in a common plane, namely $\boldsymbol{\pi}$ [32]

The world point $\boldsymbol{X}$, the image pair $(\boldsymbol{x}, \boldsymbol{x}')$, and the camera centres $\boldsymbol{C}$, $\boldsymbol{C}'$ are all coplanar. Therefore, also the rays back-projected from $\boldsymbol{x}$ and $\boldsymbol{x}'$ lie on such plane denoted by $\boldsymbol{\pi}$. Assuming that $\boldsymbol{x}$ is known, how is the corresponding point $\boldsymbol{x}'$ constrained? From what has been previously stated, $\boldsymbol{x}'$ must lie on $\boldsymbol{\pi}$ hence it belongs to the line $\boldsymbol{l}'$, intersection of $\boldsymbol{\pi}$ with the second image plane. This line is the image in the second view of the ray back-projected from $\boldsymbol{x}$ and is named *epipolar line*. Figure A.2 illustrates the geometric entities involved in the epipolar geometry:

- epipole (e), the point of intersection of the line joining the camera centres (baseline) with image plane. It can be also defined as the image in one view of the camera centre of the other view.
- *epipolar plane* (π), plane containing the baseline: there is a pencil of epipolar planes.
- *epipolar line* (l), defined as intersection of the epipolar plane with the imaged plane. All epipolar lines intersect at the epipole.

Due to the epipolar constraint, the search for the correspondent point is thus restricted to the line $\boldsymbol{l}'$ and not extended at the entire image plane (Figure A.3).

It turns out that there exist a projective mapping from points to lines, which is represented by the Fundamental Matrix $\boldsymbol{F}$

$$\boldsymbol{l}' = \boldsymbol{F}\boldsymbol{x} \tag{A.3}$$

F is 3×3 matrix which encapsulates the epipolar geometry and whose main properties are outlined as follows

Result 3.4 *The Fundamental Matrix satisfies the condition that for any pair of corresponding points* $\boldsymbol{x} \longleftrightarrow \boldsymbol{x}'$ *in the two images*

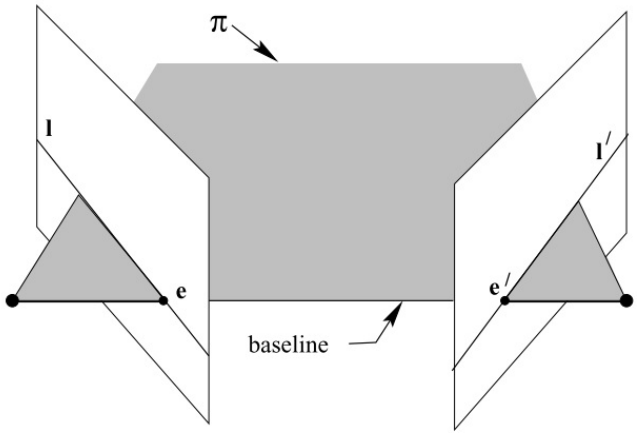

Figure A.2: Any plane $\boldsymbol{\pi}$ containing the baseline is an *epipolar plane* and intersects the image planes in the corresponding *epipolar lines* $\boldsymbol{l}$ and $\boldsymbol{l}'$. The camera *baseline* intersects each image at the epipoles $\boldsymbol{e}$ and $\boldsymbol{e}'$ [32]

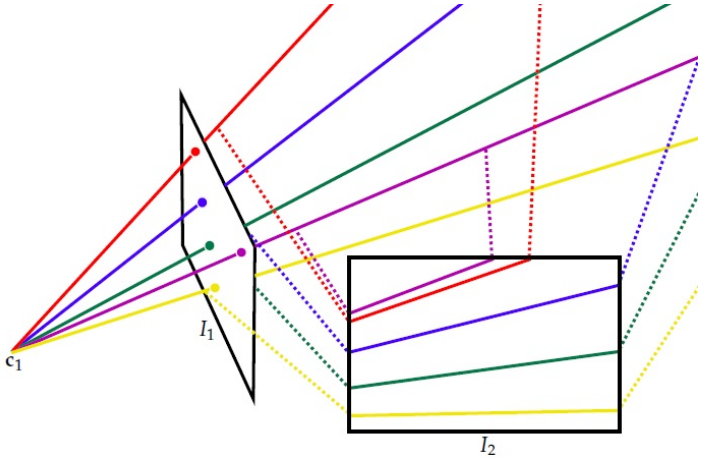

Figure A.3: Each point in I_1 is constrained to lie on its corresponding epipolar line in I_2 [20]. In presence of translation parallel to the image plane, and the rotation axis perpendicular to the image plane, a specific geometry of the epipolar line arises: the intersection between the baseline and the image plane is at infinity, therefore epipolar lines are parallel.

$$\boldsymbol{x}'^T \boldsymbol{F} \boldsymbol{x} = 0 \tag{A.4}$$

Proof
Points $\boldsymbol{x}$ and $\boldsymbol{x}'$ are corresponding, therefore the latter lies on the epipolar line $\boldsymbol{l}' = \boldsymbol{F}\boldsymbol{x}$. In other words, it is possible to write $0 = \boldsymbol{x}'^T \boldsymbol{l}' = \boldsymbol{x}'^T \boldsymbol{F}\boldsymbol{x}$.

Result 3.4 has characterized the fundamental matrix only in terms of correspondence points, thus $\boldsymbol{F}$ can be computed from image correspondences alone (Section **??**). Furthermore,

Result 3.5 *Given a Fundamental Matrix* $\boldsymbol{F}$*, the epipole* $\boldsymbol{e}'$ *is its left null-vector while* $\boldsymbol{e}$ *is right null-vector*

$$\boldsymbol{e}'\boldsymbol{F} = 0 \quad \boldsymbol{F}\boldsymbol{e} = 0 \tag{A.5}$$

Proof
For any point $\boldsymbol{x}$, other than the epipole, the epipolar line $\boldsymbol{l}' = \boldsymbol{F}\boldsymbol{x}$ contains the epipole $\boldsymbol{e}'$. In other words, we may write $\boldsymbol{e}'^T \boldsymbol{l}' = \boldsymbol{e}'^T (\boldsymbol{F}\boldsymbol{x}) = \left(\boldsymbol{e}'^T \boldsymbol{F}\right)\boldsymbol{x} = \boldsymbol{0}$. This holds for all $\boldsymbol{x}$, therefore $\boldsymbol{e}'\boldsymbol{F} = \boldsymbol{0}$. Similarly $F\boldsymbol{e} = \boldsymbol{0}$

The following result is frequently used

Result 3.6 *The Fundamental Matrix corresponding to a pair of camera matrices* $\boldsymbol{P} = [\boldsymbol{I}_{\boldsymbol{3\times 3}}|\boldsymbol{0}_{3\times 1}] = [\boldsymbol{I}|\boldsymbol{0}]$ *(canonical form) and* $\boldsymbol{P}' = [\boldsymbol{M}|\boldsymbol{m}]$ *is equal to*

$$\boldsymbol{F} = \boldsymbol{m} \times \boldsymbol{M} = [\boldsymbol{m}]_{\boldsymbol{x}} \boldsymbol{M} \tag{A.6}$$

In order to show that a given fundamental matrix determines the pair of camera matrices up to right multiplication by a projective transformation, we first introduce the following lemma

Lemma 3.1 *Suppose the rank 2 matrix* $\boldsymbol{F}$ *can be decomposed in two different ways as* $\boldsymbol{F} = [\boldsymbol{a}]_x \boldsymbol{A}$ *and* $\boldsymbol{F} = [\tilde{\boldsymbol{a}}]_x \tilde{\boldsymbol{A}}$*; then*

$$\tilde{\boldsymbol{a}} = k\boldsymbol{a} \quad \tilde{\boldsymbol{A}} = k^{-1}\left(\boldsymbol{A} + \boldsymbol{a}\boldsymbol{v}^T\right) \tag{A.7}$$

for some non-zero constant k and 3-vector $\boldsymbol{v}$

Proof
Note that $\boldsymbol{a}^T\boldsymbol{F} = \boldsymbol{a}^T [\boldsymbol{a}]_{\boldsymbol{x}} \boldsymbol{A} = \boldsymbol{a}^T (\boldsymbol{a} \times \boldsymbol{A}) = \boldsymbol{0}_{\boldsymbol{1\times 3}} = \boldsymbol{0}$. Similarly, $\tilde{\boldsymbol{a}}^T\boldsymbol{F} = \boldsymbol{0}$. Since $\boldsymbol{F}$ has rank 2 it follows that $\tilde{\boldsymbol{a}} = k\boldsymbol{a}$ as required. In light of the the previous result, $[\boldsymbol{a}]_x \boldsymbol{A} = [\tilde{\boldsymbol{a}}]_x \tilde{\boldsymbol{A}}$ can be further manipulated

$$[\tilde{\boldsymbol{a}}]_x \tilde{\boldsymbol{A}} - [\boldsymbol{a}]_x \boldsymbol{A} = k[\boldsymbol{a}]_x \tilde{\boldsymbol{A}} - [\boldsymbol{a}]_x \boldsymbol{A} = [\boldsymbol{a}]_x \left[k\tilde{\boldsymbol{A}} - \boldsymbol{A}\right] = 0$$

Therefore, it follows that $k\tilde{\boldsymbol{A}} - \boldsymbol{A} = \boldsymbol{a}\boldsymbol{v}^T$. Indeed

$$[\boldsymbol{a}]_x \boldsymbol{a}\boldsymbol{v}^T = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} a_1 v_1 & a_1 v_2 & a_1 v_3 \\ a_2 v_1 & a_2 v_2 & a_2 v_3 \\ a_3 v_1 & a_3 v_2 & a_3 v_3 \end{bmatrix} = 0_{3\times 3}$$

Therefore we have $k\tilde{\boldsymbol{A}} - \boldsymbol{A} = \boldsymbol{a}\boldsymbol{v}^T$. Writing this expression out it is obtained

$$\boldsymbol{k}\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{a}\boldsymbol{v}^{\boldsymbol{T}} \quad \rightarrow \quad \tilde{\boldsymbol{A}} = \boldsymbol{k}^{-\boldsymbol{1}}\left(\boldsymbol{A} + \boldsymbol{a}\boldsymbol{v}^{\boldsymbol{T}}\right)$$

We have now the necessary knowledge to prove the following theorem

Theorem 3.2

Let $\boldsymbol{F}$ *be a fundamental matrix and let* $\left(\boldsymbol{P}, \boldsymbol{P}'\right)$ *and* $\left(\tilde{\boldsymbol{P}}, \tilde{\boldsymbol{P}}'\right)$ *be two pairs of camera matrices such that F is the fundamental matrix corresponding to each of these pairs. Then there exists a non-singular 4×4 matrix H such that* $\tilde{\boldsymbol{P}} = \boldsymbol{P}\boldsymbol{H}$ *and* $\tilde{\boldsymbol{P}}' = \boldsymbol{P}'\boldsymbol{H}$.

Proof
Let's first simplify the problem, assuming that each of the two pair of camera matrices is in canonical form. This can be achieved by applying a projective transformations to each pair. Thus, $\boldsymbol{P} = \tilde{\boldsymbol{P}} = [\boldsymbol{I}|\boldsymbol{0}]$ and $\boldsymbol{P}' = [\boldsymbol{A}|\boldsymbol{a}]$ and $\tilde{\boldsymbol{P}}' = [\tilde{\boldsymbol{A}}|\tilde{\boldsymbol{a}}]$. According to Equation A.6 the fundamental matrix can then be written

$$\boldsymbol{F} = [\boldsymbol{a}]_x \boldsymbol{A} = [\tilde{\boldsymbol{a}}]_x \tilde{\boldsymbol{A}}$$

Applying Equation A.7 to the two camera matrices we obtain $\boldsymbol{P}' = [\boldsymbol{A}|\boldsymbol{a}]$ and $\tilde{\boldsymbol{P}}' = \left[k^{-1}\left(\boldsymbol{A} + \boldsymbol{a}\boldsymbol{v}^T\right)|k\boldsymbol{a}\right]$. In order to prove that these camera pairs is projectively related, consider the following matrix $\boldsymbol{H}$

$$\boldsymbol{H} = \begin{bmatrix} k^{-1}I_{3\times 3} & \boldsymbol{0}_{3\times 1} \\ k^{-1}\boldsymbol{v}^T & k \end{bmatrix} \tag{A.8}$$

It is possible to verify that $\boldsymbol{P}\boldsymbol{H} = k^{-1}\tilde{\boldsymbol{P}}$ besides, $\boldsymbol{P}'\boldsymbol{H} = \tilde{\boldsymbol{P}}'$

$$\boldsymbol{P}\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} k^{-1} & 0 & 0 & 0 \\ 0 & k^{-1} & 0 & 0 \\ 0 & 0 & k^{-1} & 0 \\ k^{-1}v_1 & k^{-1}v_2 & k^{-1}v_3 & k \end{bmatrix} =$$

$$\begin{bmatrix} k^{-1} & 0 & 0 & 0 \\ 0 & k^{-1} & 0 & 0 \\ 0 & 0 & k^{-1} & 0 \end{bmatrix} = k^{-1}[I|\boldsymbol{0}] = k^{-1}\tilde{P}$$

$$\boldsymbol{P}'\boldsymbol{H} = [\boldsymbol{A}|\boldsymbol{a}]\,\boldsymbol{H} =$$

$$\begin{bmatrix} A_{11}k^{-1} + k^{-1}v_1a_1 & A_{12}k^{-1} + k^{-1}v_1a_1 & A_{13}k^{-1} + k^{-1}v_1a_1 & a_1k \\ A_{21}k^{-1} + k^{-1}v_1a_1 & A_{22}k^{-1} + k^{-1}v_1a_1 & A_{23}k^{-1} + k^{-1}v_1a_1 & a_2k \\ A_{31}k^{-1} + k^{-1}v_1a_1 & A_{32}k^{-1} + k^{-1}v_1a_1 & A_{33}k^{-1} + k^{-1}v_1a_1 & a_3k \end{bmatrix}$$

$$= \left[k^{-1}\left(\boldsymbol{A} + \boldsymbol{a}\boldsymbol{v}^T\right)|k\boldsymbol{a}\right] = \left[\tilde{\boldsymbol{A}}|\tilde{\boldsymbol{a}}\right] = \tilde{\boldsymbol{P}}'$$

It has been proved that $\boldsymbol{P}$,$\boldsymbol{P}'$ and $\tilde{\boldsymbol{P}}$,$\tilde{\boldsymbol{P}}'$ are projectively related.
Through Result 3.7, a specific formula for a pair of cameras with canonical form will be derived

Result 3.7 *A non-zero matrix F is the fundamental matrix corresponding to a pair of camera matrices P and P' if and only if* $\boldsymbol{P}'^{\boldsymbol{T}}\boldsymbol{F}\boldsymbol{P}$ *is skew symmetric*

Proof
The condition that $\boldsymbol{P}'^{\boldsymbol{T}}\boldsymbol{F}\boldsymbol{P}$ is skew-symmetric is equivalent to $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{P}'^{\boldsymbol{T}}\boldsymbol{F}\boldsymbol{P}\boldsymbol{X} = 0$ for all $\boldsymbol{X}$. Substituting $\boldsymbol{x} = \boldsymbol{P}\boldsymbol{X}$ and $\boldsymbol{x}' = P'\boldsymbol{X}$, it is obtained $\boldsymbol{x}'^{\boldsymbol{T}}\boldsymbol{F}\boldsymbol{x} = 0$ which is the defining equation of the fundamental matrix and thus holds true.

Result 3.8 *Let* $\boldsymbol{F}$ *be a fundamental matrix and* $\boldsymbol{S}$ *any skew-symmetric matrix. Define the pair of camera matrices*

$$\boldsymbol{P} = [\boldsymbol{I}|\boldsymbol{0}] \quad \boldsymbol{P}' = \left[\boldsymbol{S}\boldsymbol{F}|\boldsymbol{e}'\right]$$

where $\boldsymbol{e}'$ *is the epipole such that* $\boldsymbol{e}'^{\boldsymbol{T}}\boldsymbol{F} = \boldsymbol{0}$. *Then* $\boldsymbol{F}$ *is the fundamental matrix corresponding to the pair* $\left(\boldsymbol{P}, \boldsymbol{P}'\right)$

Proof
Applying the previous result we only need to verify that $\boldsymbol{P'^T F P}$ is skew symmetric. Writing out this expression we obtain

$$\left[\boldsymbol{SF}|\boldsymbol{e'}\right]^T \boldsymbol{F}\,[I|\mathbf{0}] = \begin{bmatrix} \boldsymbol{F^T S^T} \\ \boldsymbol{e'}^T \end{bmatrix} [\boldsymbol{F}|\mathbf{0}] = \begin{bmatrix} \boldsymbol{F}^T\boldsymbol{SF} & 0_{3\times 1} \\ \boldsymbol{e'}^T\boldsymbol{F} & 0 \end{bmatrix}$$

Remembering that $\boldsymbol{e'}$ is the left null vector of $\boldsymbol{F}$ (Equation A.5), the only term left is taken into account

$$\boldsymbol{F}^T\boldsymbol{SF} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} 0 & s_2 & -s_3 \\ -s_2 & 0 & -s_1 \\ s_3 & s_1 & 0 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} = 0_{3\times 3}$$

A good choice for $\boldsymbol{S}$ is $\boldsymbol{S} = \left[\boldsymbol{e'}\right]_x$, which leads to the following useful result

Result 3.8 *The general formula for a pair of canonic camera matrices corresponding to a Fundamental Matrix $\boldsymbol{F}$ is given by*

$$\boldsymbol{P} = [\boldsymbol{I}|\mathbf{0}] \quad \boldsymbol{P'} = \left[\left[\boldsymbol{e'}\right]_x \boldsymbol{F} + \boldsymbol{e'}\boldsymbol{v}^T|\lambda\boldsymbol{e'}\right] \tag{A.9}$$

where $\boldsymbol{v}$ is any 3-vector and λ a non-zero scalar.

Appendix B

Homography H_K Derivation

In this Appendix, the derivation of the Equation 3.8 is presented.

Equation 3.7 can be re-written as follows

$$\boldsymbol{H_k} = \frac{\boldsymbol{x_1}}{\begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix}} \tag{B.1}$$

Through the camera matrix $\boldsymbol{M_1}$, the 2-D image point $\boldsymbol{x_1}$ is related to the 3-D world point $\boldsymbol{X}$

$$\boldsymbol{x_1} = \boldsymbol{M_1}\left(\boldsymbol{X} - \boldsymbol{C_1}\right) \tag{B.2}$$

Therefore, substituting it back to the previous equation we obtain

$$\boldsymbol{H_k} = \frac{\boldsymbol{M_1}\left(\vec{X} - \boldsymbol{C_1}\right)}{\begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix}} \tag{B.3}$$

$\vec{X}$ is replaced with the right term of Equation 3.3

$$\boldsymbol{H_k} = \frac{\boldsymbol{M_1}\left[s\boldsymbol{M_2^{-1}}\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} + \boldsymbol{C_2} - \boldsymbol{C_1}\right]}{\begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix}} = \frac{\boldsymbol{M_1}\left(\boldsymbol{C_2} - \boldsymbol{C_1}\right) + \boldsymbol{M_1}\boldsymbol{M_2^{-1}}s\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}}{\begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix}} \tag{B.4}$$

Substituting the expression for the scale factor s (Equation 3.5) the following is attained

$$\boldsymbol{H_k} = \frac{\boldsymbol{M_1}\left(\boldsymbol{C_2} - \boldsymbol{C_1}\right) - \boldsymbol{M_1}\boldsymbol{M_2^{-1}}\frac{\boldsymbol{N^T C_2} + z_k}{\boldsymbol{N^T M_2^{-1}}\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}}\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}}{\begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix}} \tag{B.5}$$

which can be re-arranged as follows

$$\boldsymbol{H_k} = \frac{\boldsymbol{M_1}\left(\boldsymbol{C_2}-\boldsymbol{C_1}\right)\boldsymbol{N^T}\boldsymbol{M_2^{-1}}\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix} - \boldsymbol{M_1}\boldsymbol{M_2^{-1}}\left[\boldsymbol{N^T}\boldsymbol{C_2}+z_k\right]\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}}{\boldsymbol{N^T}\boldsymbol{M_2^{-1}}\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}\begin{pmatrix} c_1 \\ r_1 \\ 1 \end{pmatrix}}$$

$$\boldsymbol{H_k} = \frac{\boldsymbol{M_1}\left(\vec{C}_2-\boldsymbol{C_1}\right)\boldsymbol{N^T}\boldsymbol{M_2^{-1}} - \boldsymbol{M_1}\boldsymbol{M_2^{-1}}\left(\boldsymbol{N^T}\vec{C}_2+z_k\right)}{\boldsymbol{N^T}\boldsymbol{M_2^{-1}}\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}} \tag{B.6}$$

Since the homography $\boldsymbol{H_k}$ may be changed multiplying by an arbitrary non-zero scale factor, without altering the projective transformation, the scalar term $\vec{N}^T\boldsymbol{M_2^{-1}}\begin{pmatrix} c_2 \\ r_2 \\ 1 \end{pmatrix}$ at the denominator can thus be omitted. Moreover, since the quantity $\boldsymbol{N^T}\vec{C}_2+z_k$ is a scalar, it can be written to the left side of factor $\boldsymbol{M_1}\boldsymbol{M_2^{-1}}$ obtaining

$$\boldsymbol{H_k} = \boldsymbol{M_1}\left(\vec{C}_2-\vec{C}_1\right)\vec{N}^T\boldsymbol{M_2^{-1}} - \left(\boldsymbol{N^T}\vec{C}_2-z_k\right)\boldsymbol{M_1}\boldsymbol{M_2^{-1}} \tag{B.7}$$

Appendix C

Harris Corner Detector

This appendix introduces one of the earliest key-point detector, namely *Harris Corner detector*. Presented by Harris and Stephens in 1988 [10], it is still widely employed nowadays [20] due to its simplicity.

To describe its working principle, let us now consider a single image: we would like to assess how do pixels in W change by shifting the kernel window by a vector $\Delta\vec{x} = (\Delta x, \Delta y)$ (Figure C.1).

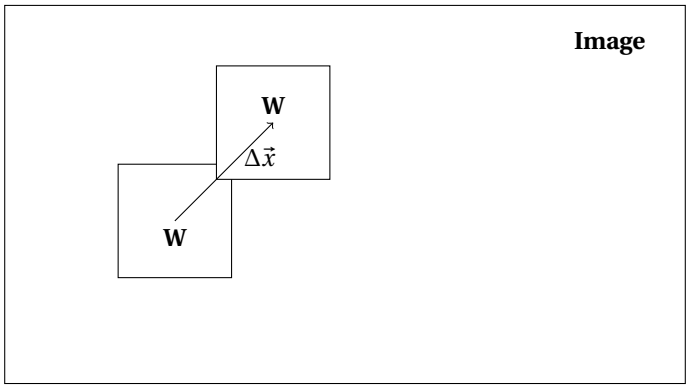

Figure C.1: The Kernel W (2N+1×2N+1) is shifted by a vector $(\Delta x, \Delta y)$ within the image boundaries

Each pixel is compared before and after through the sum of the squared differences (SSD)

$$SSD_{(W)} = \sum_{x=-N}^{N} \sum_{y=-N}^{N} \left[I(x+\Delta x, y+\Delta y) - I(x,y) \right]^2 \tag{C.1}$$

Considering a Taylor Series expansion of $I(x+\Delta x, y+\Delta y)$ and assuming that $\Delta\vec{x}$ is small, the following first order approximation holds

$$I(x+\Delta x, y+\Delta y) \approx I(x,y) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y \tag{C.2}$$

Substituting the previous relation back into Equation C.1 yields

$$SSD_{(W)} = \sum_{x=-N}^{N} \sum_{y=-N}^{N} \left[I(x,y) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y - I(x,y) \right]^2 = \sum_{x=-N}^{N} \sum_{y=-N}^{N} \left[\begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right]^2$$

Writing the expression out, it is obtained

$$SSD_{(W)} = \sum_{x=-N}^{N} \sum_{y=-N}^{N} \left[\left(\frac{\partial I}{\partial x}\right)^2 \Delta x^2 + 2\left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right)\Delta x \Delta y + \left(\frac{\partial I}{\partial x}\right)^2 \Delta x^2 \right] \tag{C.3}$$

Equation C.3 can be further manipulated yielding

$$SSD_{(W)} = \sum_{x=-N}^{N}\sum_{y=-N}^{N} \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$= \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \sum_{x=-N}^{N}\sum_{y=-N}^{N} \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \Delta \boldsymbol{x}^{\boldsymbol{T}} M \Delta \boldsymbol{x} \tag{C.4}$$

where $\boldsymbol{M}$ is called *second moment matrix*.
To detect features, the $\boldsymbol{M}$ matrix has to be computed for each pixel of W. To ensure that no displacement exists for which the SSD value is small, the eigenvalues of $\boldsymbol{M}$ should be both large (Figure C.2).

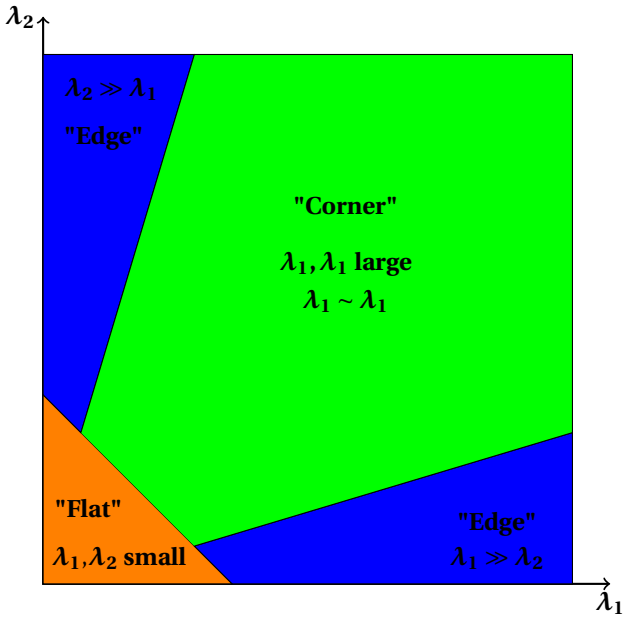

Figure C.2: Classification of image pixels according to the eigenvalues of M

This can be either achieved by enforcing a minimal value for the smallest eigenvalue or computing the *corner response function* [10]

$$R = \lambda_1 \lambda_2 - \alpha (\lambda_1 + \lambda_2)^2 \tag{C.5}$$

with α a constant whose value ranges from 0.04 to 0.06. In their original paper, Harris and Stephens considered the exact computation of the eigenvalues to be too expensive and instead suggested the following response function

$$R = \det(M) - \alpha \mathrm{Tr}(M) \tag{C.6}$$

where Tr represents the trace of the matrix (e.g. the sum of its diagonal terms). The next step consists to identify points with large corner response (R > threshold). The so-called feature points are represented by local maxima of R.

Appendix D

Parametric Analysis - Input Images

The appendix hereby displays the input image pairs employed for the assessment of the optimal correlation window W and depth step size Δz (Section 4.3).

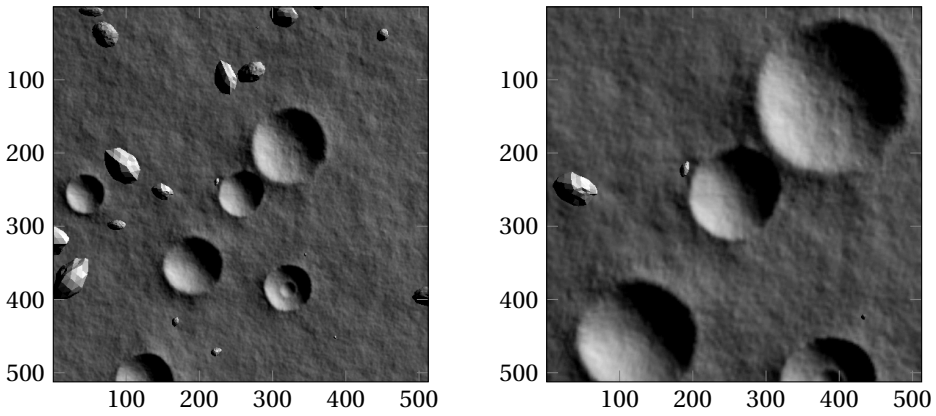

Figure D.1: Scene 1, image I_1 at 150m elevation (left) and I_2 (right) taken when the altitude is halved

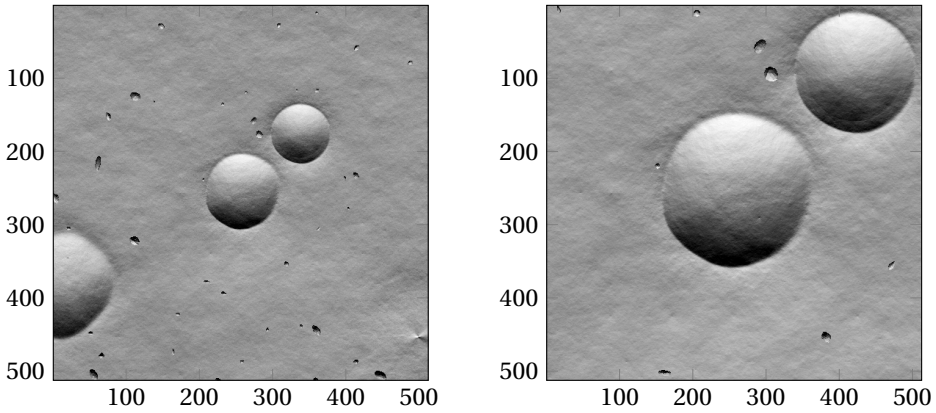

Figure D.2: Scene 2, image I_1 at 600m elevation (left) and I_2 (right) taken when the altitude is halved

Appendix E

Least-Square Minimization through SVD

In this appendix it is discussed how to solve a linear system of equations through SVD. Its drafting is fully based on [32].

Consider a system of equations expressed in the form $\boldsymbol{Ax} = \boldsymbol{b}$. Let $\boldsymbol{A}$ be an $m \times n$ matrix (with $m > n$) of rank n. In general this system will not have a solution, unless $\boldsymbol{b}$ lies in the span of the columns of $\boldsymbol{A}$. Therefore, it makes sense to seek a vector $\boldsymbol{x}$ which is closest to provide a solution to the system: in other words, we seek $\boldsymbol{x}$ such that the norm $||\boldsymbol{Ax} - \boldsymbol{b}||$ is minimized. The vector $\boldsymbol{x}$ is known as the *least-square solution*, and it is conveniently found employing the SVD. Given the previously defined matrix $\boldsymbol{A}$, through SVD it can be factorized as follows

$$\boldsymbol{A} = \boldsymbol{UDV}^{\boldsymbol{T}} \tag{E.1}$$

where $\boldsymbol{U}$ is an $m \times n$ matrix with orthogonal columns, $\boldsymbol{D}$ is an $n \times n$ diagonal matrix, and $\boldsymbol{V}$ is an $n \times n$ orthogonal matrix. Thus, applying Equation E.1 the following holds

$$||\boldsymbol{Ax} - \boldsymbol{b}|| = ||\boldsymbol{UDV}^{\boldsymbol{T}} - \boldsymbol{b}|| \tag{E.2}$$

Due to the norm-preserving property $||\boldsymbol{Ux}|| = ||\boldsymbol{x}||$ for any vector $\boldsymbol{x}$, the previous equation can be re-written as

$$||\boldsymbol{UDV}^{\boldsymbol{T}}\boldsymbol{x} - \boldsymbol{b}|| = ||\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{UDV}^{\boldsymbol{T}}\boldsymbol{x} - \boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{b}|| \tag{E.3}$$

Furthermore, since $\boldsymbol{U}$ has orthogonal columns $\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{U} = \boldsymbol{I}_{\boldsymbol{n}\times\boldsymbol{n}}$. Therefore, the quantity we want to minimize is

$$||\boldsymbol{DV}^{\boldsymbol{T}}\boldsymbol{x} - \boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{b}|| \tag{E.4}$$

Writing $\boldsymbol{y} = \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{x}$ and $\boldsymbol{b}' = \boldsymbol{U}^{\boldsymbol{T}}$, the problem becomes one of minimizing $||\boldsymbol{Dy} - \boldsymbol{b}'||$

$$\left[\begin{array}{cccc} d_1 & & & \\ & d_2 & & \\ & & \cdots & \\ & & & d_n \\ \hline & & 0 & \end{array}\right] \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \left(\begin{array}{c} b'_1 \\ b'_2 \\ \vdots \\ b'_n \\ \hline b'_{n+1} \\ \vdots \\ b'_m \end{array}\right) \tag{E.5}$$

It is straightforward that the nearest $\boldsymbol{Dy}$ that approaches $\boldsymbol{b}'$ is the vector $\left(b'_1, b'_2, \ldots, b'_n, 0, \ldots, 0\right)^T$. This is achieved by setting $y_i = b'_i / d_i$ for $i = 1, \vdots, n$. Note that the assumption rank$(\boldsymbol{A}) = n$ ensures that $d_i \neq 0$. Finally, the solution $\boldsymbol{x}$ is retrieved by

$$\boldsymbol{x} = \boldsymbol{V}\boldsymbol{y} \tag{E.6}$$

For the sake of a direct computer implementation, the algorithm can be sum up to only 4 simple steps

(i) Find the SVD $\rightarrow \boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}}$

(ii) Set $\boldsymbol{b}' = \boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{b}$

(iii) Find the vector $\boldsymbol{y}$ defined by $y_i = b'_i / d_i$, where d_i is the i-th diagonal entry of $\boldsymbol{D}$

(iv) The solution is $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{y}$

Appendix F

Mission Scenario 1 - Input Image Pairs

The appendix hereby displays the input descent sequence employed in the mission scenario 1 (Section 6.1).

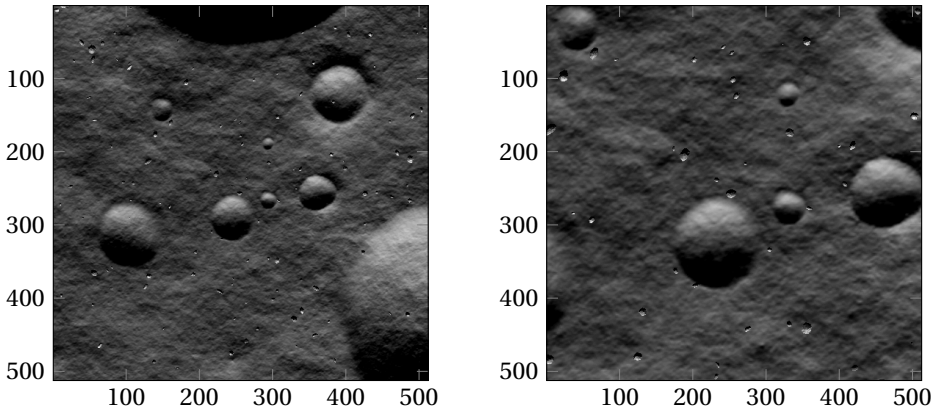

Figure F.1: Scene 1, image I_1 at 1000m elevation (left) and I_2 (right) taken when the altitude is halved

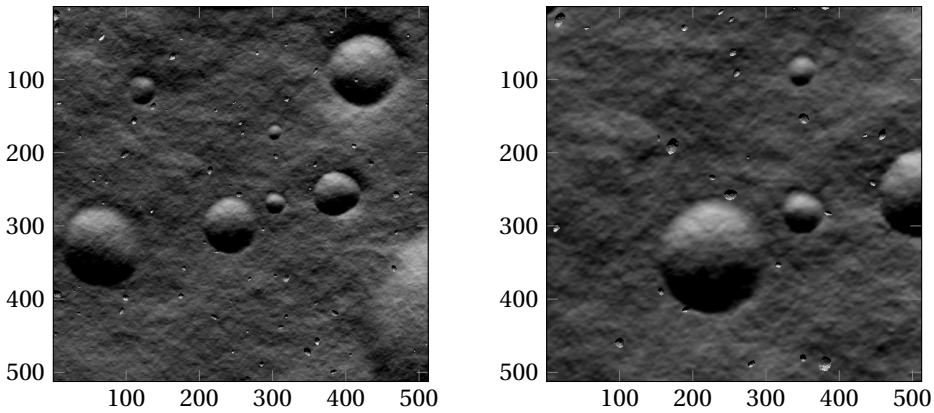

Figure F.2: Scene 2, image I_1 at 800m elevation (left) and I_2 (right) taken when the altitude is halved

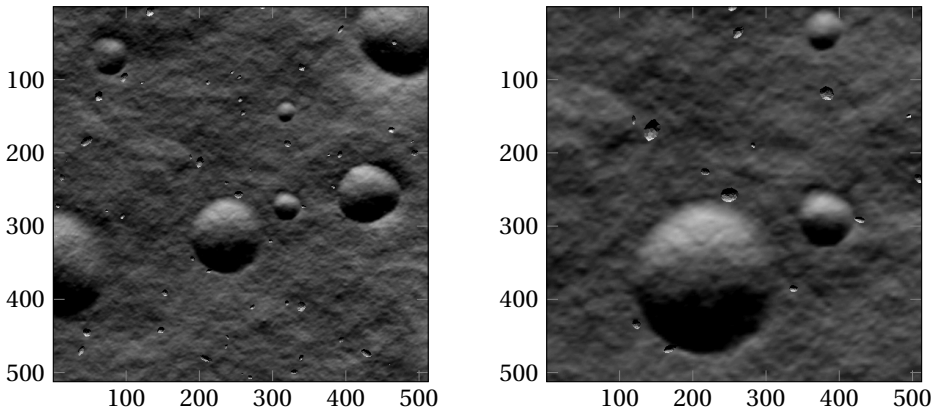

Figure F.3: Scene 3, image I_1 at 600m elevation (left) and I_2 (right) taken when the altitude is halved

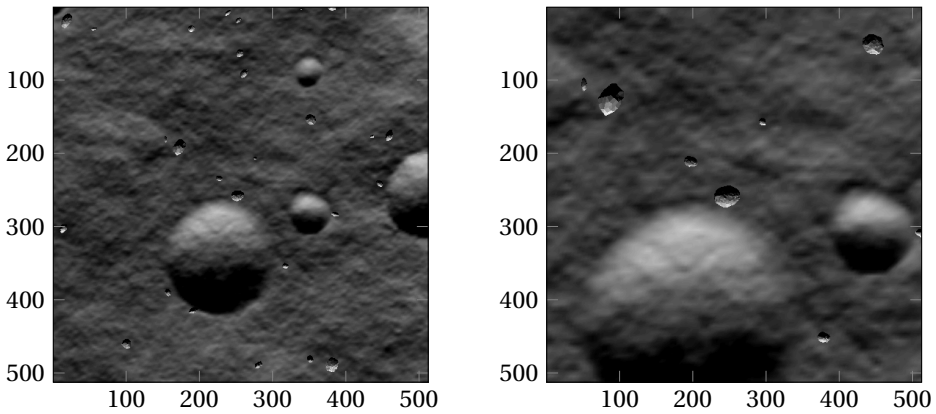

Figure F.4: Scene 4, image I_1 at 400m elevation (left) and I_2 (right) taken when the altitude is halved

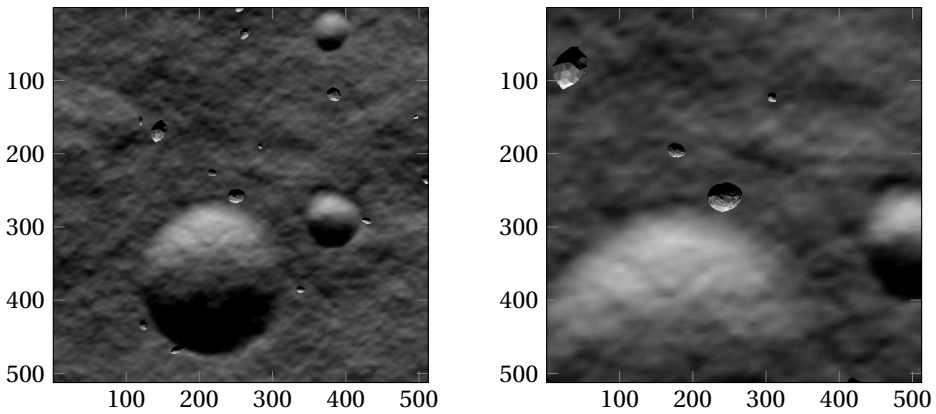

Figure F.5: Scene 5, image I_1 at 300m elevation (left) and I_2 (right) taken when the altitude is halved

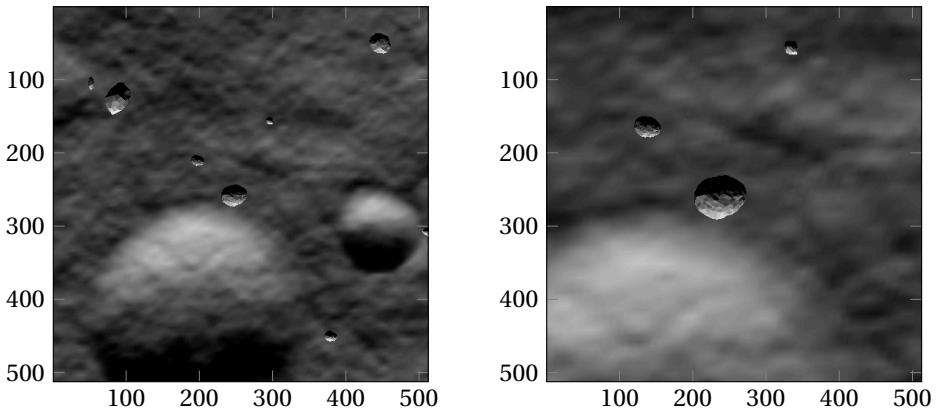

Figure F.6: Scene 6, image I_1 at 200m elevation (left) and I_2 (right) taken when the altitude is halved

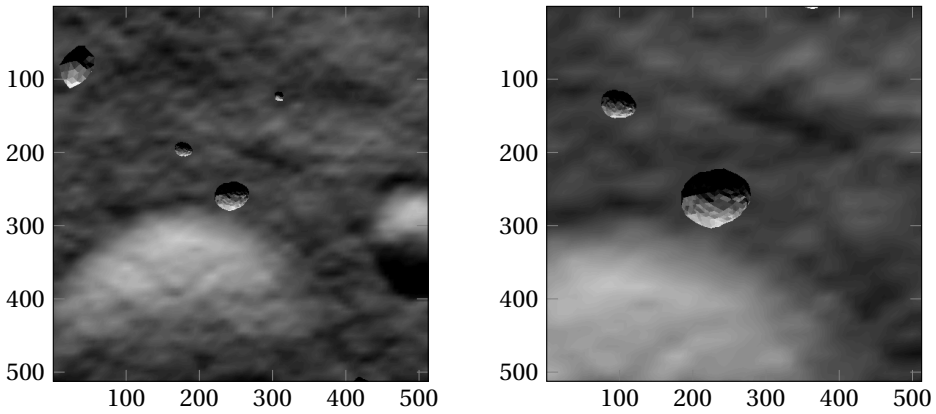

Figure F.7: Scene 7, image I_1 at 150m elevation (left) and I_2 (right) taken when the altitude is halved

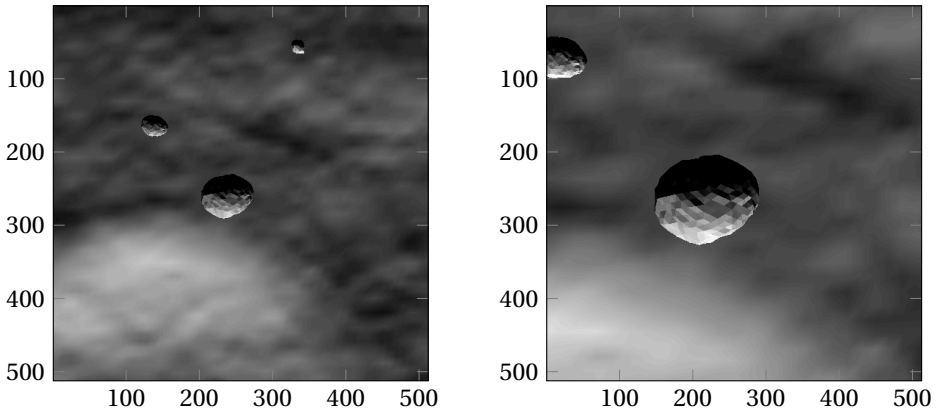

Figure F.8: Scene 8, image I_1 at 100m elevation (left) and I_2 (right) taken when the altitude is halved

Appendix G

Mission Scenario 1 - DEM

This appendix shows the shape recovery results for the mission scenario 1 (Section 6.1). The computed DEMs are displayed together with their error maps and ground-truths.

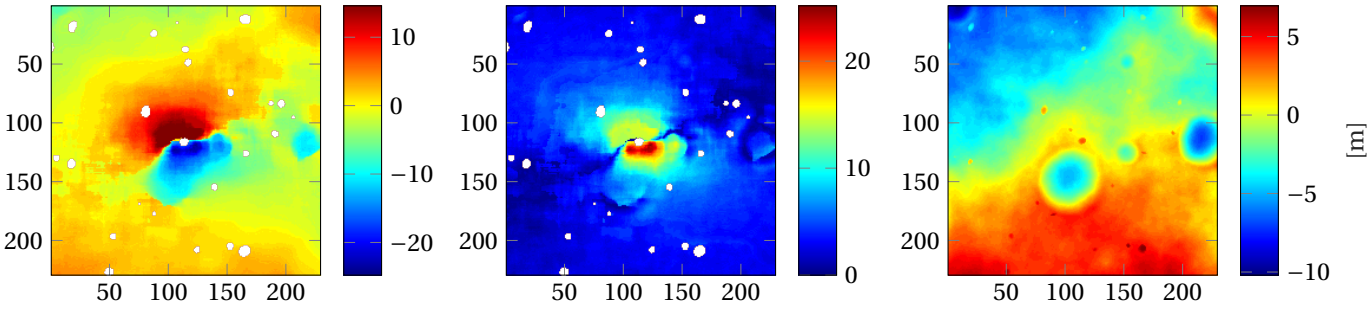

Figure G.1: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 500 m

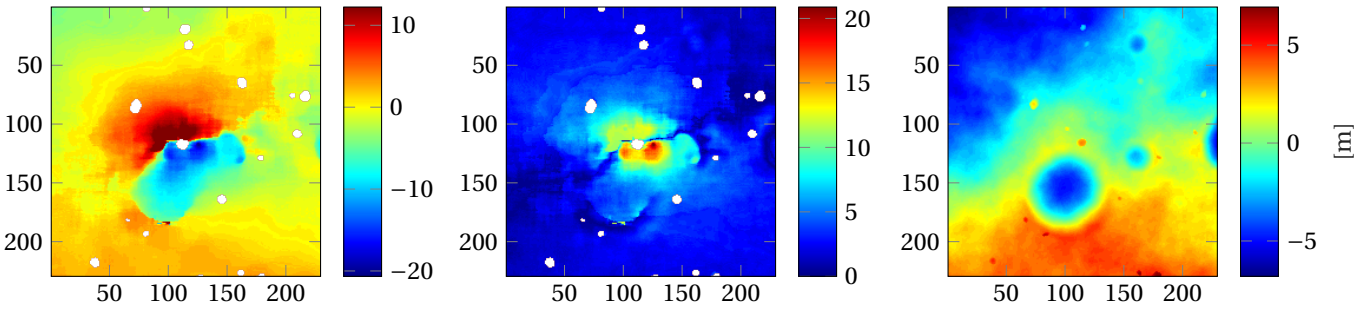

Figure G.2: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 400 m

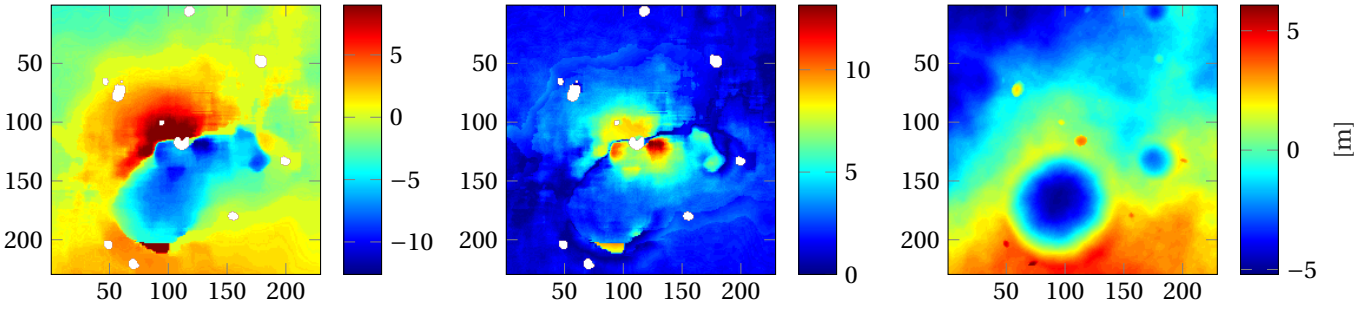

Figure G.3: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 300 m

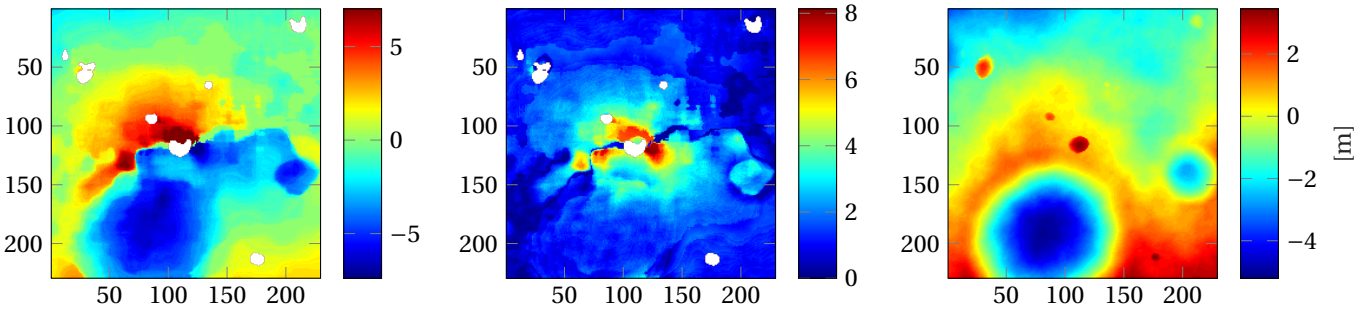

Figure G.4: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 200 m

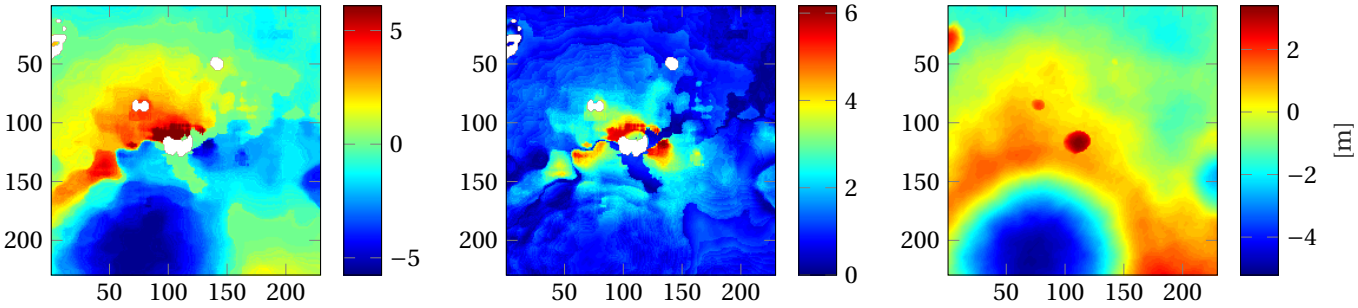

Figure G.5: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 150 m

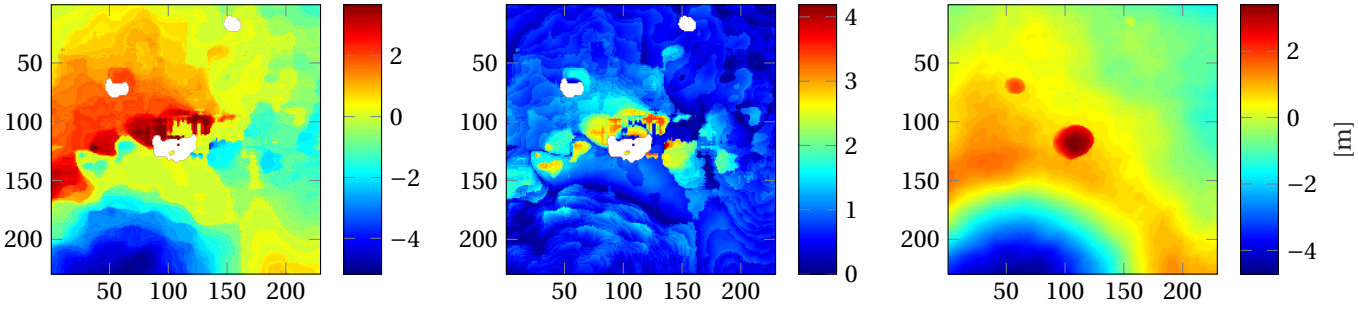

Figure G.6: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 100 m

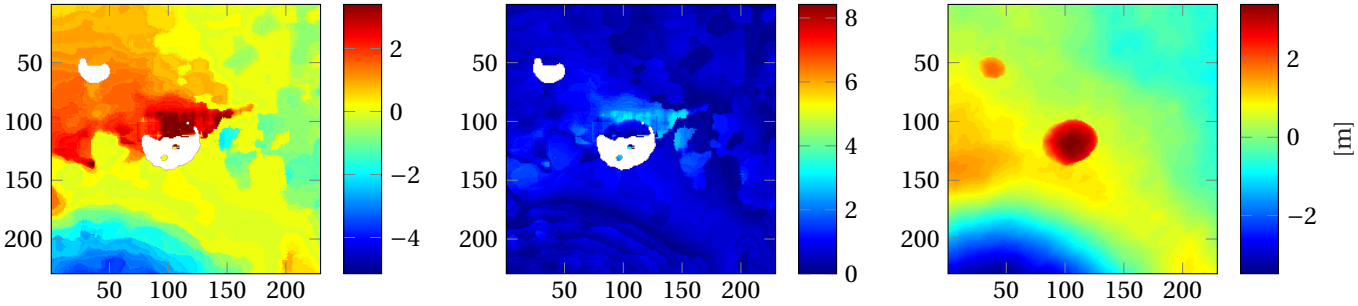

Figure G.7: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 75 m

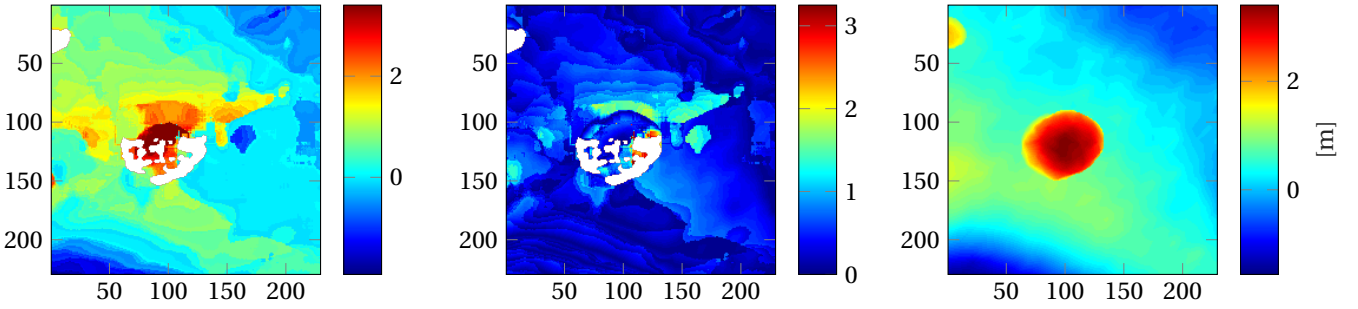

Figure G.8: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 50 m

Appendix H

Mission Scenario 1 - Final Hazard Maps

The appendix hereby displays the hazard maps for the mission scenario 1 (Section 6.1). Also the related hazard-error maps, indicating the location of the undetected hazards (dark-gray regions) and false alarms (light-gray regions), are present.

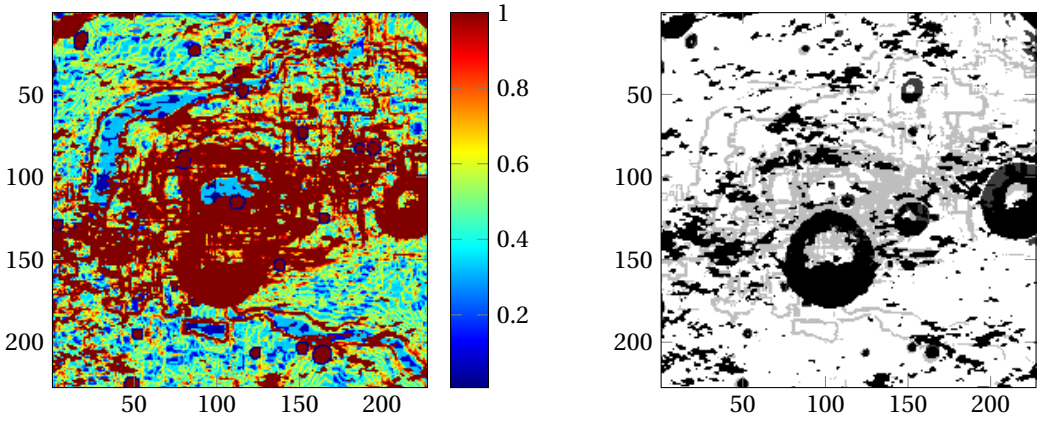

Figure H.1: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 500 m

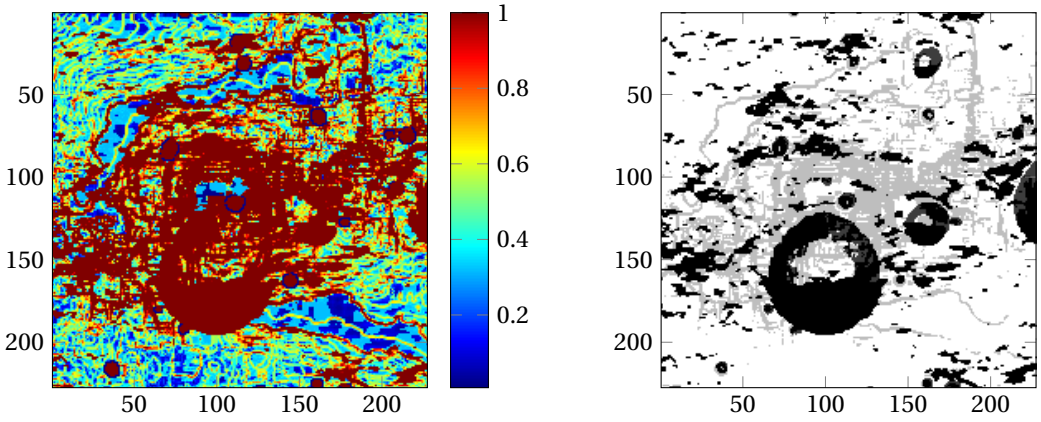

Figure H.2: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 400 m

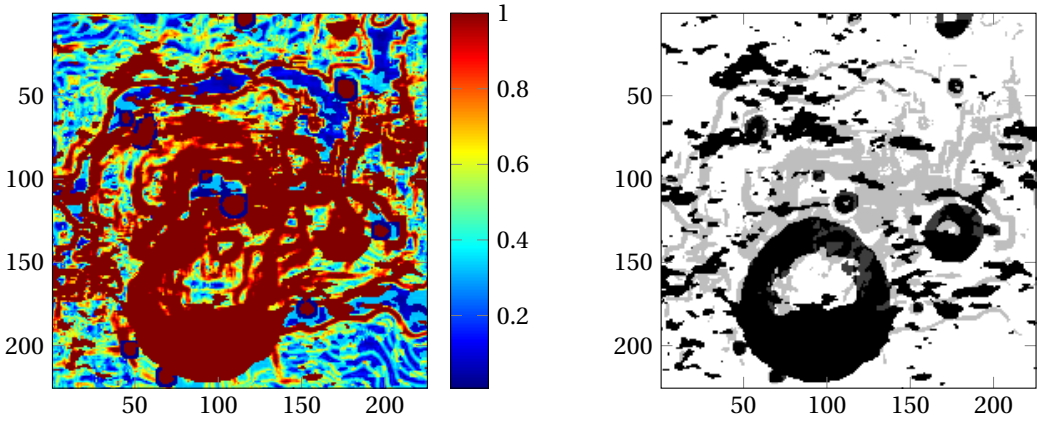

Figure H.3: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 300 m

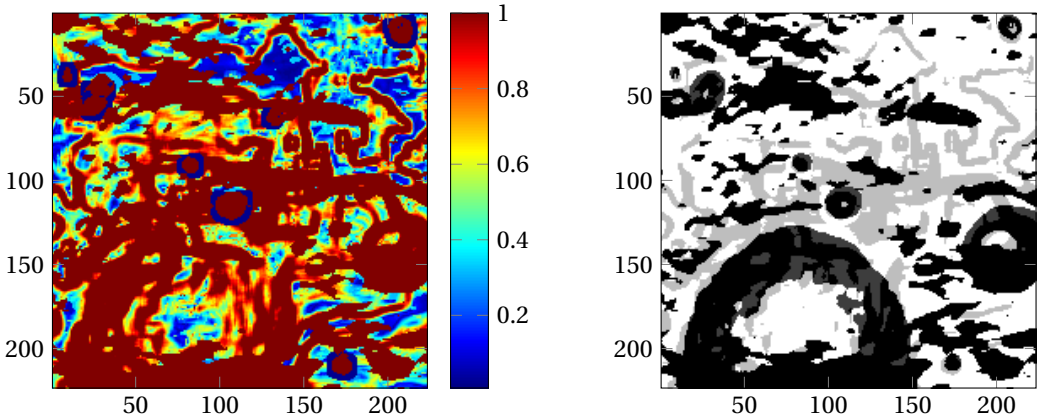

Figure H.4: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 200 m

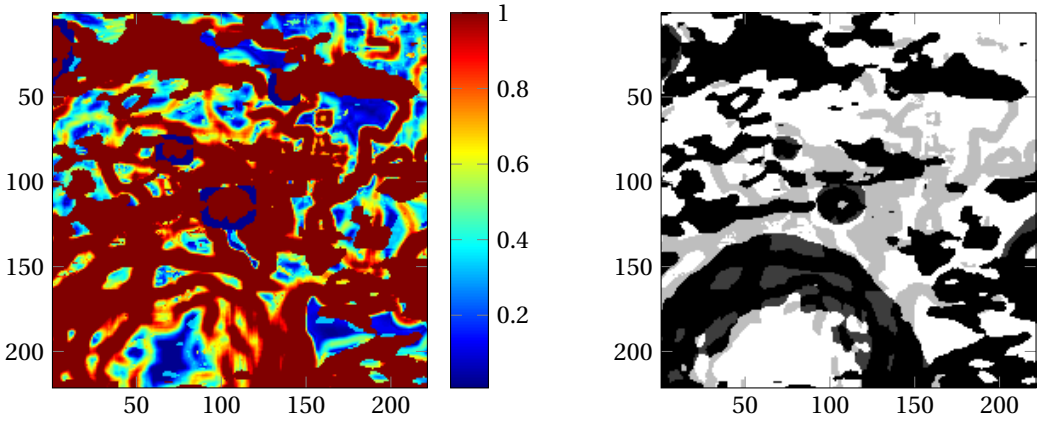

Figure H.5: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 150 m

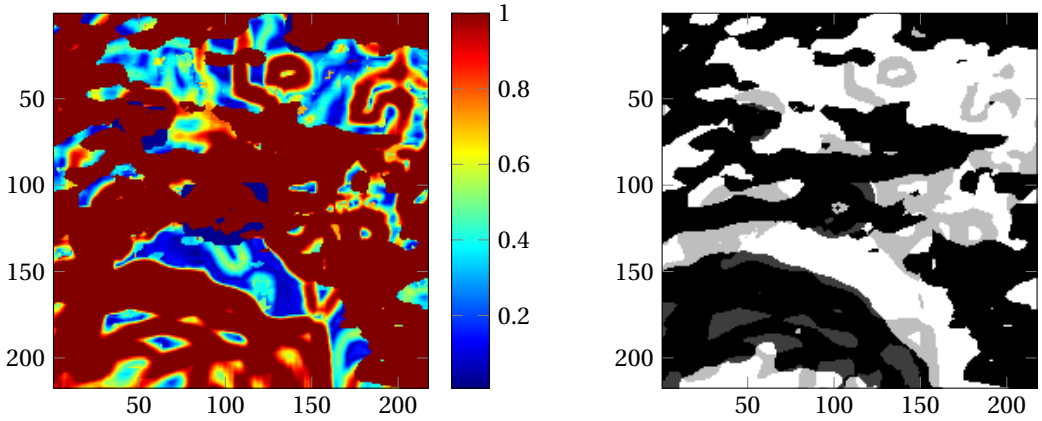

Figure H.6: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 100 m

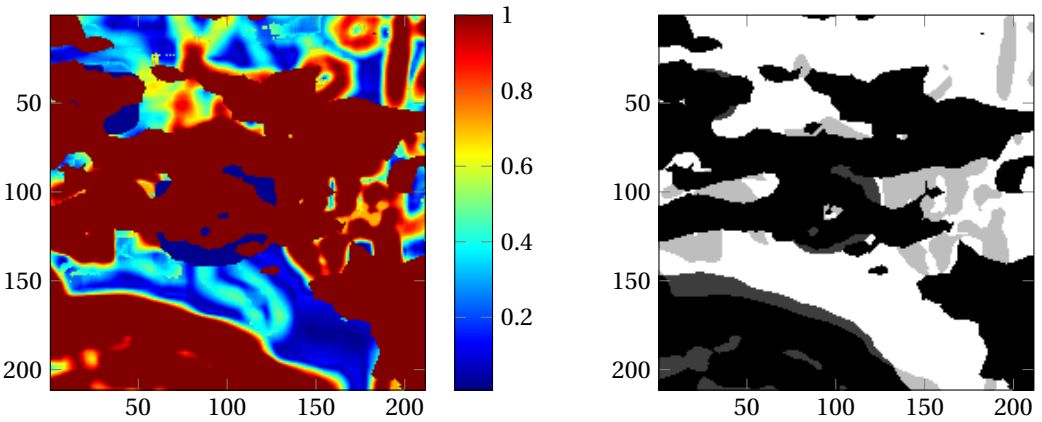

Figure H.7: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 75 m

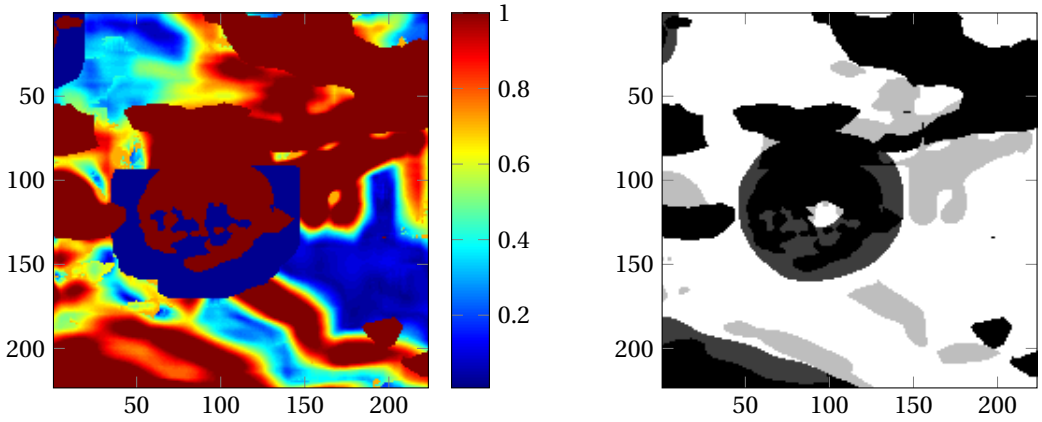

Figure H.8: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 50 m

Appendix I

Mission Scenario 2 - Input Image Pairs

The appendix hereby displays the input descent sequence employed in the mission scenario 2 (Section 6.2).

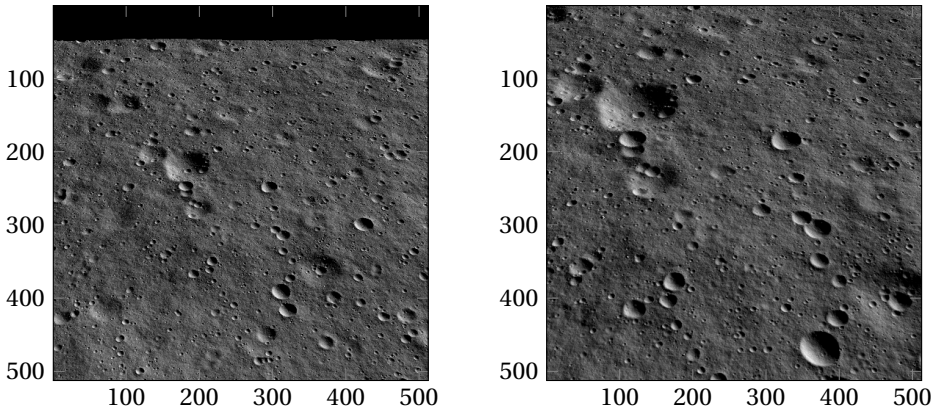

Figure I.1: Scene 8, image I_1 at 2400 m elevation (left) and I_2 (right) taken when the altitude is halved

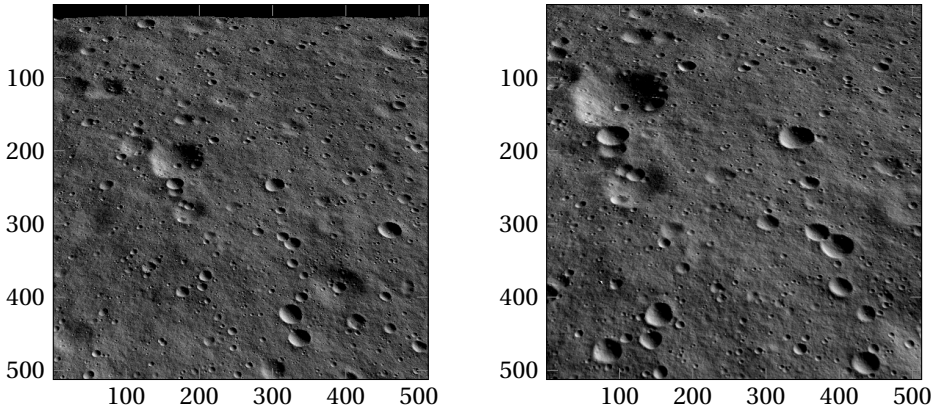

Figure I.2: Scene 7, image I_1 at 2000 m elevation (left) and I_2 (right) taken when the altitude is halved

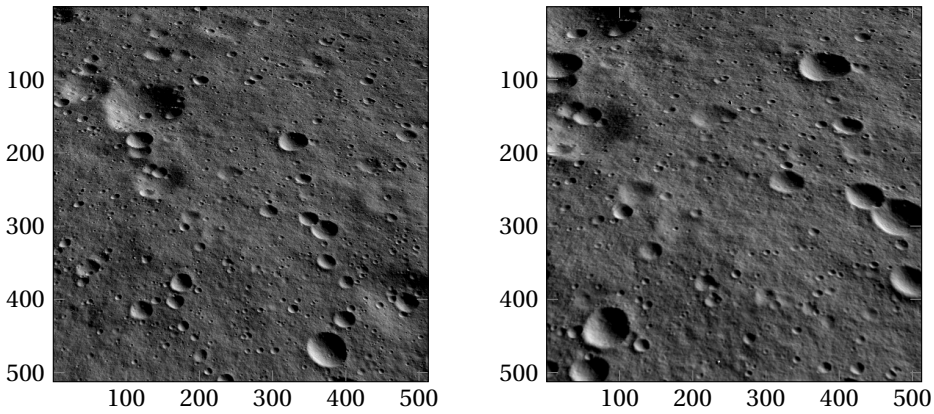

Figure I.3: Scene 6, image I_1 at 1200 m elevation (left) and I_2 (right) taken when the altitude is halved

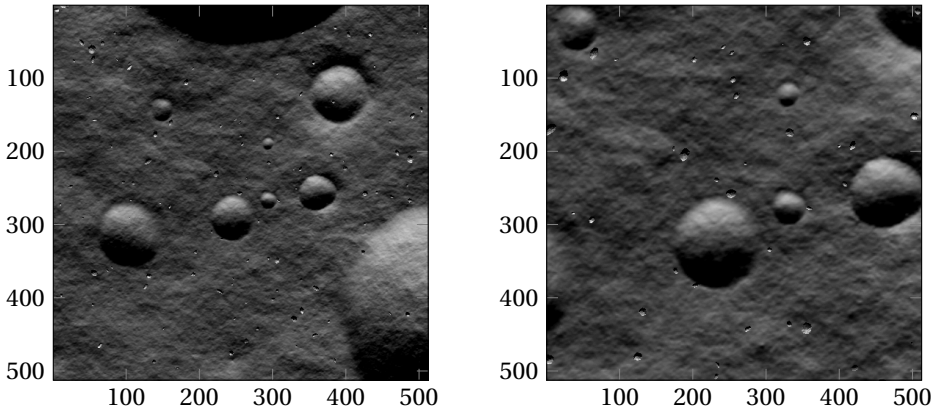

Figure I.4: Scene 5, image I_1 at 1000 m elevation (left) and I_2 (right) taken when the altitude is halved

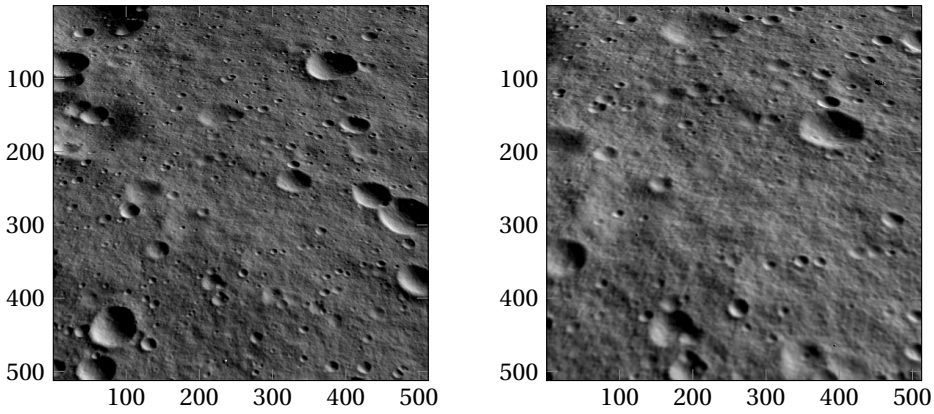

Figure I.5: Scene 4, image I_1 at 600 m elevation (left) and I_2 (right) taken when the altitude is halved

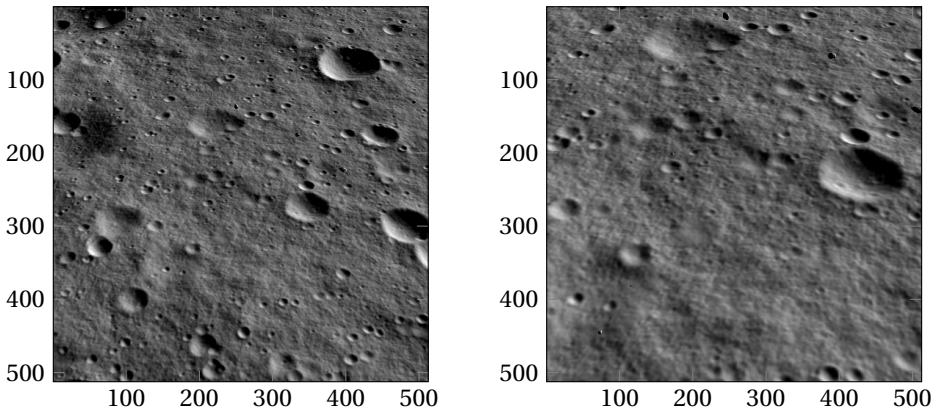

Figure I.6: Scene 3, image I_1 at 500 m elevation (left) and I_2 (right) taken when the altitude is halved

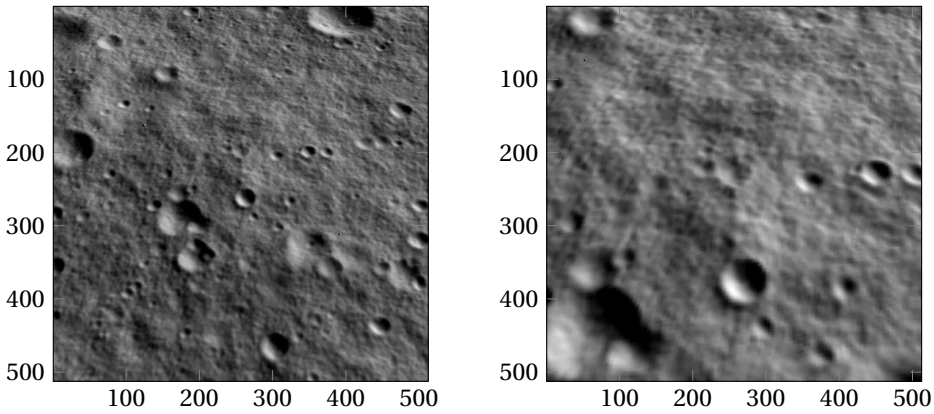

Figure I.7: Scene 2, image I_1 at 300 m elevation (left) and I_2 (right) taken when the altitude is halved

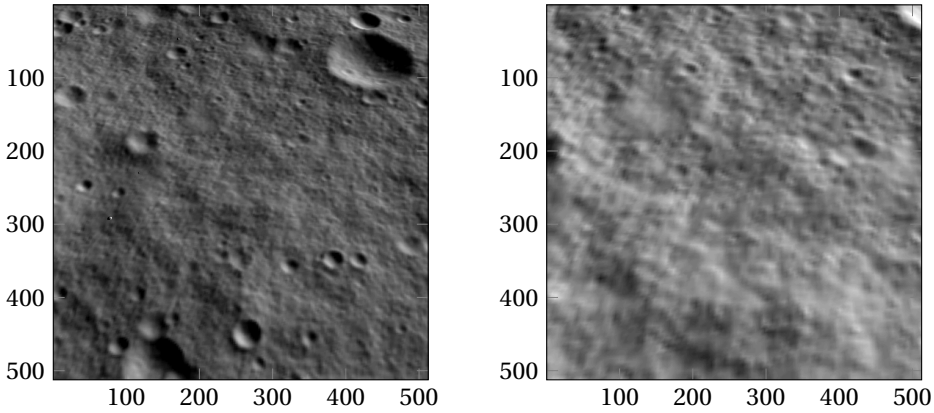

Figure I.8: Scene 1, image I_1 at 250 m elevation (left) and I_2 (right) taken when the altitude is halved

Appendix J

Mission Scenario 2 - DEM

This appendix shows the shape recovery results for the mission scenario 2 (Section 6.2). The computed DEMs are displayed together with their error maps and ground-truths.

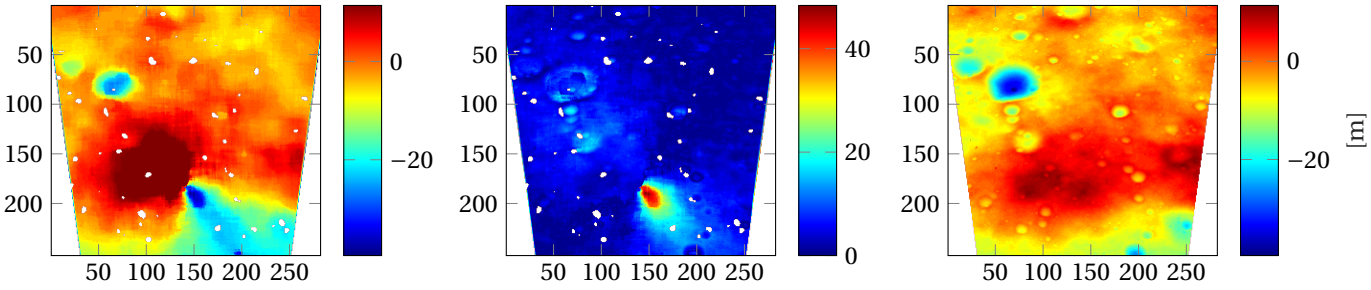

Figure J.1: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 1200 m

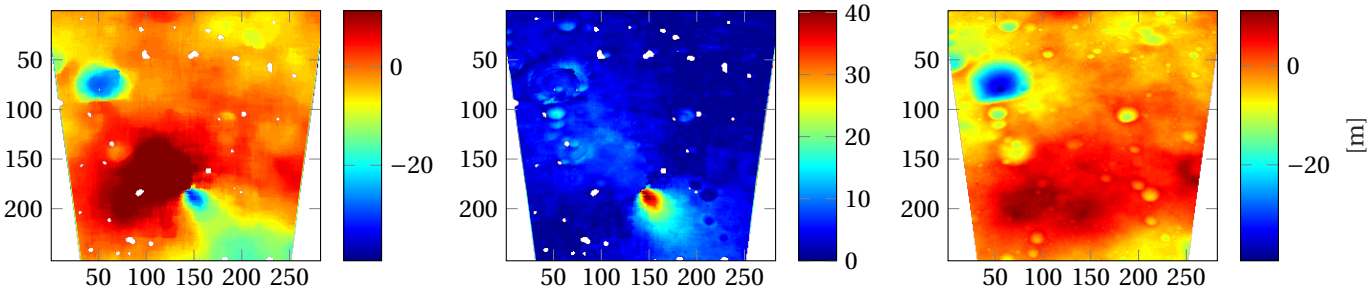

Figure J.2: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 1000 m

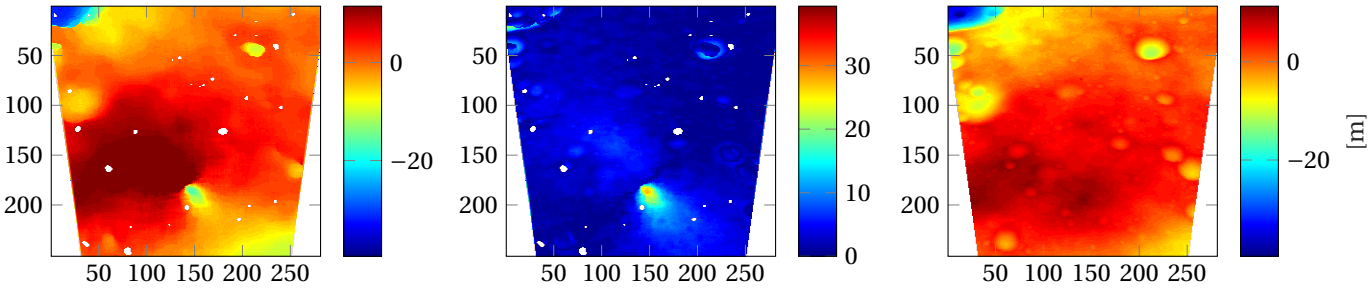

Figure J.3: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 600 m

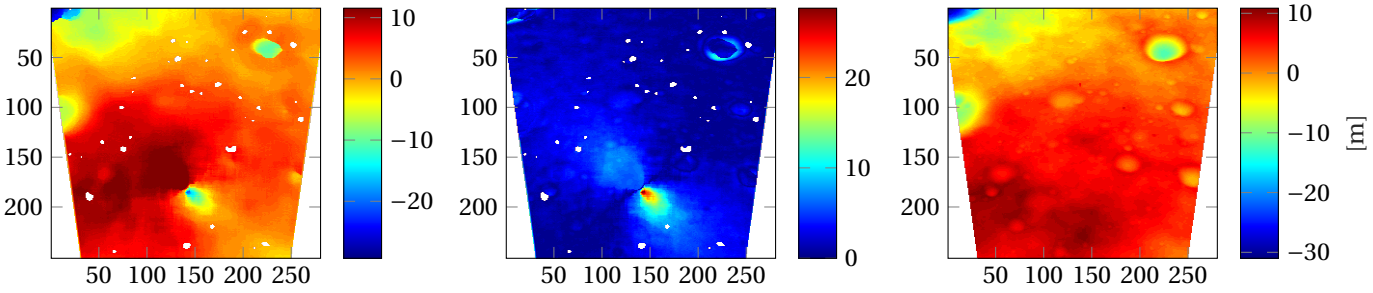

Figure J.4: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 500 m

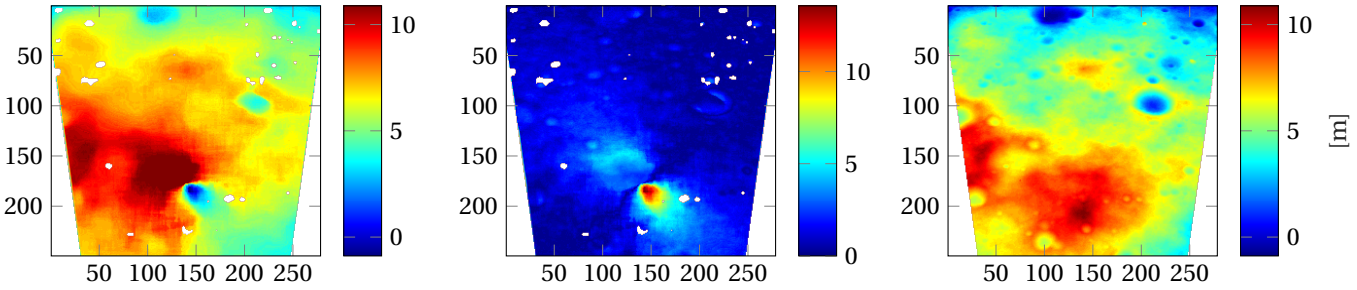

Figure J.5: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 300 m

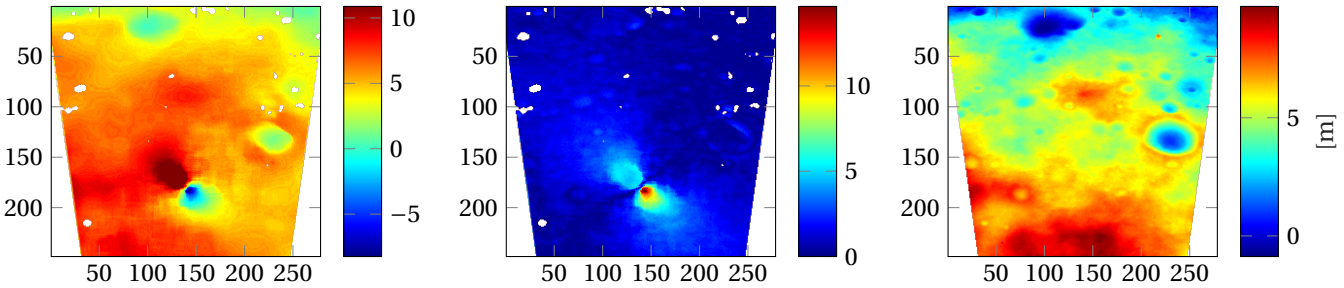

Figure J.6: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 250 m

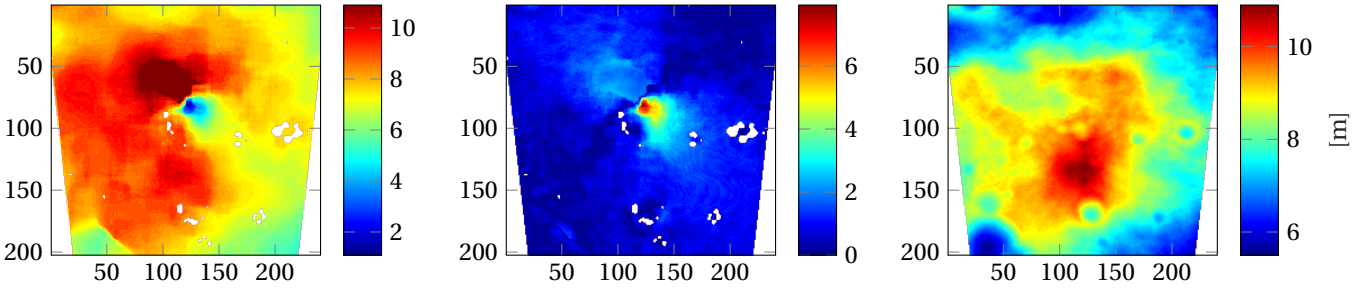

Figure J.7: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 150 m

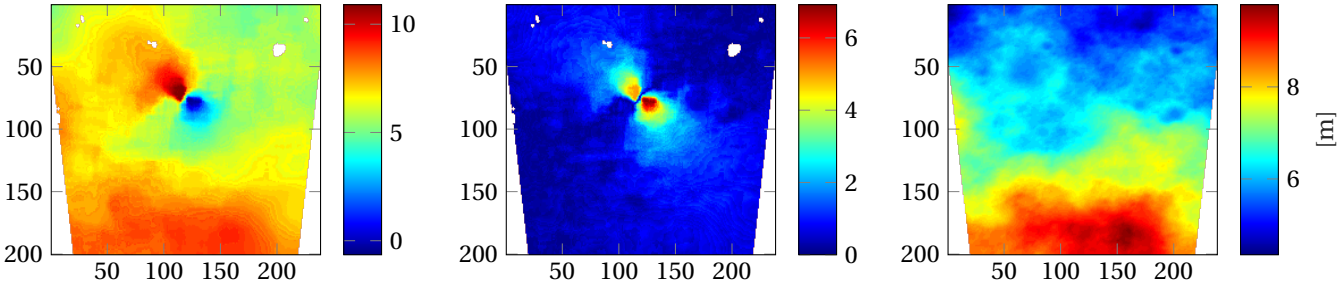

Figure J.8: Computed DEM (left) DEM error (center) and ground-truth (right). Altitude h = 125 m

Appendix K

Mission Scenario 2 - Final Hazard Maps

The appendix hereby displays the hazard maps for the mission scenario 2 (Section 6.2). Also the related hazard-error maps, indicating the location of the undetected hazards (dark-gray regions) and false alarms (light-gray regions), are present.

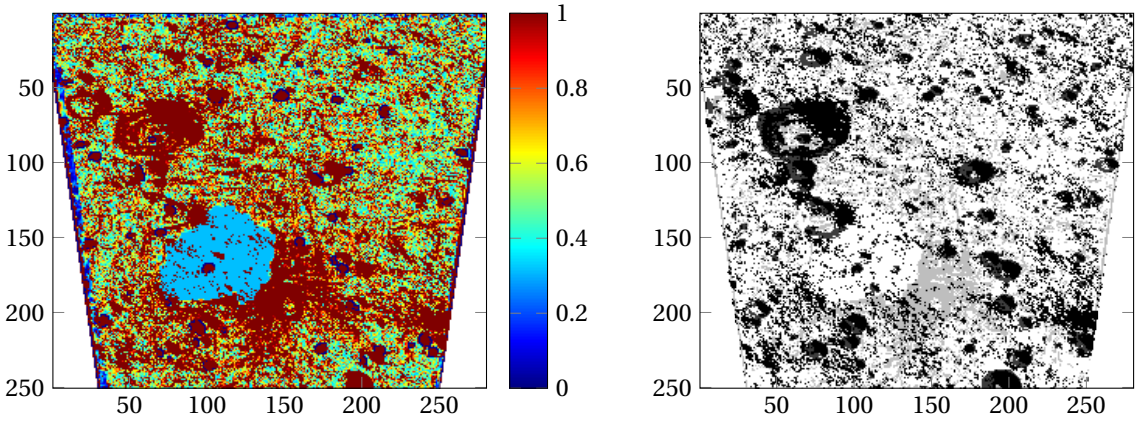

Figure K.1: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 1200 m

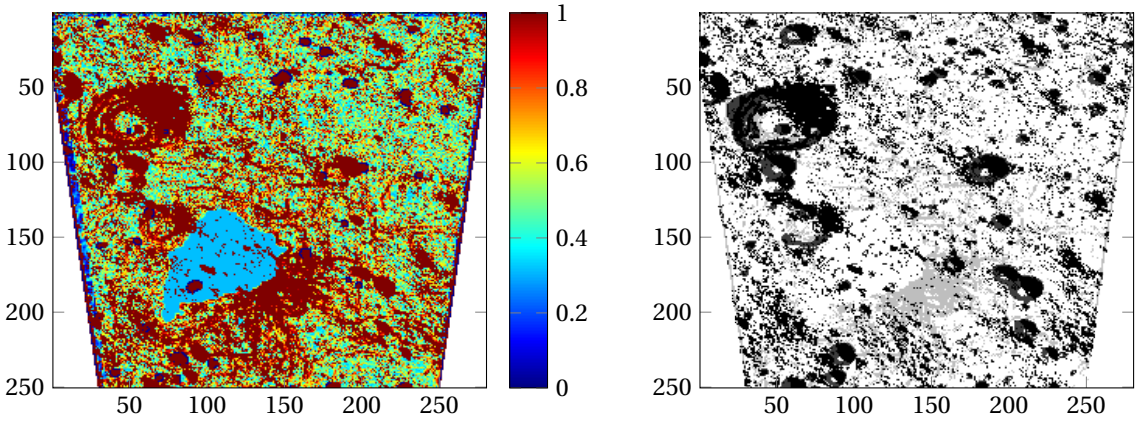

Figure K.2: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 1000 m

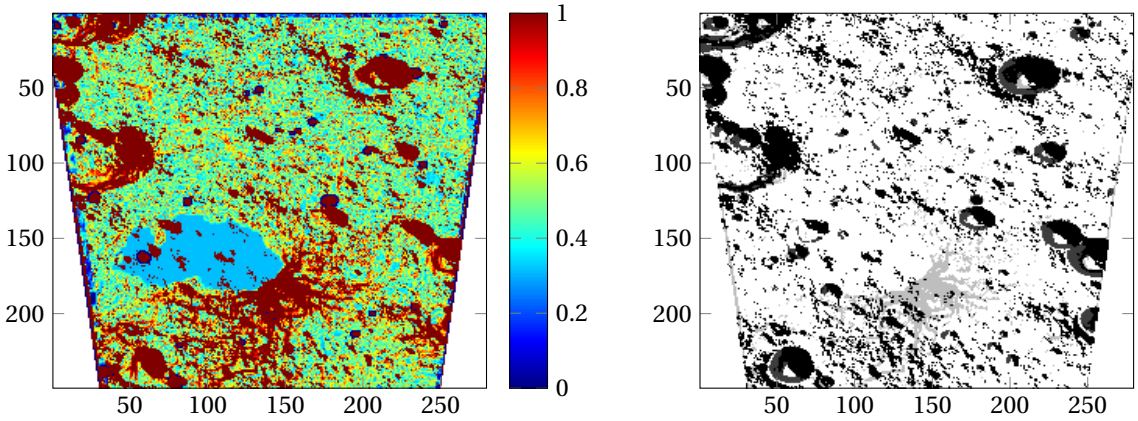

Figure K.3: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 600 m

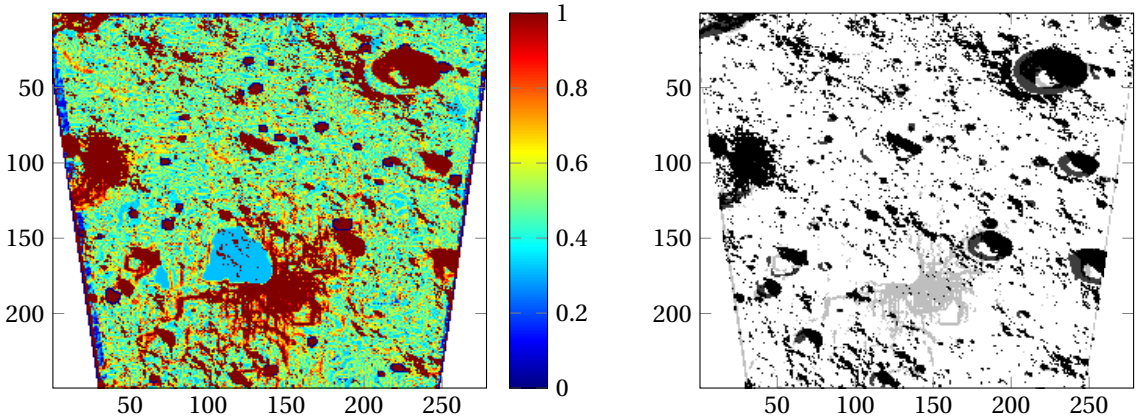

Figure K.4: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 500 m

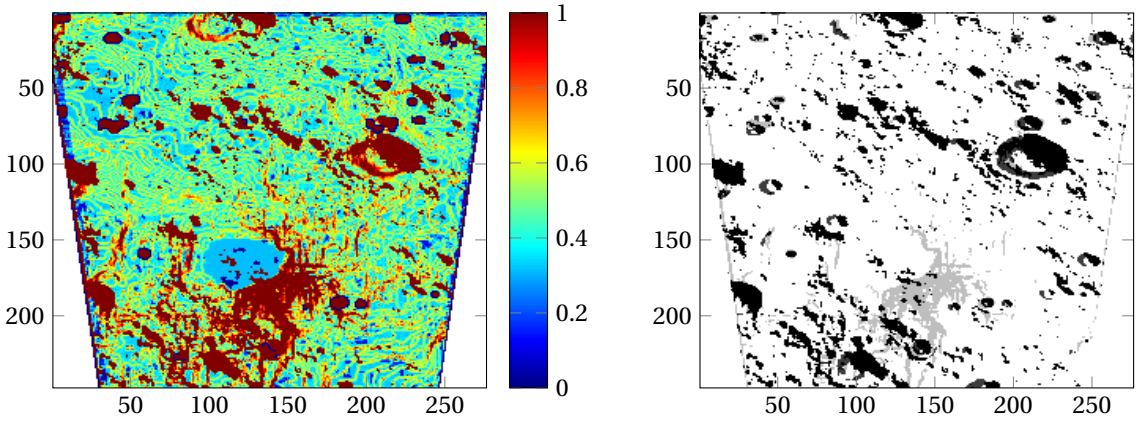

Figure K.5: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 300 m

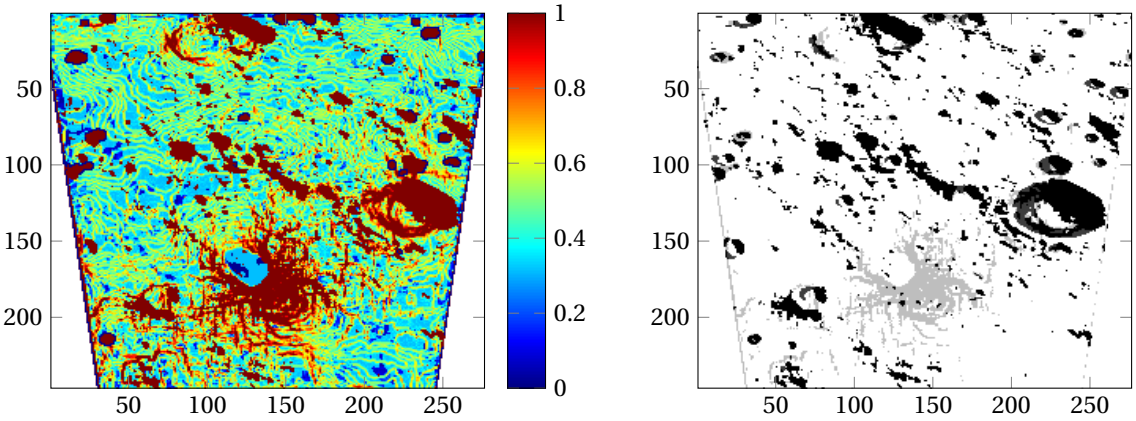

Figure K.6: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 250 m

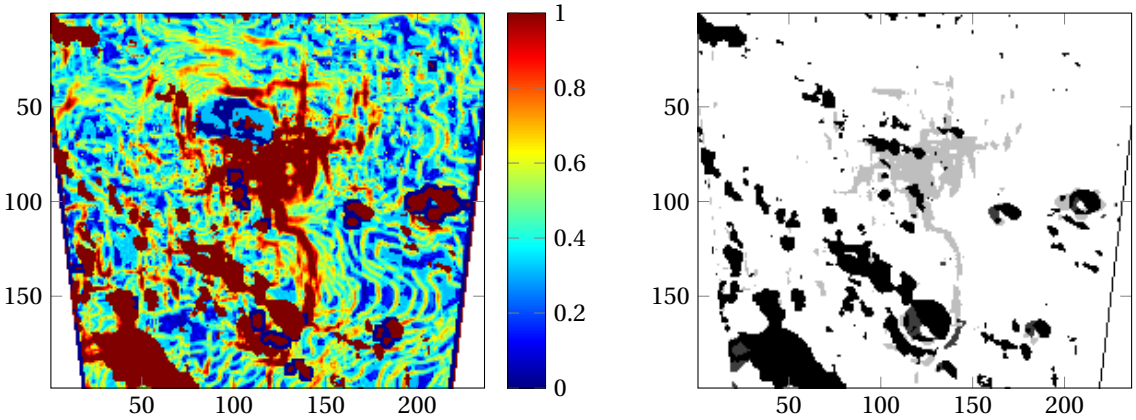

Figure K.7: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 150 m

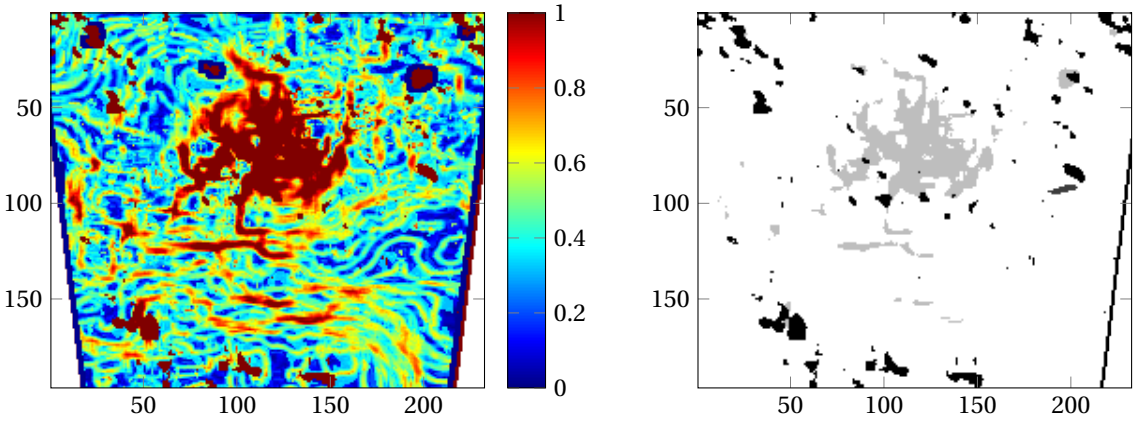

Figure K.8: Scaled hazard map (left) and hazard mapping errors darkgray=FP, white=TP, lightgray=FN, black=TN (right). Altitude h = 125 m

Appendix L

Feasible Scenarios - Slope and Roughness

This appendix hereby shows the computed slope and roughness maps for the feasible scenarios (Section 6.2).

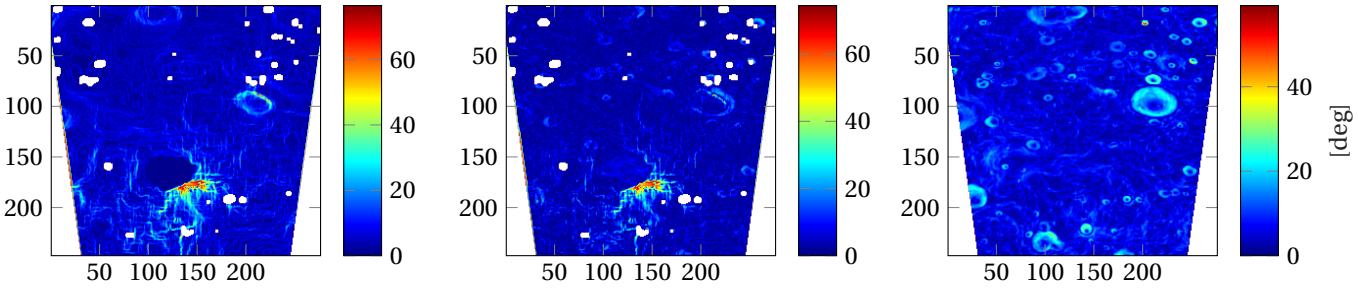

Figure L.1: Computed DEM slope (left) slope error (center) and ground-truth slope (right). Altitude h = 300 m

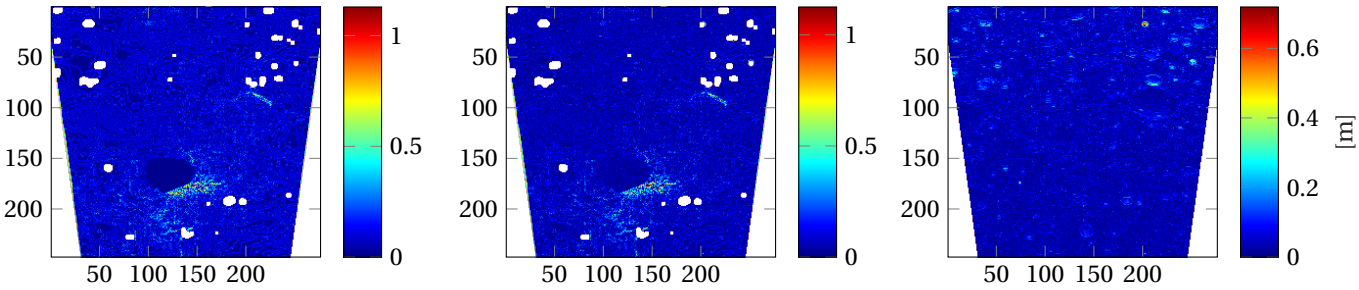

Figure L.2: Computed DEM roughness (left) roughness error (center) and ground-truth roughness (right). Altitude h = 300 m

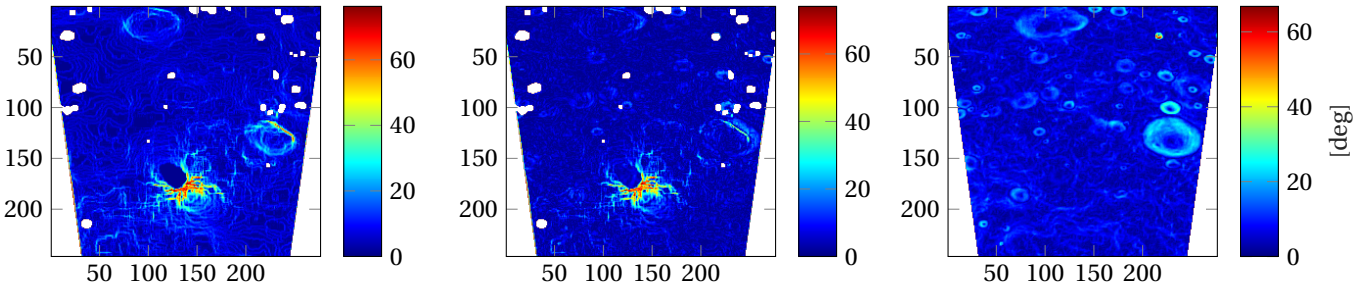

Figure L.3: Computed DEM slope (left) slope error (center) and ground-truth slope (right). Altitude h = 250 m

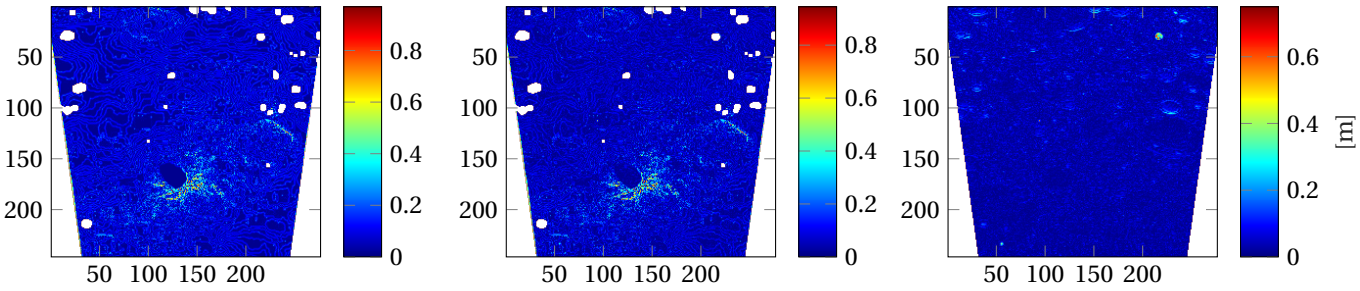

Figure L.4: Computed DEM roughness (left) roughness error (center) and ground-truth roughness (right). Altitude h = 250 m

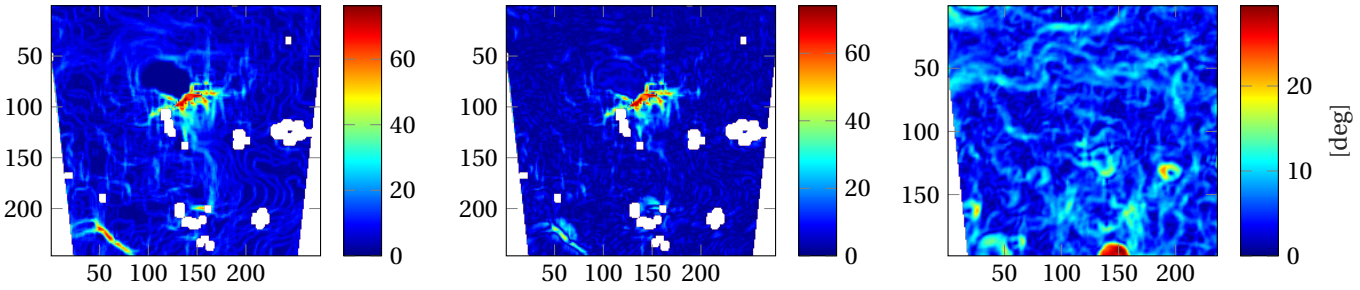

Figure L.5: Computed DEM slope (left) slope error (center) and ground-truth slope (right). Altitude h = 150 m

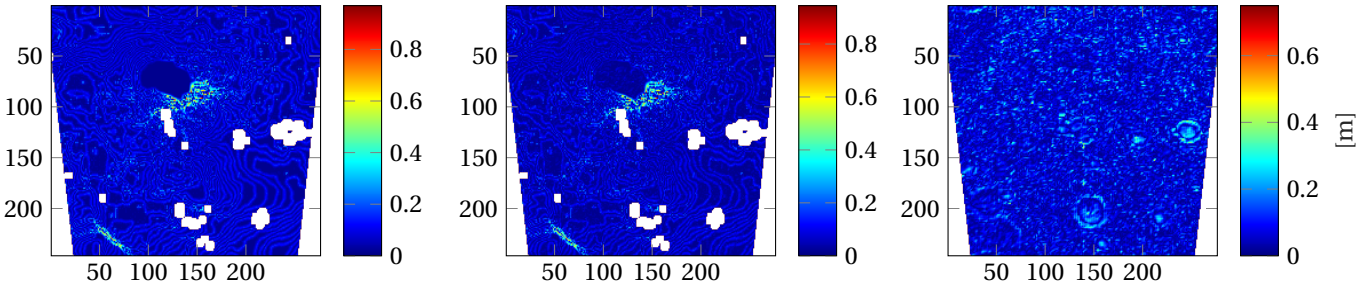

Figure L.6: Computed DEM roughness (left) roughness error (center) and ground-truth roughness (right). Altitude h = 150 m

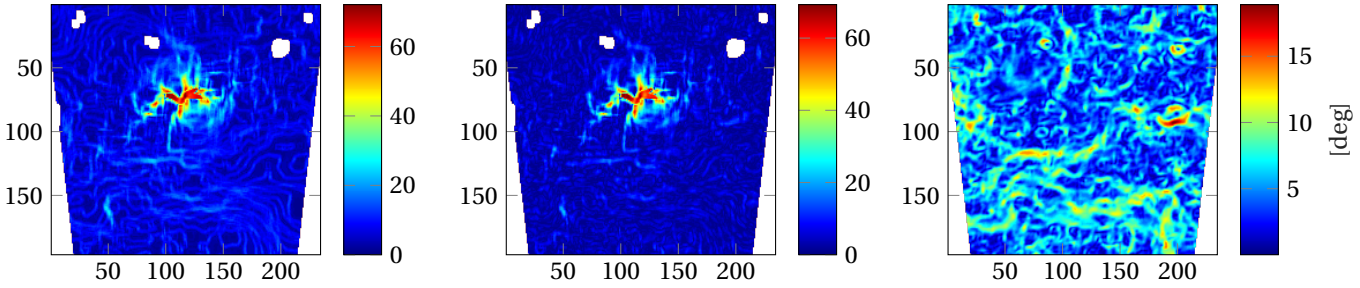

Figure L.7: Computed DEM slope (left) slope error (center) and ground-truth slope (right). Altitude h = 125 m

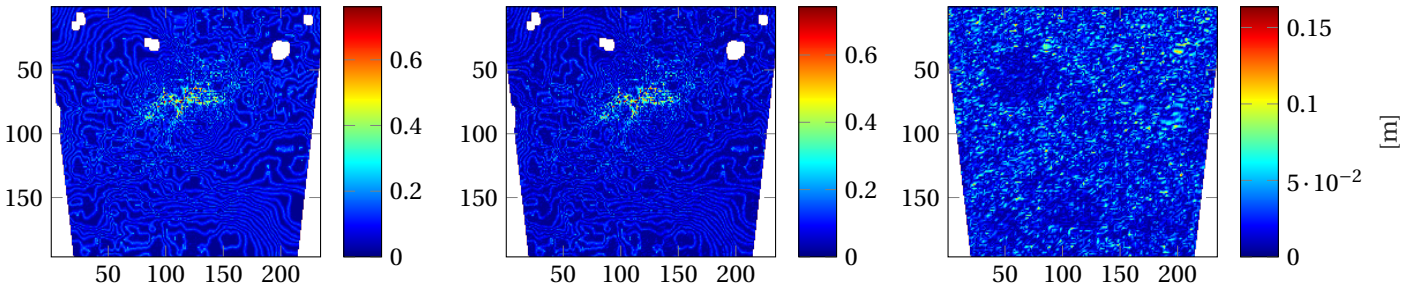

Figure L.8: Computed DEM roughness (left) roughness error (center) and ground-truth roughness (right). Altitude h = 125 m

Bibliography

[1] Lunar Landing Systems Study - ESA Contract 9558/91/NL/JG(W0 N24). Technical report, Matra Marconi Space, 1997.

[2] NASA Space Technology Roadmaps and Priorities: Restoring NASA's Technological Edge and Paving the Way for a New Era in Space. Technical report, 2012.

[3] Foessel-Bunting A. and Whittaker W. MMW-Scanning Radar for Descent Guidance and Landing Safeguard. In *6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2001.

[4] Huertas A. and Madison R. Passive Imaging Based Multicue Hazard Detection for Spacecraft Safe Landing. *2006 IEEE Aerospace Conference*, 2006.

[5] Huertas A., Cheng Y., and Matthies L.H. Automatic hazard detection for landers. *Intelligence, R & A in Space*, 2008.

[6] Vanzella A. Literature Study: Structure from Motion Hazard Detection. Technical report, TU Delft, 2016.

[7] Parreira B., Vasconcelos J.F., Montaño J., Ramón J., and Penin L.F. Hazard Detection and Avoidance in ESA Lunar Lander: Concept and Performance. *AIAA Guidance, Navigation and Control Conference*, pages 1–12, 2013.

[8] Parreira B., Vasconcelos J.F., Oliveira R., Caramagno A., Motrena P., Dinis J., and Rebordão J. Performance Assessment of Vision Based Hazard Avoidance During Lunar and Martian Landing. Technical Report 1, DEIMOS and Faculty of Sciences of the University of Lisbon, 2010.

[9] Chu C. Development of Advanced Entry, Descent, and Landing Technologies for Future Mars Missions. In *Aerospace Conference*, Big Sky, MT, USA, 2006. IEEE.

[10] Harris C. and Stephens M. A Combined Corner and Edge Detector. *Procedings of the Alvey Vision Conference*, 1988.

[11] Meng C., Zhou Na., Xue X., and Jia Y. Homography-based depth recovery with descent images. *Machine Vision and Applications*, 24(5):1093–1106, 2013.

[12] Epp C.H. and Smith T.B. Autonomous Precision Landing and Hazard Detection and Avoidance Technology (ALHAT). *2007 IEEE Aerospace Conference*, pages 1–7, 2007.

[13] Neveu D., Mercier G., Hamel J.F., Simard Bilodeau V., Woicke S., Alger M., and Beaudette D. Passive versus active hazard detection and avoidance systems. *CEAS Space Journal*, 7(2):159–185, 2015.

[14] Mooij E. Re-entry Systems: Lecture Notes AE4870B, 2015.

[15] Hirschmuller H. 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*, 2:807–814, 2005.

[16] Ye J. and Yu J. Ray geometry in non-pinhole cameras: A survey. *The Visual Computer*, 2013.

[17] Carpenter J. D., Fisackerly R., De Rosa D., and Houdou B. Scientific preparations for lunar exploration with the European Lunar Lander. *Planetary and Space Science*, 74(1):208–223, 2012.

[18] Montgomery J.F., Johnson A.E., and Matthies L.H. The Jet Propulsion Laboratory Autonomous Helicopter Testbed: A Platform for Planetary Exploration Technology Research and Development. 23(3):1–23, 2006.

[19] Stocky J.F., Stevens C.M., Nelson R.M., and Seybold C. NASA's New Millennium ST9 TRGS Mission, 2007.

[20] Engel J.J. *Autonomous Camera-Based Navigation of a Quadrotor*. PhD thesis, Technical University of Munich, 2011.

[21] Carson J.M., Bailey E.S., Trawny N., Johnson A.E., Roback V.E., Amzajerdian F., and Werner R.A. Operations concept, hardware implementation and ground-test verification of a hazard detection system for autonomous and safe precision lunar landing. *Advances in the Astronautical Sciences*, 150:2275–2292, 2014.

[22] Carson J.M., Trawny N., Robertson E., Robak V.E., Pierrottet D., Devolites J., Hart J., and Estes N.J. Preparation and Integration of ALHAT Precision Landing Technology for Morpheus Flight Testing. *AIAA SPACE 2014 Conference and Exposition*, 2014.

[23] Losi L. and Lavagna M. *Visual Navigation for Autonomous Planetary Landing*. PhD thesis, Politecnico di Milano, 2016.

[24] Larson Willey J. *Applied Space Systems Engineering*. Number 1. 2009.

[25] Feetham L.M., Aouf N., Bourdarias C., and Voirin T. Single camera absolute motion based digital elevation mapping for a next generation planetary lander. *Acta Astronautica*, 98(1):169–188, 2014.

[26] Bajracharya M. Single image based hazard detection for a planetary lander. *Automation Congress, 2002 Proceedings of the 5th Biannual World*, 14:585–590, 2002.

[27] Strömgren M. *Mitigation of GX209HA Processor for Usage in Space*. PhD thesis, Lulea University of Technology, 2016.

[28] Barker M.K., Mazarico E., Neumann G.A., Zuber M.T., Haruyama J., and Smith D.E. A new lunar digital elevation model from the Lunar Orbiter Laser Altimeter and SELENE Terrain Camera. *Icarus*, 273:346–355, 2016.

[29] Trawny N., Mourikis A.I., Roumeliotis S.I., Johnson A.E., and Montgomery J. Vision-Aided Inertial Navigation for Pin-Point Landing using Observations of Mapped Landmarks. *Journal of Field Robotics*, 24:357–378, 2007.

[30] Lunghi P., Ciarambino M., and Lavagna M. Vision-Based Hazard Detection With Artificial Neural Networks for Autonomous Planetary Landing. In *13th Symposium on Advanced Space Technologies in Robotics and Automation - ASTRA*, 2015.

[31] Rogata P., Di Sotto E., Camara F., Caramagno A., Rebordao J.M., Correia B., Duarte P., and Mancuso S. Design and performance assessment of hazard avoidance techniques for vision-based landing. *Acta Astronautica*, 61, 2007.

[32] Hartley R. and Zisserman A. *Multiple View Geometry in computer vision*. Cambridge University Press, 2nd edition, 2004.

[33] Weih R.C. and Mattson T.L. Modeling Slope in a Geographic Information System. *Journal ofthe Arkansas Academy of Science*, 58:100–108, 2004.

[34] Brook S., Chen B., Dachille F., and Kaufman A. Forward Image Warping. *IEEE Visualization*, 1999.

[35] Li S., Jiang X., and Tao T. Guidance Summary and Assessment of the Chang'e-3 Powered Descent and Landing. *Journal of Spacecraft and Rockets*, 53(2):258–277, 2016.

[36] Recker S., Shashkov M. M., Hess-flores M., Baltrusch R., Butkiewicz M. A., and Joy K. I. Structure-From-Motion Systems for Scene Measurement. *Quality Digest*, 2014.

[37] Woicke S. and Mooij E. A Stereo-Vision Based Hazard-Detection Algorithm for Future Planetary Landers. *LPI Contributions*, 2014.

[38] Woicke S. and Mooij E. A stereo-vision hazard-detection algorithm to increase planetary lander autonomy. *Acta Astronautica*, 122:42–62, 2016.

[39] Woicke S. and Mooij E. Passive Hazard Detection for Planetary Landing. *AIAA Guidance, Navigation, and Control Conference*, 2016.

[40] Striepe S. A., Epp C.D., and Robertson E.A. Autonomous Precision Landing and Hazard Avoidance Technology (ALHAT) Project Status as of May 2010. In *IEEE Transactions on Aerospace and Electronic Systems*, number May, 2010.

[41] Parkes S.M., Martin I., Dunstan M., and Matthews D. Planet Surface Simulation with PANGU. *Space OPS 2004 Conference*, 2004.

[42] Jiang X., Li S., and Tao T. Innovative hazard detection and avoidance guidance for safe lunar landing. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 230(11):2086–2103, 2016.

[43] Xiong Y., Olson Calrk F., and Matthies L.H. Computing depth maps from descent images. *Machine Vision and Applications*, 16(3):139–147, 2005.

[44] Xiong Y. and Matthies L.H. Error Analysis of a Real-Time Stereo System. In *Computer Vision and Pattern Recognition Proceedings*, San Juan, Puerto Rico, USA, 1997. IEEE.