



HAL
open science

Navigation visuelle pour l'atterrissage planétaire de précision indépendante du relief

J. Delaune

► **To cite this version:**

J. Delaune. Navigation visuelle pour l'atterrissage planétaire de précision indépendante du relief. Systèmes embarqués. Institut Supérieur de l'Aéronautique et de l'Espace - ISAE, 2013. Français. NNT: . tel-01021431

HAL Id: tel-01021431

<https://theses.hal.science/tel-01021431>

Submitted on 9 Jul 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Université
de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)

Présentée et soutenue par :

Jeff DELAUNE

le jeudi 4 juillet 2013

Titre :

Navigation visuelle pour l'atterrissage planétaire de précision
indépendante du relief

École doctorale et discipline ou spécialité :

EDSYS : Systèmes embarqués et robotique

Unité de recherche :

Équipe d'accueil ISAE-ONERA CSDV

Directeur(s) de Thèse :

M. Guy LE BESNERAIS (directeur de thèse)

M. Jean-Loup FARGES (co-directeur de thèse)

Jury :

M. Éric MARCHAND - INRIA / IRISA (rapporteur)

M. Philippe MARTINET - École Centrale de Nantes / IRCCYN (rapporteur)

M. Simon LACROIX - CNRS / LAAS (examineur)

M. Antoine MANZANERA - ENSTA / U2IS (examineur)

M. Guy LE BESNERAIS - ONERA / DTIM (directeur)

M. Jean-Loup FARGES - ONERA / DCSD (co-directeur)

INSTITUT SUPÉRIEUR DE L'AÉRONAUTIQUE ET DE L'ESPACE
ONERA - SYSTEMS CONTROL AND FLIGHT DYNAMICS DEPT.

DOCTORAL THESIS

submitted in fulfillment of the requirements for the degree of
Doctor of Philosophy in Embedded Systems and Robotics

by

Jeff DELAUNE

VISION-BASED NAVIGATION FOR PINPOINT PLANETARY LANDING ON ANY RELIEF

Thesis defended on July 4th, 2013, in front of the jury composed of :

Mr	ÉRIC MARCHAND	INRIA / IRISA	(Reviewer)
Mr	PHILIPPE MARTINET	École Centrale de Nantes / IRCCYN	(Reviewer)
Mr	SIMON LACROIX	CNRS / LAAS	(Examinator)
Mr	ANTOINE MANZANERA	ENSTA / U2IS	(Examinator)
Mr	GUY LE BESNERAIS	ONERA / DTIM	(Director)
Mr	JEAN-LOUP FARGES	ONERA / DCSD	(Co-director)

Advisors out of jury :

Mr	THOMAS VOIRIN	ESA
Mr	CLÉMENT BOURDARIAS	Astrium ST
Mr	MARTIAL SANFOURCHE	ONERA / DTIM

To my parents

ACKNOWLEDGEMENTS

WHEN I decided to enter PhD studies a while ago, I remember I wanted to assess my personal limits in terms of pushing scientific research about a specific issue on my own. Had I ever been so wrong ? A PhD is anything but an individual undertaking. It is a collective adventure, made possible and fueled by many people and institutions along the way. Here are those who counted for mine and I am thankful for.

I met my director Guy Le Besnerais knowing exactly which problem I was willing to tackle, landing robots with unreached precisions on other planets, but I did not have many clues of how I would eventually proceed. He taught me computer vision, a field I keep working in now, and that science is also about creativity, a skill he employs in many more places than the lab only. Jean-Loup Farges provided priceless help to all my data fusion and filtering problems. Ten minutes of his time often saved a whole day of work for me. Thomas Voirin took our work to the next level by providing a budget, contacts, and technical supervision for the design of the hardware test bench Visilab. I had my funniest hours of work doing this. Clément Bourdarias ensured what we did was consistent with aerospace vehicle designs for short-term industrial applications. Last but not least, Martial Sanfourche was the best unofficial advisor I could have wished for. I did not know a man could make such an efficient use of machines as Martial does, and learning how to fly them on quadrotors was so much fun with him as a training partner.

Beyond my supervisors, several institutions got together to fund this PhD and provide me with a range of service throughout it : ONERA in Toulouse and Palaiseau, ESA-ESTEC in Noordwijk, and Astrium Space Transportation in Les Mureaux. I am especially grateful to some people. Marcel van Slogteren was always available for me at the workshop in ESA-ESTEC when I needed to machine new parts or advice for Visilab. At ONERA in Toulouse, Monique Péron took a special interest in my situation and helped me through all the administrative procedures as I relocated five times in three years in the four previous places. Alain Piquereau, Vincent Fuertes, Paul Chavent, and Pierre Escalas took me on memorable UAV helicopter tests and acquired flight data at the end of my PhD which shall be the basis for future work. Finally, Patrick Fabiani gave this study very valuable visibility by allowing communications in international conferences and journals, sometimes planned at the last minute.

Anyone who has been through it knows that a PhD is not a regular work experience or a simple extension of studies. You carry it in your mind 24/7 for three years. You live with it. You talk about it with people around you, your friends, your family. In return, you get their support (or you kick them out). Little talks can make big differences. I tried to gather together in a list people which I consider related to the achievement of this PhD, but I know I am forgetting many. Thibaut, I am still not sure of how he supported me as an office mate for a year and a half with Coco. The ONERA Palaiseau team : Aurélien, Pauline, Laure, Paul, Fabrice, Champ', Vincent, Bertrand, Philippe, Hicham, Patrick and Françoise. The ONERA Toulouse team : Nico

Guy, Lara, Simon, Yann, Laure, Aurélie, Razvan, Yoko, Olivier, Guillaume, Philippe and Valérie. The ESTEC team : Charlène, Jonathan, Samir, Steeve, Olivier, Phil, and Monique. The Astrium team : Jean-Philippe, Marine, Laurent, Sorya and Julien. My friends, always there : Yan', Ka-Wai, Jérém', Marc, Ken, Kara, Nico Sarda, Julie, Ludo, Andy and Olivier. Laura deserves credit for living with me and cheering while I was getting crazy writing the thesis. And my longest support, never failing : my parents have always believed in my dreams and keep pushing me on my way to them.

Thank you all !

Paris, July 2013

CONTENTS

LIST OF FIGURES	xii
LIST OF TABLES	xiii
ACRONYMS	xvi
PREFACE	xvii
1 INTRODUCTION	1
1.1 ROBOTIC EXPLORATION OF THE SOLAR SYSTEM	4
1.1.1 Goals and strategy	4
1.1.2 Successful landing missions	4
1.1.3 Landing accuracy	6
1.2 PINPOINT PLANETARY LANDING CHALLENGES	8
1.2.1 Autonomy	9
1.2.2 Accuracy	9
1.3 REFERENCE MISSION AT THE LUNAR SOUTH POLE	10
1.4 TERRAIN SENSOR SELECTION	11
1.4.1 Altimeter	11
1.4.2 Lidar	12
1.4.3 Camera	12
1.4.4 Trade-off	12
1.5 OBJECTIVES AND CONTRIBUTIONS	13
1.6 THESIS OVERVIEW	14
2 ABSOLUTE VISUAL NAVIGATION: AEROSPACE LITERATURE REVIEW	15
2.1 ABSOLUTE IMAGE MATCHING	18
2.1.1 Full-image correlation	19
2.1.2 Photometric landmark description	19
2.1.3 Geometric landmark description	20
2.1.4 Discussion	21
2.2 DATA FUSION ARCHITECTURES	22
2.2.1 Loose integration	24
2.2.2 Tight integration	24
2.2.3 Discussion	27
2.3 PERFORMANCE EVALUATION	28
2.3.1 Software simulation	28
2.3.2 Hardware experiment	29
2.3.3 Discussion	30
3 A TIGHT VISUAL-INERTIAL FILTER FOR LANDING ON ANY RELIEF	31

3.1	LION FILTER OVERVIEW	33
3.1.1	Inertial propagation	34
3.1.2	Tight image measurements	34
3.2	FILTER IMPLEMENTATION	35
3.2.1	State propagation	35
3.2.2	Covariance propagation	36
3.2.3	Filter update	37
3.2.4	State management	38
3.3	POINT-BASED SOFTWARE SIMULATION TESTS	38
3.3.1	Simulation environment	38
3.3.2	Performance	40
4	2D/3D GEOMETRIC VISUAL LANDMARK RECOGNITION	45
4.1	IMAGE-BASED SOFTWARE SIMULATION ENVIRONMENT	47
4.2	LANDMARK RECOGNITION USING IMAGE REPROJECTION	48
4.2.1	Offline map	48
4.2.2	Online image processing	49
4.2.3	Navigation performance	53
4.2.4	Discussion: the repeatability issue	58
4.3	REPEATABILITY IMPROVEMENT THROUGH LANDMARK SCALE MANAGEMENT	59
4.3.1	Scale-space model	59
4.3.2	Scale-invariant extractor selection	60
4.3.3	Scale reprojection formula	61
4.3.4	Comparison of repeatability rates	63
4.4	LANDMARK RECOGNITION USING SCALE-AUGMENTED IMAGE REPROJECTION	64
4.4.1	Offline map	65
4.4.2	Online image processing	65
4.4.3	Navigation performance	66
5	HARDWARE TEST BENCH DESIGN & ANALYSIS	71
5.1	LUNAR-REPRESENTATIVE REQUIREMENTS ANALYSIS	73
5.2	TEST BENCH DESIGN	75
5.2.1	Camera and lens	75
5.2.2	Planetary surface model	77
5.2.3	Mock-up platform	80
5.2.4	Camera platform	82
5.2.5	Illumination system	84
5.3	TEST BENCH OPERATIONS	86
5.3.1	Image acquisition	87
5.3.2	True pose computation	88
5.4	ABSOLUTE NAVIGATION ERROR OBSERVABILITY	89
5.4.1	Manual ground truth accuracy	89
5.4.2	Observable absolute navigation error model	93
5.5	VISUAL GROUND TRUTH REFINEMENT	95
5.5.1	Method	95
5.5.2	Performance	95
6	HARDWARE PERFORMANCE EVALUATION	99
6.1	ACCURACY TESTS	101
6.1.1	NASA LRO orbital data	101
6.1.2	Scalings	102

6.1.3	Nominal descent accuracy evaluation	103
6.2	ROBUSTNESS TESTS	103
6.2.1	Camera sensor change	106
6.2.2	Off-nadir descent camera inclination	107
6.2.3	Illumination change	110
6.2.4	Non-flat terrain	110
7	CONCLUSION	115
7.1	MAIN CONTRIBUTIONS	117
7.2	PERSPECTIVES	118
7.2.1	Validation	118
7.2.2	Extending Lion	120
A	ELEMENTS OF QUATERNIONS FOR ATTITUDE REPRESENTATION	123
A.1	DEFINITION	125
A.2	MULTIPLICATION	126
A.3	RELATIONSHIP WITH ROTATION MATRICES	127
A.4	TIME DERIVATIVE	129
B	NAVIGATION SYSTEMS	131
B.1	NAVIGATION FRAMES	133
B.1.1	Inertial frame	133
B.1.2	Global frame	134
B.1.3	Body frame	134
B.1.4	Local frame	135
B.2	SENSOR INPUT	135
B.2.1	Inertial measurement unit	135
B.2.2	Star tracker	136
B.2.3	Terrain sensors	136
B.2.4	Summary	137
B.3	TERRAIN MAP DATA SOURCE	138
B.3.1	Orbital image	138
B.3.2	Digital elevation model	138
B.4	ERROR STATISTICS	138
B.4.1	Mean	139
B.4.2	Dispersion	139
C	KALMAN OPTIMAL FILTERING AND DERIVATION OF <i>Lion</i> EKF EQUATIONS	141
C.1	KALMAN AND EXTENDED KALMAN FILTERING	143
C.1.1	Continuous-time Kalman filter	143
C.1.2	Discrete-time Kalman filter	144
C.1.3	Extended Kalman filter	145
C.2	DERIVATION OF <i>Lion</i> EKF EQUATIONS	146
C.2.1	Linearized propagation model	146
C.2.2	Linearized measurement model	149
C.2.3	State augmentation	152
D	VISILAB FRAMES	155
D.1	MOCK-UP	157
D.2	CALIBRATION PATTERN	157
D.3	CAMERA CALIBRATION	157

D.4	WORKING CAMERA	158
E	HARDWARE PERFORMANCES TABLES	159
E.1	ACCURACY EVALUATION	161
E.1.1	Nominal descent accuracy	161
E.1.2	Contribution of Visilab environmental error	162
E.2	ROBUSTNESS EVALUATION	163
E.2.1	Camera sensor change	163
E.2.2	Off-nadir descent camera inclination	163
E.2.3	Illumination change	164
E.2.4	Non-flat terrain	166
	REFERENCES	167
	PUBLICATIONS	173

LIST OF FIGURES

1.1	Gale crater near martian equator viewed from orbit	5
1.2	Self-portrait of the Curiosity rover	7
1.3	The GNC loop	8
1.4	Lunar landing reference scenario	11
1.5	Topographic map of the lunar south pole area.	13
2.1	Vision-based navigation framework	17
2.2	Inertial navigation architecture	23
2.3	Loosely-coupled visual-inertial fusion	24
2.4	Tightly-coupled visual-inertial fusion	25
2.5	Simultaneous Localization And Mapping visual-inertial EKF scheme . .	26
2.6	Sliding windows visual-inertial EKF scheme	27
3.1	Navigation system architecture	33
3.2	Point-based simulation environment	39
3.3	Evolution of the number of absolute image measurements in point-based simulation	41
3.4	Attitude estimation Monte Carlo errors in point-based simulation . . .	43
3.5	Velocity estimation Monte Carlo errors in point-based simulation	43
3.6	Position estimation Monte Carlo errors in point-based simulation	44
4.1	Descent image generated with PANGU	48
4.2	Map generation process	49
4.3	Harris image features extracted online and predicted by the filter	50
4.4	Shape context signature	51
4.5	GHT accumulation grid	52
4.6	Putative matching method comparison	54
4.7	GHT attitude estimation Monte Carlo errors in image-based simulations	55
4.8	GHT velocity estimation Monte Carlo errors in image-based simulations	56
4.9	GHT position estimation Monte Carlo errors in image-based simulations	57
4.10	GHT number of absolute image measurements in Monte Carlo image-based simulations	57
4.11	Scale change illustration	60
4.12	Scale reprojection geometry	62
4.13	Repeatability rate comparison	64
4.14	Landmark selection process	66
4.15	<i>Lion</i> attitude estimation Monte Carlo errors in image-based simulation .	67
4.16	<i>Lion</i> velocity estimation Monte Carlo errors in image-based simulation .	68
4.17	<i>Lion</i> position estimation Monte Carlo errors in image-based simulation	69
4.18	<i>Lion</i> number of absolute image measurements in Monte Carlo image-based simulations	69

5.1	Initial equipment available in the lab	74
5.2	Optical lens sizing	75
5.3	Visilab camera and lens mounted on a tripod head	76
5.4	NASA DEM of the lunar south pole selected	78
5.5	Candidate landing sites near the lunar south pole	79
5.6	DEM artefacts are visible at the bottom left with the vertical sizing approach	80
5.7	Visilab DEM versus initial NASA DEM	81
5.8	Planetary surface mock-up being delivered in Visilab	82
5.9	Support platform and mock-up assembly in Visilab	83
5.10	Camera robotic platform in Visilab	84
5.11	Example images of the same area in Visilab with different illumination conditions	85
5.12	The Visilab test bench during data acquisition at ESA-ESTEC	86
5.13	Visilab frames	89
5.14	Mock-up scanning process	92
5.15	Frames used in the absolute navigation error model in Visilab	94
6.1	Position estimation errors at high altitude with the low resolution map	104
6.2	Comparison of position estimation errors at low altitude with both the low and high resolution maps (LR and HR, respectively).	105
6.3	Contribution of Visilab environmental error to final position dispersion	106
6.4	Real image from the Visilab camera sensor and virtual image of the same area by PANGU.	107
6.5	Comparison of estimated position error with a Visilab or a PANGU map	108
6.6	Comparison of estimated position error for different camera tilt angles .	109
6.7	Comparison of estimated position error for azimuth light change with respect to the map.	111
6.8	Comparison of estimated position error for elevation light change with respect to the map.	112
6.9	Visilab DEM modified with a spherical distortion	113
6.10	Comparison of estimated position error at orbital altitudes with and without planar terrain assumption.	114
7.1	Helicopter UAV test flight trajectory over the Caylus site near Toulouse	119
B.1	Detailed GNC architecture	133
B.2	Inertial frame	134
B.3	Global frame	134
B.4	Body frame	135
B.5	Body and local frames	135
B.6	Pixel footprint of the descent camera images as a function of altitude during a lunar descent	137

LIST OF TABLES

1.1	Timeline of successful soft landings	6
1.2	Terrain sensor trade-off	12
2.1	Inertial, relative and absolute navigation trade-off	18
2.2	Absolute image matching literature review summary	21
2.3	Data fusion scheme trade-off	28
2.4	Performance evaluation setups	29
3.1	Dispersion of the navigation error at touchdown with various point-based terrain elevation ranges	41
3.2	Dispersion of the navigation error at the end of the visual phase with various point-based terrain elevation ranges	42
4.1	GHT dispersion of the navigation error for all runs on image-based simulations	55
4.2	GHT dispersion of the navigation error for converging runs only on image-based simulations	58
4.3	Dispersion of <i>Lion</i> navigation error for all runs on image-based simulations	67
5.1	Visilab hardware requirements	73
5.2	Main parameters of the camera mounted with the 3.5-mm C-mount lens	76
5.3	Mock-up manufacturing options	77
5.4	Characteristics of Visilab mock-up and DEM	81
5.5	Course ranges of the camera robotic platform in Visilab	85
5.6	Frame-to-frame pose uncertainties in Visilab	90
5.7	Camera support platform deformation uncertainty in Visilab	91
5.8	Manual ground truth accuracy in Visilab	93
5.9	Visual ground truth accuracy in Visilab	96
5.10	Summary of ground truth accuracies in Visilab (3 RMS)	96
6.1	Position error dispersion requirements at lunar descent key altitudes	101
6.2	NASA LRO orbital data summary	102
6.3	Scaled parameters for each sequence acquired in Visilab	102
B.1	Sensor signals and update rates	138
B.2	Monte Carlo test estimates	139
E.1	<i>Lion</i> navigation performance during lunar coasting phase	161
E.2	<i>Lion</i> navigation performance during lunar braking phase	162
E.3	Contribution of Visilab environmental error to final position dispersion	162
E.4	<i>Lion</i> navigation robustness to camera sensor change	163
E.5	<i>Lion</i> navigation robustness to off-nadir descent camera inclination	163

E.6	Lion navigation robustness to light azimuth change	164
E.7	Lion navigation robustness to light elevation change	165
E.8	Lion navigation robustness to non-flat terrain	166

ACRONYMS

AG	Approach Gate
ASI	Italian Space Agency (Agenzia Spaziale Italiana)
DEM	Digital Elevation Model
DLR	Deutsches zentrum für Luft- und Raumfahrt (German Aerospace Center)
DoG	Difference-of-Gaussian
DOI	Descent Orbit Insertion
EADS	European Aeronautic Defence and Space company
EDL	Entry, Descent and Landing
EKF	Extended Kalman Filter
ESA	European Space Agency
ESTEC	European Space Research and TEchnology Centre
FoV	Field of View
GHT	Generalized Hough Transform
GNC	Guidance, Navigation and Control
GPS	Global Positioning System
GT	Ground Truth
HG	High Gate
IMU	Inertial Measurement Unit
JAXA	Japanese Aerospace eXploration Agency
LG	Low Gate
LIDAR	LIght Detection And Ranging
LION	Landing Inertial and Optical Navigation
LLO	Low Lunar Orbit
LoG	Laplacian-of-Gaussian
LR	Low Resolution

LRO	Lunar Reconnaissance Orbiter
NAC	Narrow-Angle Camera
NASA	National Aeronautics and Space Administration (United States)
ONERA	Office National d'Études et de Recherches Aérospatiales (French aerospace research center)
PANGU	Planet and Asteroid Natural scene Generation Utility
PDI	Powered Descent Initiation
PF	Particular Filter
PnP	Perspective-n-Point
RANSAC	RANdom Sample Consensus
RMS	Root Mean Square
SC	Shape Context
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SNR	Signal-to-Noise Ratio
TD	TouchDown
TERCOM	TERrain CONtour Matching
TG	Terminal Gate
UAV	Unmanned Air Vehicle
UKF	Unscented Kalman Filter
USSR	Union of Soviet Socialist Republics
WAC	Wide-Angle Camera

PREFACE

THIS PhD study was funded through the Network/Partnering Initiative (NPI) of the European Space Agency (ESA) by ONERA, Astrium Space Transportation and ESA-ESTEC.

ONERA is a French laboratory for applied aerospace research. This activity involved the Systems Control and Flight Dynamics Department (DCSD) in Toulouse, France, and the Modeling and Information Processing Department (DTIM) in Palaiseau, France.

Astrium Space Transportation is an industrial company of the European Aeronautic Defence and Space company (EADS). It is located in Les Mureaux, France.

ESA's European Space Research and TEchnology Centre (ESTEC) is the main technology development and test centre in Europe. It is located in Noordwijk, The Netherlands.

1

INTRODUCTION

CONTENTS

1.1	ROBOTIC EXPLORATION OF THE SOLAR SYSTEM	4
1.1.1	Goals and strategy	4
1.1.2	Successful landing missions	4
1.1.3	Landing accuracy	6
1.2	PINPOINT PLANETARY LANDING CHALLENGES	8
1.2.1	Autonomy	9
1.2.2	Accuracy	9
1.3	REFERENCE MISSION AT THE LUNAR SOUTH POLE	10
1.4	TERRAIN SENSOR SELECTION	11
1.4.1	Altimeter	11
1.4.2	Lidar	12
1.4.3	Camera	12
1.4.4	Trade-off	12
1.5	OBJECTIVES AND CONTRIBUTIONS	13
1.6	THESIS OVERVIEW	14

NAVIGATION commonly refers to the determination of the position and orientation of a vehicle with respect to a map, as well as planning the route to its destination. It historically developed with marine navigation as early sailors daring to go off the shore needed to find their way back home on the high seas, where no landmarks were available. The inventions of the compass, of the sandglass, and of the chip log enabled the integration of heading and velocity measurements from which mariners could locate themselves. This *dead reckoning* process was not accurate on the long range though, and new navigators started to look up at the stars to get *position fixes*. Portuguese sailors were the first to adapt an astrolabe on board their ships to measure the latitude using the angular height of the Sun at noon, or that of any known star when crossing the celestial meridian at night. All these techniques were mastered before the year 1500, but not until the 18th century and the precision of the chronometers instead of sandglasses were the seamen able to determine their longitude by comparing their reference time to the celestial one calculated from constellations.

Next navigation highlights were pushed by the emergence of the aerospace industry during the course of the 20th century. Dead reckoning evolved from compasses and chip logs to Inertial Measurement Units (IMUs), which were first developed for early rockets. By sensing angular velocities and non-gravitational accelerations, IMUs eventually helped landing men on the Moon and they are now employed on any plane, boat or submarine. Likewise, the use of stars has been replaced by referenced radio signals to triangulate a position either by finding the directions of those signals or by analyzing their encoded information. Radio navigation first appeared over North Atlantic to help steer the ships, but quickly airships and airplanes started querying position fixes too. The number of terrestrial radio beacons exploded with aerial activities, both military and commercial. If they are still in use for instrument landing, global positioning for both airplanes and ships is now dominated by satellite radio signals, like the Global Positioning System (GPS), which are more performant in terms of accuracy, integrity and coverage.

Today, research on navigation is twofold. On the one hand, it seeks to improve performance over territories navigation has already helped men to conquer: land, air, Earth orbit and the surface of the seas. In particular, anyone can now enjoy the benefits of having a complete set of navigation sensors inside their pocket with the recent development of location-based services featured by new smartphones equipped with GPS antennas, compasses and IMUs. For the providers of such services, better navigation performance also means more business opportunities. On the other hand though, some navigation researchers keep trying to figure out ways to explore new spaces further in the solar system, deep down in underground caves or at the bottom of the oceans. In this category, this thesis introduces the *Lion* novel autonomous navigation system, which aims at landing a robotic probe within 100 m of a surface target on another planet than Earth. Such a performance is called *pinpoint landing* and has only been achieved once in history by commander Alan B. Shepard and lunar module pilot Edgar D. Mitchell during the Apollo 14 mission to the Moon on February 5th, 1971. They landed within 50 meters of the selected target. No unmanned robotic probe has ever reached this accuracy yet.

Section 1.1 reminds the reader about the goals and strategy of robotic planetary exploration, previous spacecraft which successfully landed on extraterrestrial worlds and highlights the benefits of a pinpoint landing capability for future missions. Section 1.2 explores what makes autonomous precision landing such a navigation challenge. The lunar south pole reference mission scenario is detailed in Section 1.3.

Section 1.4 introduces terrain navigation as a solution to pinpoint landing and trades off various terrain sensors. Eventually, Section 1.5 states the objectives of the thesis and Section 1.6 provides an overview of its content.

1.1 ROBOTIC EXPLORATION OF THE SOLAR SYSTEM

1.1.1 Goals and strategy

Modern planetary exploration by robotic vehicles primarily aims at improving our scientific knowledge of extraterrestrial bodies in the solar system to make discoveries hopefully also useful here on Earth. Robots may also be sent as scouts for future human exploration.

The exploration strategy is that first missions to a target object study its surface, as well as its atmosphere if there is one, safely from orbit. The new maps allow scientists to detect candidate areas for subsequent surface exploration with instruments placed on a static lander or on a mobile rover. Some concepts of balloon and airships were also proposed recently to get *in situ* atmospheric measurements but none has flown yet.

Figure 1.1 shows a view from orbit of the Gale crater on Mars. It includes many signs that water, a key ingredient of life as we know it, flowed over there in the past. Scientists selected this area as the landing site for the *Curiosity* rover of the National Aeronautics and Space Administration (NASA) which landed in August, 2012. This is a good example of the rationale and the complementarity in the orbit-to-surface exploration strategy. Mars has been considered as a priority for planetary exploration since the 1970s and the end of the Apollo program because of its similarities with Earth. It is a perfect showcase for robotic planetary exploration as it has been visited by tens of missions and is currently home to three active orbiters¹ and two active rovers².

1.1.2 Successful landing missions

Luna 1 is the first known attempt to leave the gravitational field of the Earth. It was launched on January 2nd, 1959 by the Union of Soviet Socialist Republics (USSR) and flew by the Moon two days later. In September the same year, Luna 2 impacted our natural satellite and becomes the first man-made object to ever reach the surface of another celestial body outside of Earth. These missions marked the start of the expansion of the influence of Man in the solar system. In the context of the Cold War, the Space Race made both the Soviet Union and the United States try and launch up to 12 missions per year beyond Earth orbit between 1959 and the end of the manned Apollo program in 1972. This incredible pace then dropped with both budget and the political interest for manned planetary exploration once the course to prestige had been won by the United States. Spacefaring nations have kept sending unmanned probes, but now only addressing scientific goals as there were no future manned mission to pave the way for. These robotic explorers are our eyes and hands out in the solar system, in worlds we cannot yet afford to visit technologically and financially.

Entry, Descent and Landing (EDL) is by far the most critical phase of all planetary exploration missions. It takes a spacecraft travelling at tens of thousands of kilometer

¹Orbiters: Mars Odyssey (NASA), Mars Express (ESA), Mars Reconnaissance Orbiter (NASA).

²Rovers: Mars Exploration Rover *Opportunity* (MER-B, NASA), Mars Science Laboratory *Curiosity* (MSL, NASA).



Figure 1.1 – Gale crater, near martian equator, is viewed from orbit by the THEMIS camera on board the Mars Odyssey spacecraft. Layers and structures at the base of the central mountain, Mount Sharp, are thought to have been shaped by water-carried sediments on early Mars (Credit: NASA, JPL, Arizona State University).

per hour through interplanetary space and lands it softly on the surface of a planet. It challenges aerodynamics, thermal, mechanical, material, communication, propulsion and control engineers all together. This is the price to pay to access the science opportunity and the inspirational pictures of the landscape at the surface.

Table 1.1 draws up a list of planetary exploration programs which included successful soft landing missions, i.e. those in which the spacecraft suffered no damage. There have been 37 of them up to date, the last one was for the Mars Science Laboratory *Curiosity* rover shown in Figure 1.2.

Table 1.1 – *Timeline of successful soft landings. Date intervals are for touchdown between the first and the last successful mission of each program. Eros and Itokawa are asteroids. Titan is the largest moon of Saturn. NASA, ESA, ASI and JAXA are respectively the US, European, Italian and Japanese space agencies.*

Program	Target	Agency / Country	Successful landings	Dates
<i>Luna</i>	Moon	USSR	6	Feb, 1966 - Aug, 1976
<i>Surveyor</i>	Moon	NASA	5	Jun, 1966 - Jan, 1968
<i>Apollo</i>	Moon	NASA	6 (manned)	Jul, 1969 - Dec, 1972
<i>Viking</i>	Mars	NASA	2	Jul - Sep, 1976
<i>Venera</i>	Venus	USSR	8	Dec, 1970 - Mar, 1982
<i>Vega</i>	Venus	USSR	2	Jun, 1985
<i>Mars Pathfinder</i>	Mars	NASA	1	Jul, 1997
<i>NEAR Shoemaker</i>	Eros	NASA	1	Feb, 2001
<i>MER</i>	Mars	NASA	2	Jan, 2004
<i>Huygens</i>	Titan	ESA, ASI, NASA	1	Jan, 2005
<i>Hayabusa</i>	Itokawa	JAXA	1	Nov, 2005
<i>Phoenix</i>	Mars	NASA	1	May, 2008
<i>MSL</i>	Mars	NASA	1	Aug, 2012

1.1.3 Landing accuracy

Among all research into improving mission return of surface planetary exploration, the accurate landing capability may be the most critical technology. To understand this assertion, one needs to understand the motivations for those missions. They can split into three categories, not necessarily exclusive:

- *Science measurements:* The landing site was selected for its scientific interest. However, the area where relevant *in situ* science measurements can be made is limited. Landing out of it would mean mission failure for a static lander, or would waste valuable time and energy for a rover.
- *Engineering constraints:* Surface hazards can make a mission fail or significantly limit its lifetime. For instance, landing on a boulder or on a terrain too much sloped can damage the lander or make it tip over, shadows can limit or even prevent the recharge of the battery from solar panels, etc. Unfortunately, surface regions where those hazards can be avoided may be of limited extent. In



Figure 1.2 – Self-portrait of the Curiosity rover within the Gale crater on Mars. Images were shot on October 31st, 2012 and show 5-km-high Mount Sharp in the background (Credit: NASA, JPL, Malin Space Science Systems).

Section 1.3, we describe the reference mission scenario for this thesis which targets 100-m wide areas at the lunar south pole that are the only places to offer continuous periods of sunlight.

- *Surface assets*: Some missions plan to land in the vicinity of a surface asset previously landed. In a Mars sample return scenario for example, an ascent vehicle is landed close to a rover which has already collected the samples. This rover then brings the samples inside the ascent vehicle so that they can be brought back to Earth. In those cases, the more accuracy, the less roving time to make the two vehicles meet, the less risk.

All things considered, improving the precision at landing allows to decrease the minimum size requirement for science surface targets, makes more regions accessible from an engineering point of view, and globally limits the risk at mission level.

1.2 PINPOINT PLANETARY LANDING CHALLENGES

A pinpoint landing system is a “collection of elements that together are capable of landing a spacecraft within 100 m of the target” (Wolf et al., 2004). Some mission designs already plan for such a capability.

The landing performance depends on that of the Guidance, Navigation and Control systems (GNC), also called the piloting loop. *Guidance* is responsible for determining the reference path from current position to target. *Control* computes the actuations needed to achieve this reference trajectory. Both of these systems need at least the position, velocity, and attitude of the body frame tied to the vehicle with respect to some reference frame to be operational. *Navigation* refers the estimation of those three quantities forming the state vector. The GNC process is illustrated in Figure 1.3. In addition, Appendix B details the standard GNC architecture, the state vector model, and provides useful background about sensors and cartographic data relevant to planetary landing for reader unfamiliar with the domain.

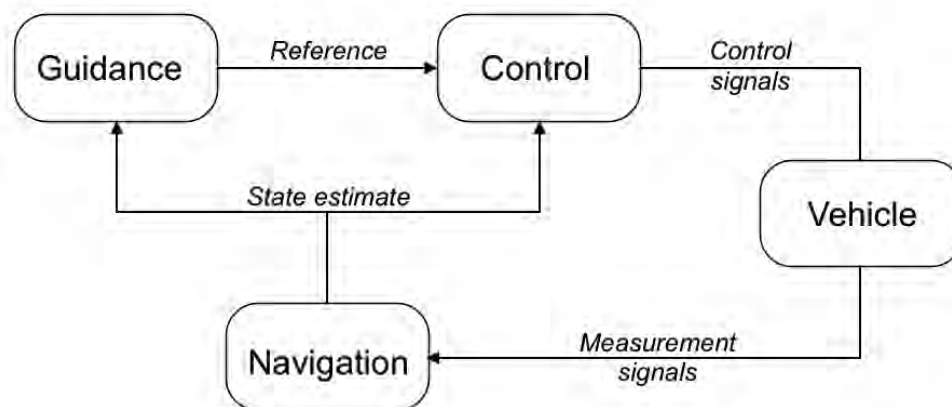


Figure 1.3 – The GNC loop. Navigation estimates the position, velocity and attitude of the spacecraft so that guidance and control can steer it on the correct trajectory.

Two requirements are set upon the GNC system during the soft landing manoeuvre : *autonomy* and *precision*.

1.2.1 Autonomy

Autonomy is a critical asset for any planetary soft landing. From a GNC point of view, it means that the navigation, guidance and control tasks are all performed automatically on-board. Mission success must not rely on a real-time communication link with a remote resource.

The most obvious advantage to any mission is robustness to link failures. In space exploration missions, link cutoffs happen when the spacecraft goes behind a planet with respect to Earth if there is no third-party spacecraft relaying data.

But autonomy is also needed to avoid delays. Due to the limited propagation velocity of electromagnetic signals, operations at long range from Earth-based resources create a lag in the communication loop which prevents any real-time terrestrial remote intervention. For a Mars landing, even at the shortest theoretical distance of 56 million kilometers from Earth, any event is known with a delay of more than three minutes. Any piloting action is thus applied at least six minutes later than what would be possible using on-board autonomy. This is enough time for a mission to fail.

In a lunar mission, the communication lag with the Earth is only roughly one second. This results in a two-second delay for a two-way exchange. It is one of the few space exploration targets for which communication delay is not such an issue, even though it causes serious piloting disturbances. However, link failures will happen when the vehicle is above the far side of the Moon, which concern most of the possible parking orbits. In order to be robust to any orbit choice, and also to a lesser extent to be robust to accidental link failures and avoid communication delays, a fully autonomous GNC loop is thus required to land on the Moon.

1.2.2 Accuracy

Current navigation technology has not changed much since the first landing attempts in the 1960s: the position and the orientation of the landing vehicle are still estimated by integration of the specific force³ and angular rate data provided by an IMU with a similar order of precision. Localization is initialized in orbit from Earth-based range-rate radar measurements with an accuracy of the order of a kilometer on the Moon and worse for further planets. The spacecraft then commits to descent but, unlike on Earth, there are no GPS satellites nor radio beacons available to constrain position error accumulation due to the integration of noisy and biased inertial data during dead reckoning. Close to the ground, an embedded radar senses the height and the horizontal velocity using the Doppler effect to ensure soft landing, but it cannot correct the horizontal position drift with respect to the targeted landing site (Cheatham and Bennett, 1968).

Landing accuracy can be modeled at first order by an ellipse centered on the map point selected as a landing site and which is sized up according to the 3σ values of a 2D normal probability distribution for touchdown location, with σ the standard deviation over the minor or major axes. On missions flown up to date, the length of the major axis of the landing ellipse is of the order of a few kilometers on the Moon, and tens of kilometers on Mars if guided atmospheric entry is performed (Braun and Manning, 2006). These performances are not sufficient for pinpoint landing.

To reach pinpoint accuracy, intense research is being led on the matching of terrain sensor data with a map to infer the position of the vehicle. This process is called *absolute terrain navigation* and it is the key capability to reduce landing uncertainty.

³Non-gravitational acceleration

Other GNC contributions to the landing error are due to relative terrain navigation and control uncertainties, but they arise to less than 3 meters in total (Johnson et al., 2008). Thus at first order, the 100-m pinpoint landing requirement can be translated into a 100-m absolute terrain navigation error requirement.

1.3 REFERENCE MISSION AT THE LUNAR SOUTH POLE

The ESA lunar lander is a future mission, planned by the European space agency, for which the science objective is to investigate the effects of the south pole environment of the Moon for future robotic and human explorers, as well as to look for resources (Fisackerly et al., 2012; Neal, 2009). This is the first planetary robotic exploration mission to call for a pinpoint landing. Indeed, this robotic probe can only survive in daylight using its solar panels as a power source and thus the candidate landing sites are the few locations at the lunar south pole which are continuously illuminated for several months. However, these areas are just about 100 m in radius, hence the requirement for pinpoint landing (Vanoutryve et al., 2010). This mission was selected as the reference test scenario for vision-based navigation in this thesis.

Up to date, two types of approaches have been used to achieve soft landing on the Moon. Direct descent was used by NASA's *Surveyor* spacecraft between 1966 and 1968. Descent from parking orbit was inaugurated by the *Eagle* lunar module on *Apollo 11* and was also used later by the Soviet unmanned landers. The first approach is shorter, so the spacecraft is less exposed to space hazardous conditions before landing. However, parking in lunar orbit allows system checks to be run and potential failures to be recovered before committing to descending towards the lunar surface. As surviving the cislunar environment is better understood nowadays, the opportunity for system checks offered by the descent from parking orbit is preferred and this scenario is selected in this study. The lander is assumed to be delivered in a circular orbit of radius 100 km, called Low Lunar Orbit (LLO). This altitude is usually chosen in order to be close enough to the surface without suffering from disturbances due to anomalies in the gravity field.

Figure 1.4 shows the standard trajectory of the lunar descent scenario. The spacecraft is on LLO at 100 km of altitude when it commits to descent at the Descent Orbit Insertion burn (DOI). There, it starts coasting along half an ellipse around the Moon down to a 15-km altitude where Powered Descent is Initiated (PDI). The braking phase lasts about 10 minutes and aims at cutting the orbital velocity before the Approach Gate (AG). Most of the approach phase happens between High Gate (HG) and Low Gate (LG), when the target landing site is visible and can be analyzed for hazards. The guidance system will proceed to a retargeting if necessary, and command a final vertical descent at Terminal Gate (TG) above a safe area for TouchDown (TD).

Absolute terrain navigation may be employed from LLO to TG: the longer it works, the lesser the position error at touchdown. A correct DOI is important as any mistake made there will insert the spacecraft on a wrong descent orbit which will require dispersion correction and thus unnecessary fuel loss at PDI. Absolute visual measurements are definitely crucial during the braking phase as the main engines firing full thrust can quickly move the spacecraft far away if the position estimate is not accurate. It is also expected that engine vibrations add too much noise to use inertial measurements only. Finally, map-based terrain measurements may still be used during approach, but it might not be necessary as simple frame-to-frame terrain feature tracking can prevent the error from drifting if the landing site remains visible.

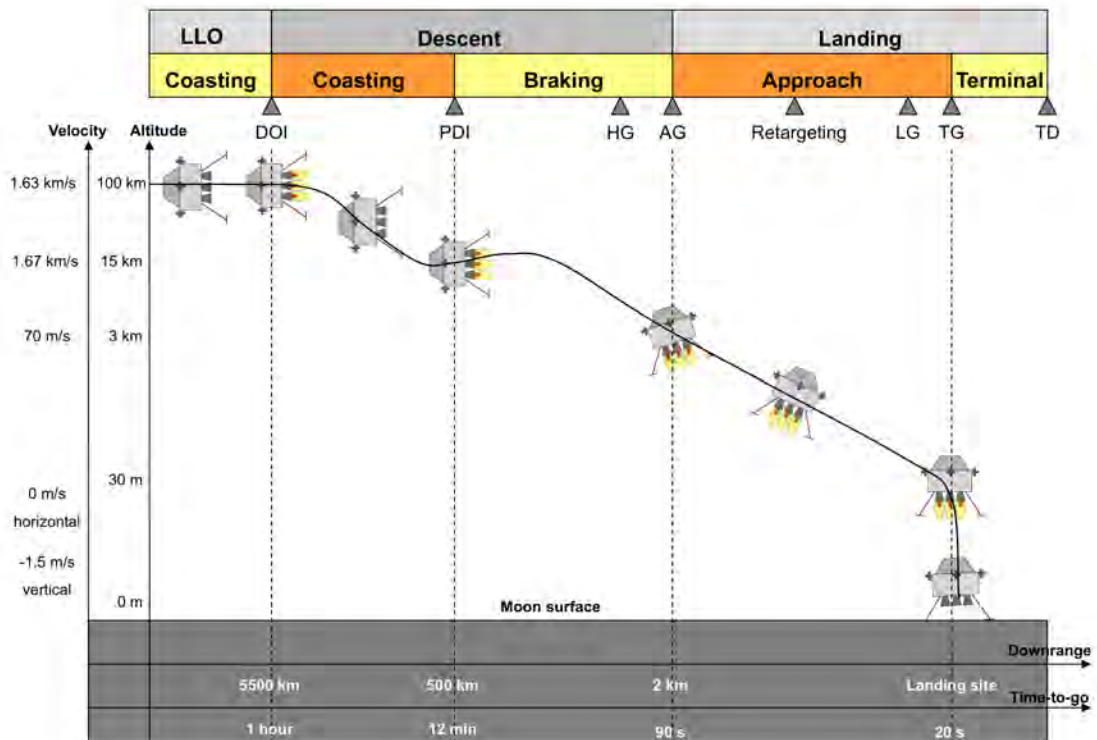


Figure 1.4 – Lunar landing reference scenario

1.4 TERRAIN SENSOR SELECTION

Pinpoint landing requires autonomous and accurate absolute terrain-relative navigation with a dedicated sensor. In this process, data from the terrain sensor is matched with an on-board map to determine the pose, i.e. the position and orientation, of the vehicle with respect to the global frame of the planet $\{g\}$ defined in Subsection B.1.2. The following section reviews the different possible terrain sensors and trade them off. Section B.2 and Section B.3 respectively describe how these sensors work and the orbital data used to create the maps.

1.4.1 Altimeter

The TERrain CONtour Matching technique (TERCOM) has been the first absolute terrain-relative navigation systems since the late 1950s to provide position fixes along the planned route of a missile using a radar altimeter to measure the height, and a barometric altimeter to measure the reference altitude (Siouris, 2004). The difference of those two figures provides the terrain profile which is compared to a reference profile stored in memory. The match is obtained by minimizing the sum of squared differences of the profiles. Based on an active device, TERCOM is independent of illumination but it can only provide a position fix along a 1D line, the planned route.

Modern filtering technique can solve for the full 3D position but they have to fight ambiguous cases depending on the terrain topography (Murangira et al., 2012). In any case, those methods are limited to the maximum operational altitude of an altimeter at about 25 km and they cannot work over a flat terrain.

1.4.2 Lidar

Range images provided by lidars sensing the surface can be matched with a reference orbital Digital Elevation Model (DEM) to solve for absolute navigation.

Hamel et al. (2006) describes various lidar processing techniques. They all detect features on the range image. Featureness is defined as combination of altitude and slope, as both of these quantities can be read simply from lidar data. Features from the online range image and the reference DEM are matched either by correlation techniques or using geometric descriptors.

Lidar-based methods share the illumination independence with altimeter ones, and they allow direct 3D pose estimation by fitting the online and reference point clouds to each other. But like altimeters too, they cannot work over flat terrains and only offer a 5-km range (Hamel et al., 2006).

1.4.3 Camera

Absolute visual navigation uses a camera on line to sense the terrain and the reference map is made from an orbital image of the landing area referenced with a DEM. Since landing missions are planned years in advance, it is anticipated, though not certain yet, that orbital imagery with similar illumination conditions to that of the landing will be available to create the map.

Conte and Doherty (2009) match the whole descent image to orbital one using cross-correlation, but other methods involve landmark detection and matching using a radiometric descriptions (Trawny et al., 2007) or a geometric one (Pham et al., 2012). Landmarks may be detected as generic image features (Lowe, 2004), or specific terrain structures like craters (Cheng and Ansar, 2005).

Cameras are very lightweight, consume low power and take little place on a spacecraft compared to a lidar. As a passive device, they can work from any distance to the observed scene. Nevertheless, they can only work if the terrain is illuminated and motion recovery involves more processing than on active sensors.

1.4.4 Trade-off

Advantages and drawbacks for altimeter, lidar and camera terrain-relative navigation systems are summarized in Table 1.2.

Table 1.2 – *Terrain sensor trade-off*

Sensor	Advantages	Drawbacks
Altimeter	Illumination-independent	Incompatible with flat terrains Limited range: 25 km
Lidar	Illumination-independent Easier pose estimation	Incompatible with flat terrains Limited range: 5 km
Camera	Unlimited range Lightweight	Needs illumination Complex motion recovery

Unlike active sensors such as altimeters or lidars, cameras can work at any distance from an illuminated terrain. That makes them the only sensor physically able to provide an absolute terrain-relative navigation output from the 100-km orbit to

touchdown in the lunar reference scenario of Section 1.3. Cameras are light, cheap, and they can acquire data whether the terrain is flat or hilly. The illuminated-terrain requirement is not a constraint as the mission has to land in daylight for solar panels anyway. Furthermore, the use of a camera for planetary landing was already validated by NASA for the Mars exploration rovers in 2004 when it replaced the traditional doppler radar for horizontal velocity estimation (Johnson et al., 2007).

1.5 OBJECTIVES AND CONTRIBUTIONS

In this thesis, we propose a full navigation system called *Lion* which is able to match details of an image of the surface taken by an on-board camera during a planetary descent with landmarks from a map created before the mission, so as to infer the position and orientation of the vehicle in real time. This process is called *absolute visual navigation* and the map can be made from DEMs orbital images with similar illumination conditions as during descent.

Camera images are 2-dimensional data. Even though they can acquire data over any type of terrain, using them to understand 3D terrain information is not straightforward as one dimension is lost in the projection process. However, there are two key moments of a lunar descent when absolute visual navigation is required but the scene is 3D: at DOI and at low altitude. The former case is certain as the terrain appears spheric and not flat in the field of view of a landing camera on orbit. The latter case is likely, especially with the relief of the lunar south pole illustrated in Figure 1.5 (Mazarico et al., 2012), if the landing area is not flat. Ultimately, the philosophy of pinpoint landing is to be able to target any landing site in a mapped area, whatever its surrounding topography.

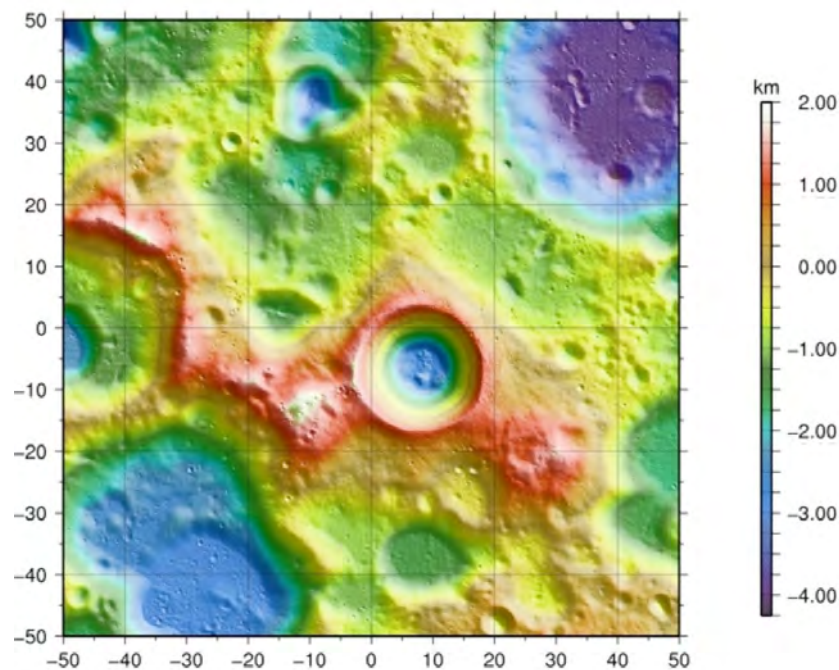


Figure 1.5 – Topographic map of the lunar south pole area. Stereographic projection with axes in kilometers, the original resolution is 10 m per pixel (Mazarico et al., 2012).

This thesis thus aims at designing an absolute vision-based navigation system for

planetary descent and landing with pinpoint accuracy on any terrain relief. The main contributions are:

- A generic image-to-map matcher which makes no topography assumption and can work over any kind of terrain, 2D or 3D,
- An individual management of the image extraction scale for each mapped landmark, guided by the navigation filter, in order to significantly improve the quantity and the quality of correct descent image measurements,
- A new lunar-representative optical test bench for navigation, with simple and cheap ground truth determination,
- Experimental demonstration of the pinpoint landing performance of the image-to-map matcher integrated in a tight visual-inertial filtering scheme,
- A simple system engineering tool to size the navigation error budgets.

1.6 THESIS OVERVIEW

A detailed literature review of the filtering architectures, landmark recognition techniques and performance test benches for aerospace applications is provided in Chapter 2. Chapter 3 details the implementation of a visual-inertial tight filtering scheme compatible with any terrain relief and demonstrates its performances with ideal image-to-map matching over various topographies for the lunar landing scenario. Chapter 4 develops the novel 3D-compatible matching algorithm that searches for each landmark in the image within an area and at a scale predicted by the navigation filter. Chapter 5 details the design of an indoor lunar-analogue optical test facility. It also develops navigation error models to evaluate performance observability due to the limited ground truth accuracy. Chapter 6 assesses the performance of the whole navigation system over key sequences of a scaled lunar descent flight, and its robustness to sensor change, tilt, illumination, and spherical topography. Eventually, Chapter 7 offers a final discussion and concludes over the contributions of this work.

2

ABSOLUTE VISUAL NAVIGATION: AEROSPACE LITERATURE REVIEW

CONTENTS

2.1	ABSOLUTE IMAGE MATCHING	18
2.1.1	Full-image correlation	19
2.1.2	Photometric landmark description	19
2.1.3	Geometric landmark description	20
2.1.4	Discussion	21
2.2	DATA FUSION ARCHITECTURES	22
2.2.1	Loose integration	24
2.2.2	Tight integration	24
2.2.3	Discussion	27
2.3	PERFORMANCE EVALUATION	28
2.3.1	Software simulation	28
2.3.2	Hardware experiment	29
2.3.3	Discussion	30

ABSOLUTE visual navigation matches an image taken by an on-board camera with a terrain map in order to localize the spacecraft in real time. For planetary landing, that means measuring the state vector $x_g = [q_g^b{}^T \ v_{gb}^g{}^T \ p_{gb}^g{}^T]^T$ made of the attitude quaternion q_g^b representing the rotation from the global frame $\{g\}$ to the spacecraft body frame $\{b\}$, the velocity v_{gb}^g and the position p_{gb}^g of the body frame with respect to the global frame, and resolved in the global frame axes. $\{g\}$ is tied to the map of the planet in which a landing site was chosen, it is also the frame in which the landing error is measured. The use of quaternions for attitude representation is presented in Appendix A. All navigation frames are defined in Section B.1.

Absolute measurements may be aided by inertial and relative visual navigation. *Inertial* navigation integrates accelerometer and gyroscope data to measure the change of pose of the spacecraft with respect to an inertial frame $\{i\}$. *Relative* visual navigation can retrieve the motion of camera between the images of a sequence with respect to a local frame $\{l\}$ fixed with respect to the terrain. For instance, $\{l\}$ may be the camera frame at the first image of the sequence, or a frame tied to an observed surface feature like a rock or a mountain. The fusion architecture of absolute, relative and inertial measurements is the core of vision-based navigation illustrated in Figure 2.1. Relevant surveys can already be found in the literature. Johnson and Montgomery (2008) review terrain-relative navigation approaches for pinpoint lunar landing while Bonin-Font et al. (2008) provide an exhaustive survey of vision-based navigation systems for mobile robots in general.

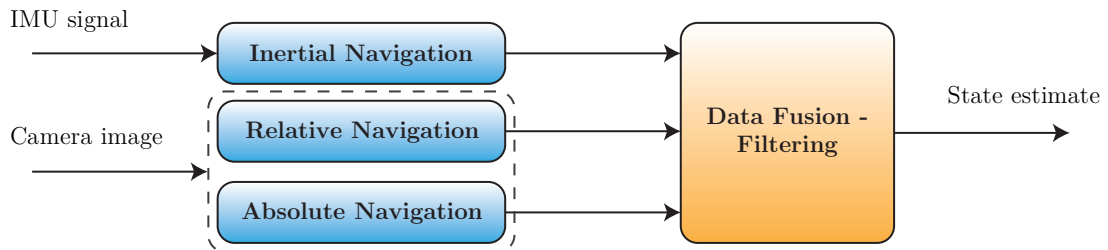


Figure 2.1 – Vision-based navigation framework. Absolute visual navigation may be aided by relative and inertial measurements through a data fusion scheme.

Table 2.1 summarizes the advantages and drawbacks of absolute, relative and inertial navigations. By providing position fixes directly in the global frame $\{g\}$, absolute navigation is the only solution to bring the error in the estimation of x_g down within the 100-m pinpoint accuracy. However, it operates at a relatively slow rate from 1 to 5 Hz which introduces delay in the control loop and makes rapid motion unobservable. Relative terrain navigation needs shorter processing times and thus can work at higher rates between 10 and 60 Hz¹, but it can only estimate the motion between two images. As such, it is a dead reckoning process and it can only limit the divergence of the initial navigation error in $\{g\}$ but it cannot correct it. Eventually, inertial sensors provide self-contained measurements which are independent of external conditions. That is a strong advantage as that means they can back up visual navigation over a shadowed or textureless area. Their high-bandwidth and nearly-instantaneous estimation capability also enable rapid motion observability and solve the control delay issues. However, the integration of inertial bias and noise leads to a rapid divergence

¹The difference between the maximum rate of 60 Hz for relative navigation and the 100-Hz value for cameras stated in Subsection B.2.3 comes from data processing.

of the error observed in missions up to date, even with the aerospace-class IMU performances discussed in Subsection B.2.1.

Table 2.1 – *Inertial, relative and absolute navigation trade-off*

Navigation type	Advantages	Drawbacks
Absolute	Bounded error	Low frequency: 1-5 Hz
Relative	High frequency: 10-60 Hz	Slow error divergence
Inertial	Self-contained measurements Very high frequency: 100-1000 Hz Nearly instantaneous	Rapid error divergence

The slow update rate of absolute navigation means that another navigation type will be needed to perform efficient control of the spacecraft trajectory. Both inertial and relative navigations can solve this slow rate issue. But inertial navigation is the fastest, and it can also provide a back-up to visual navigation failures and IMUs will be onboard anyway for the interplanetary navigation phase. Thus the work presented here is focused on the development of a landing navigation system fusing inertial and absolute visual measurements. *Lion* can actually be read as an acronym for Landing Inertial and Optical Navigation. While similar research is conducted in many different areas ranging from mobile phone applications to planetary landing, this chapter only reviews the relevant literature for aerospace applications.

Section 2.1 reviews the critical step of matching the online image with the terrain map in absolute visual navigation. Section 2.2 introduces the inertial navigation framework commonly used in aerospace systems and surveys data fusion techniques to integrate absolute and inertial measurements altogether. Lastly, Section 2.3 identifies the different test setups that were used in the references reviewed throughout this chapter.

2.1 ABSOLUTE IMAGE MATCHING

Absolute visual navigation can be divided in two steps:

1. Matching the content of the online image with the content of the map,
2. Estimating the pose of the camera, and thus that of the spacecraft carrying it, from the geometry of the matches.

The second step about pose estimation is reviewed in Section 2.2 along with data fusion architectures. This section focuses on the more challenging first step that consists in recognizing the image on the map. To have a chance to do so, the navigation systems needs a description of the image content in some way. This signature can be either the image itself through global correlation, photometric information contained in small patches in the vicinity of specific image features, or the geometric arrangement of those features.

2.1.1 Full-image correlation

The most straightforward way to identify the descent image on the map is to compare pixel values with an orbital image. Conte and Doherty (2009) compute the normalized cross correlation scores of an image taken on board by a nadir-pointing camera over a research zone within a reference image of the area. The flight image was previously scaled and oriented using altitude and heading knowledge from a navigation filter. The height is measured with an altimeter. The location of the maximum score on the reference image gives the 2D absolute horizontal position on the map. Similar correlation techniques are employed during terminal guidance for cruise missiles (Siouris, 2004). Mourikis et al. (2009) also employs image correlation but using a fast Fourier transform to get a coarse estimate of the spacecraft position and initialize a navigation filter. The image is not necessarily shot nadir and is thus rectified using a homography planar transform with altitude and attitude knowledge respectively from an altimeter and a star tracker to correct perspective effect. These approaches have a low processing cost which makes them well suited for real-time applications and can work even if the terrain is poorly textured. However, they are very sensitive to differences between the descent and the orbital images due to changes of viewpoint over non-planar topography. The two following methods intend to better take 3D information into account within the full-image correlation framework.

Adams et al. (2008) proposed an approach called *APLNav* inspired from missile navigation but using two cameras with orthogonal boresights. The scenes observed by each of the two cameras are rendered online using the camera pose estimates propagated by a navigation filter, a DEM, surface reflection, camera and illumination models within an embedded virtual image generator. One camera can only measure accurately the 2D position error in a plane perpendicular to its boresight using image correlation like Conte and Doherty (2009) showed, but the two cameras aligned with a 90-deg offset provide an accurate 3D position fix to update the filter. This approach is interesting because it does take DEM information into account to virtually predict the scenes in front of the cameras, even if the terrain is not flat. Though, each correlation-based measurement happens in 2D within the images and strong difference are still going to affect them, especially over 3D scenes, due to perspective effects if the prediction error is important. In addition to that, the processing power of current space computers is far from being able to run virtual image generators in real time.

Janschek et al. (2006) make a more original use of correlation as they divide the descent image into a dense grid of smaller patches which are subsequently tracked in the online video sequence. This optical flow allows to reconstruct a 3D model of the observed surface, which is then scaled using pose predictions from the navigation filter and matched with the reference DEM map stored in the on-board memory in similar way to lidar navigation. This approach makes a strong performance as, like lidars, it is fully compatible with 3D scenes while being robust to large errors. But like lidars either, it cannot provide any measurement when the terrain is flat.

2.1.2 Photometric landmark description

Modern geometric computer vision tends to process images in terms of feature points relevant for scene estimation (Hartley and Zisserman, 2003). These points are usually detected as extrema responses of the image intensity values convolved with some operator. The surface point associated to a mapped image feature is called *landmark*. Photometric description uses intensity values in the neighborhood of a landmark in the orbital image to retrieve it in the descent image.

Mourikis et al. (2009) process the descent image and a reference orbital one with the Harris operator to extract small image patches around intensity corners. These corners correspond to locally strong intensity changes in all spatial image directions (Harris and Stephens, 1988). On descent data, each template is warped onto the mean plane of the terrain using the altitude and attitude knowledge from the navigation filter. Pseudonormalized cross correlation then allows to register the feature patches onto the orbital image. The orbital image is coregistered with a DEM to obtain the 3D coordinates of these 2D descent image features in the global frame $\{g\}$. The 2D/3D pairs are directly passed to the filter to update the state. This approach benefits from the low processing cost of patch correlation. Although the processing of feature points instead of the whole image would in theory allow it to work over complex terrain topography, the mean plane patch rectification makes correlation fail over non-flat surfaces.

Lowe (2004) proposed the Scale Invariant Feature Transform (SIFT) to extract and match features in two images of the same scene taken from different viewpoints. The algorithm is designed to be robust to translation, orientation, and scale differences between the pair of images. SIFT features are extracted as local extrema of the difference-of-Gaussian operator over the 3D space containing the image plane and the scale space. The SIFT descriptor is based on the magnitude and orientation of the gradients in a neighborhood of the feature, which are computed at its characteristic extraction scale. This information is compiled into histograms for each feature, which can then be matched by finding the minimum Euclidean distance of the histogram with those of the reference image. The Speeded Up Robust Feature algorithm (SURF) is similar to SIFT but made faster by using integral images (Bay et al., 2008). Trawny et al. (2007) have tested the use of SIFT for planetary landing navigation by matching features between the descent and the orbital images. We found SIFT had good performance even with viewpoint changes of more than 40 degrees tested during a lunar descent with similar illumination between the descent and orbital images. This viewpoint robustness also makes them able to cope with a significant range of terrain topography. In addition, they are robust to significant navigation errors as the matching does not need an initial estimate. However, the cost of such robustness is an extensive and computationally-expensive research for features through the image scale space. This approach is adapted to situations where no camera pose information is available before matching. However in planetary landing, an a priori state estimate is propagated by the filter with some accuracy and could be used to make image processing more efficient.

2.1.3 Geometric landmark description

The search for efficiency in absolute image matching has seen landmarks being described by their geometric distribution rather than by intensity information in their neighborhood.

Cheng and Ansar (2005) detect craters in images by finding edges corresponding to the same rim using intensity information, and then try to fit an ellipse model through those edges. Each mapped crater is described by a point in world coordinates, a radius and an orientation vector. Craters are matched two by two with the descent image using an invariant criterion for a pair of conics. The pose is estimated by analytically reconstructing the 3D point cloud of the observed crater centroids in the camera frame, and then by finding the parameters which transforms it to the 3D cloud map. Singh and Lim (2008) detect craters by fitting ellipses onto neighboring like-sized dark and light image regions. Assuming nadir pointing, image locations of

the centroids of the mapped craters are predicted using a priori state estimates from the navigation filter. A least mean square method is performed to match crater predictions with those extracted in the descent image. These two approaches imply no complex image processing and are computationally very efficient. But craters cannot be detected reliably anywhere, even on the Moon, and that does not make them a generic-enough landmark for planetary landing.

Pham et al. (2012) proposed to identify Harris corners detected in the orbital image with their shape context signature. Shape context describes a point using the relative spatial distribution of its neighbors (Belongie et al., 2000). The resolution of the descent image is first corrected for scale using the altitude estimate from the navigation filter, then Harris corners are detected. The descent image is rectified to the orbital image plane using the attitude estimate from the filter. The shape context signatures of Harris features are computed and compared to those of the orbital landmarks using the χ^2 distance to get match candidates. An affine vote finally removes outliers and delivers the final matches. This approach shares the low processing cost of crater-based matching methods while using more generic landmarks. Nevertheless, it makes a flat-world assumption to rectify the descent image through an homography and match surface landmarks.

2.1.4 Discussion

Table 2.2 summarizes the advantages and drawbacks of the absolute image matching algorithms reviewed in this section.

Table 2.2 – *Absolute image matching literature review summary*

Matching type	Method	Advantages	Drawbacks
Full-image correlation	Raw-image correlation (Conte and Doherty, 2009; Siouris, 2004; Mourikis et al., 2009)	Low processing cost Works with poor textures	Flat-terrain assumption
	Rendered-image correlation (Adams et al., 2008)	Works over any topography	Large error sensitivity High processing cost
	Reconstructed-DEM correlation (Janschek et al., 2006)	Works over 3D terrains Robust to large errors	Not over flat terrains
Photometric landmark description	Patch correlation (Mourikis et al., 2009)	Low processing cost	Flat-terrain assumption
	Scale-invariant local gradient distribution (Lowe, 2004; Bay et al., 2008)	Works over any topography Robust to large errors	High processing cost
Geometric landmark description	Crater-based methods (Cheng and Ansar, 2005; Singh and Lim, 2008)	Works over any topography Low processing cost	Only with craters
	Shape context (Pham et al., 2012)	Low processing cost Generic landmarks	Flat terrain assumption

Out of all the approaches reviewed, five meet the objective of working on any terrain relief, with no topography assumption, as stated in Section 1.5 (Adams et al., 2008; Trawny et al., 2007; Bay et al., 2008; Cheng and Ansar, 2005; Singh and Lim,

2008). Adams et al. (2008) render the expected descent image in a virtual generator using DEM and a priori state information to correlate the output with the actual descent image and measure the error. Though, this method cannot cope with large errors and above all the processing cost of a virtual image generator cannot be afforded by modern embedded computers. Crater-based methods like that of Cheng and Ansar (2005) or Singh and Lim (2008) have the lowest processing cost but craters features are not generic enough to be used reliably as landmarks. The scale-invariant methods left match generic landmark using the local intensity gradient distribution (Lowe, 2004; Bay et al., 2008). They can work on any topography and are robust to large errors, but their processing cost remains high because of an extensive search through the image scale space which does not make profit at all of the a priori information available from the filter. In addition, should there be changes of illumination between the orbital and the descent images, the robustness of photometric descriptors appears limited compared to that of geometric descriptors (Pham et al., 2012).

Eventually, we chose to focus on developing matching methods using these geometric descriptors to identify generic landmarks while removing the flat-terrain assumption made by Pham et al. (2012). In addition to enhanced illumination robustness, these approaches also appear promising in terms of simplicity and cost efficiency. Chapter 4 details the various methods designed. It highlights the landmark repeatability issue that led to use scale-invariant properties like Trawny et al. (2007) and Bay et al. (2008) to solve it. The processing cost is kept low by efficiently using the a priori information propagated by the navigation filter.

2.2 DATA FUSION ARCHITECTURES

Data fusion between absolute visual navigation and inertial navigation aims at benefitting from image-based position fixes with respect to the map of the planet to reach pinpoint accuracy, while profiting from the robustness and high bandwidth of IMUs. Before discussing this fusion, let us describe how inertial navigation is typically employed for navigation.

An inertial navigation system estimates the state of the frame tied to the IMU with respect to a reference frame $\{r\}$. It does so by integrating angular rate and specific force measurements with respect to an inertial frame $\{i\}$. In this study, the IMU frame is considered to be the spacecraft body frame $\{b\}$, and the reference is the global frame $\{r\} = \{g\}$. Inertial navigation has moved from research to common application in aerospace systems for decades now. Figure 2.2 illustrates the standard inertial navigation architecture (Groves, 2008).

Inertial navigation is an iterative process, measuring the change of the state from the previous iteration. An initial estimate $(\hat{\mathbf{x}}_r)_0 = \left[(\hat{\mathbf{q}}_r^b)_0^T \ (\hat{\mathbf{v}}_{rb}^r)_0^T \ (\hat{\mathbf{p}}_{rb}^r)_0^T \right]^T$ is thus needed. At each iteration, the angular rate measurements $(\boldsymbol{\omega}_{ib}^b)_m$ are integrated to get an attitude update $\hat{\mathbf{q}}_r^b$. From this, the specific force measurements $(\mathbf{f}_{ib}^b)_m$ can be projected onto the reference frame axes to obtain $\hat{\mathbf{f}}_{ib}^r$. Adding the acceleration due to the gravitational force $\boldsymbol{\gamma}_{ib}^r$ which applies on the body frame with respect to the inertial one, and proceeding to a reference frame change from $\{i\}$ to $\{r\}$, leads to the acceleration estimate which can be integrated to update velocity $\hat{\mathbf{v}}_{rb}^r$. A second integration then provides the updated position estimate $\hat{\mathbf{p}}_{rb}^r$.

Figure 2.1 introduced the global data fusion architecture which is used to fuse visual and inertial measurements. All fusion schemes reviewed are actually based on recursive Bayesian estimation, often Kalman filters or its extensions. Section C.1

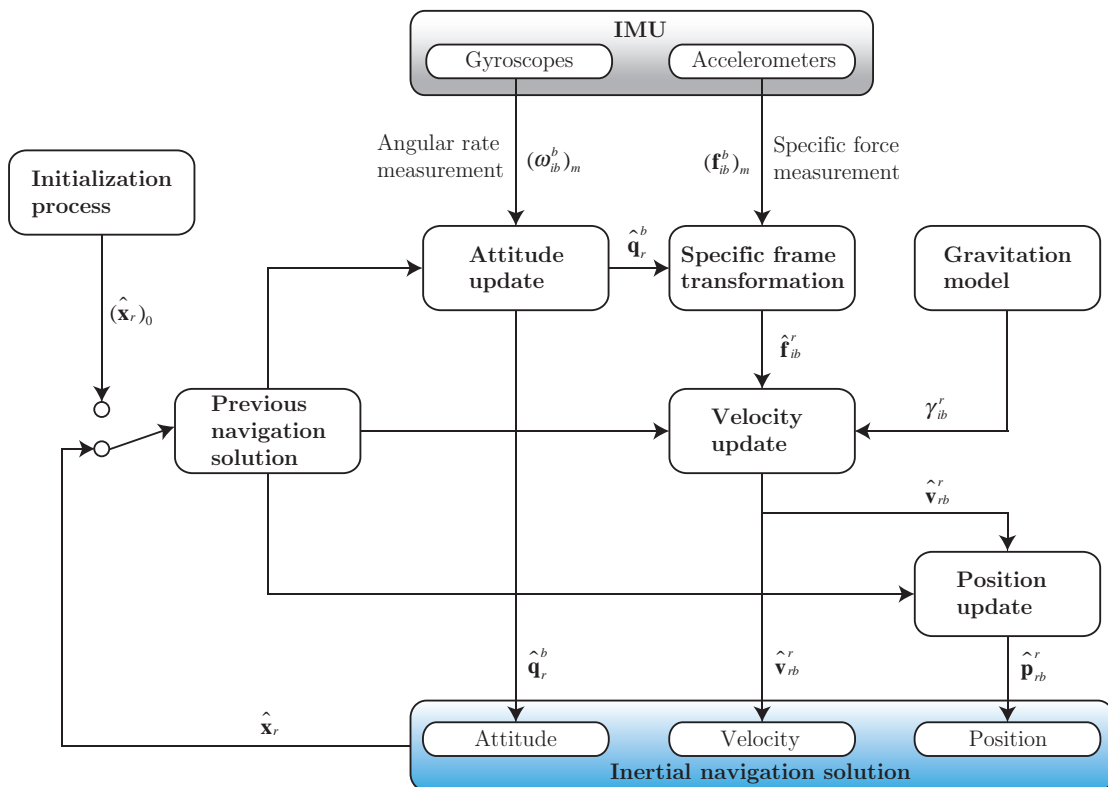


Figure 2.2 – Inertial navigation architecture. The angular rate measurements are integrated to update the attitude of the spacecraft. The specific force measurements can then be projected on the spacecraft axes and added to the gravitational force to update the velocity and the position by successive integration.

and Welch and Bishop (2006) give an introduction to standard and Extended Kalman Filters (EKF), and a detailed description of modern optimal filtering techniques was written by Candy (2009). The nature of the outputs of the inertial, relative², and absolute navigation blocks which are passed to the filtering block distinguish between two types of fusion: *loose* or *tight*.

2.2.1 Loose integration

In a loosely-coupled integration, each navigation subsystem provides an estimate of the vehicle state in the global frame $\hat{\mathbf{x}}_g = \left[\hat{\mathbf{q}}_g^{bT} \quad \hat{\mathbf{v}}_{gb}^{rT} \quad \hat{\mathbf{p}}_{gb}^{rT} \right]^T$ on its own. This process is illustrated in Figure 2.3. The sub-navigation with the highest frequency, often the inertial navigation, is used in the prediction part of the filter while the other state estimates are used as measurements in the correcting part.

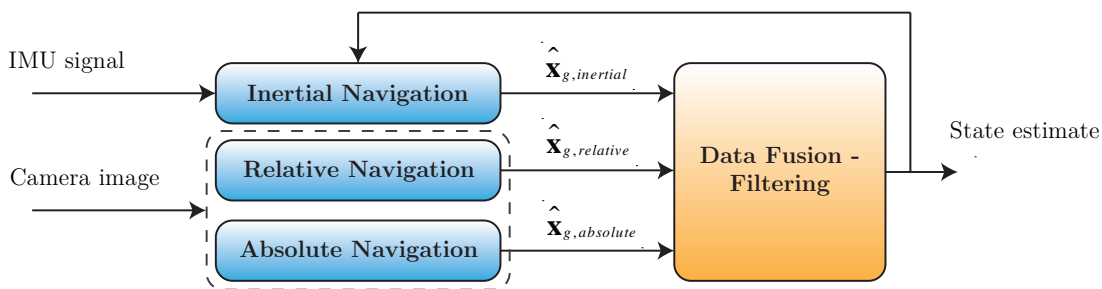


Figure 2.3 – Loosely-coupled visual-inertial fusion. Each navigation subsystem computes a state estimate $\hat{\mathbf{x}}_g$ on its own. Inertial navigation is often used in the propagation part of the filter, with visual poses used as updates.

Pham et al. (2012) estimates the absolute camera position from the image-to-map matches of one camera using least-squares minimization. Both this absolute estimate and the relative position estimated by a visual odometer can update the inertial pose propagated by a Kalman filter.

Adams et al. (2008) employ 3D position estimates computed from each of the two orthogonally-looking cameras to update a navigation filter propagating inertial state estimates.

Conte and Doherty (2009) use a linear error dynamic model in a Kalman filter. It mechanizes the inertial measurements to predict the state and performs an altitude update using a barometric altimeter. The overall fusion architecture is slightly more complex than that of Figure 2.3 as 2D horizontal position update is achieved using the output of another dynamic filter, called *point-mass filter*, which fuses a visual odometer for relative navigation and the image registration module detailed in Subsection 2.1.1 for absolute navigation. The point-mass filter solves the Bayesian filtering problem on a discretized grid and is able to handle general distributions of probability, not only the Gaussian ones.

2.2.2 Tight integration

Unlike loose fusion schemes, tight filters do not receive direct state estimate updates from the navigation subsystems but are rather fed with a rawer type of measurement

²Although this work is focused on inertial and absolute visual navigation, relative navigation appears in this section as some data fusion architectures reviewed make use of it.

from the sensors. For instance in GPS-inertial fusion, the pseudo-range and pseudo-range rates issued by the ranging processor are used as measurements in the filter. With absolute visual systems, the tight measurements are the mapped landmark coordinates (u, v) in the image plane, as shown in Figure 2.4. Then the absolute visual navigation block is only concerned with matching image features to mapped landmarks and pose estimation is centralized within the navigation filter.

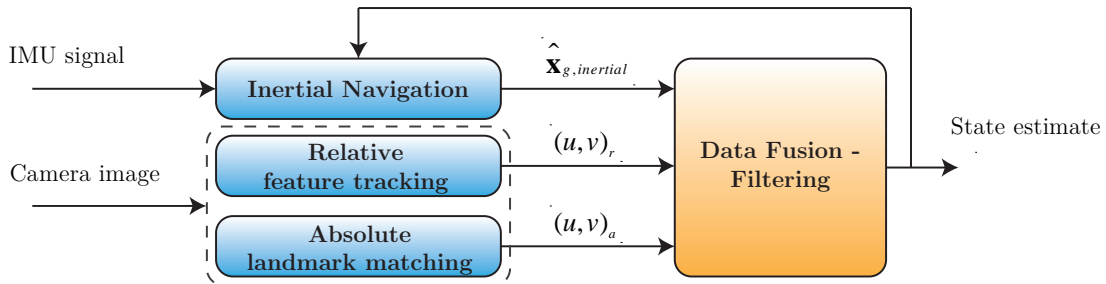


Figure 2.4 – *Tightly-coupled visual-inertial fusion. Relative and visual navigation subsystems only provide image feature coordinates measurements (u, v) to the filter, where pose estimation is centralized.*

We distinguished between two types of tight visual-inertial fusion schemes among those reviewed: Simultaneous Localization And Mapping (SLAM) and sliding-windows filters.

SLAM: state augmentation using feature coordinates

SLAM algorithms estimate the structure of the observed scene, the so-called *mapping* part, along with the pose of the camera looking at it (Davison, 2003). The absolute navigation state is thus estimated by adding the triangulated 3D coordinates $\mathbf{p}_{gl_j}^g$ in the global frame $\{g\}$ of each of the N tracked image features $(l_j)_{j \in [1, N]}$. The general SLAM state vector is then

$$\mathbf{x}_{SLAM} = \left[\mathbf{q}_g^b T \quad \mathbf{v}_{gb}^g T \quad \mathbf{p}_{gb}^g T \quad \mathbf{p}_{gl_1}^g T \quad \dots \quad \mathbf{p}_{gl_N}^g T \right]^T. \quad (2.1)$$

At first, these coordinates have large uncertainties due to the unknown feature depth in monocular vision. But the motion of the camera allows to reduce the variances of those coordinates through the sequence. SLAM is intensively used in relative navigation and building for building local maps on the fly. Absolute navigation rarely uses SLAM as the 3D coordinates of mapped features are a priori known and thus do not need to be triangulated. However, one could think of using the triangulated feature 3D coordinates for absolute matching in a similar way to the DEM matching studied by Janschek et al. (2006).

Flandin et al. (2009) fuses inertial and relative visual measurements using an EKF SLAM depicted in Figure 2.5. The motion is propagated using IMU measurements in the predictive part of the filter. Acceleration measurements by the IMU provide full motion observability so that an altimeter is not required to solve the vision scale problem. To initialize 3D feature positions, their image coordinates are backprojected on the mean plane associated to tracked features.

Likewise, Singh and Lim (2008) estimate the position and velocity of a nadir-pointing vehicle using the inertial coordinates of four tracked crater centroids. The

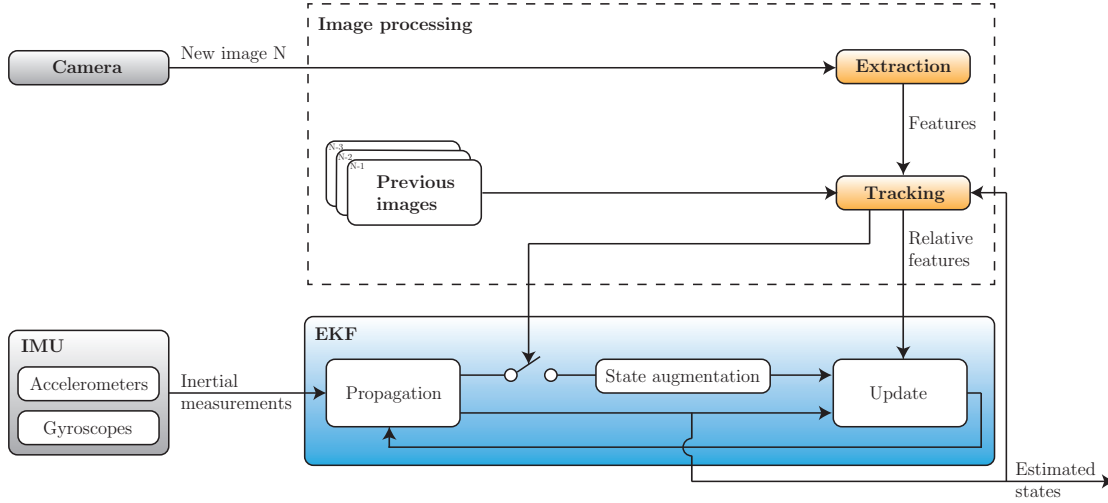


Figure 2.5 – *Simultaneous Localization And Mapping visual-inertial EKF scheme used in Flandin et al. (2009). Image features are tracked through the online video sequence to estimate their coordinates which are added after the camera pose in the state vector.*

states are not propagated with inertial measurements but using a dynamical model of the planet and of the vehicle along with the commanded thrust. Updates are done using unitary line-of-sight vector measurements from the camera to the tracked craters in inertial frame along with an altitude measurement from the altimeter.

Caballero et al. (2008) also implemented an EKF-SLAM. Propagation is done from visual measurements directly as they estimate from the homography transform between two images, assuming the terrain is flat.

Sliding windows: state augmentation using camera poses

Alternatively, sliding-windows architectures estimate the state by appending to it static camera poses $\left\{ \left(\mathbf{q}_g^{b_{k-j}}, \mathbf{p}_{g^{b_{k-j}}}^g \right) \right\}_{j \in \llbracket 1; M \rrbracket}$ corresponding to the M previous images prior to current image k , with $\{b_{k-j}\}$ the spacecraft body frame at the acquisition time of image $k - j$. These extra states are the so-called *sliding windows*. The general state vector of these filters is thus

$$\mathbf{x}_{SW} = \left[\mathbf{q}_g^{b^T} \quad \mathbf{v}_{g^b}^g{}^T \quad \mathbf{p}_{g^b}^g{}^T \quad \mathbf{q}_g^{b_{k-1}^T} \quad \mathbf{p}_{g^{b_{k-1}}}^g{}^T \quad \dots \quad \mathbf{q}_g^{b_{k-M}^T} \quad \mathbf{p}_{g^{b_{k-M}}}^g{}^T \right]^T. \quad (2.2)$$

Image feature 3D coordinates are not estimated any more like in SLAM, but instead the extra static pose states allow the filter to deal with image processing delays and integrate measurements over more than one image. Image measurements can actually only directly observe these static pose states and the delayed update of the dynamic navigation state happens through correlation with the inertial measurements in the error covariance matrix of the filter. Note that the simplest way to do tight filtering would be to add no window at all and make the image measurements directly update the dynamic position and attitude states $\left(\mathbf{q}_g^b, \mathbf{p}_{g^b}^g \right)$ in the filter. However to handle time delays associated with image processing, at least one static camera pose is always appended to the dynamic state.

Since there are usually a lot less camera poses than tracked points considered, based on the Kalman filtering architecture detailed in Section C.1 and for a similar number of image feature measurements, the computational cost of sliding-windows

EKF is less than those of SLAM EKF. Furthermore, this cost grows only linearly with the number of landmarks in sliding-windows, while it grows quadratically with SLAM.

Mourikis and Roumeliotis (2007) and Simard Bilodeau et al. (2010) integrate relative visual measurements tightly with inertial motion estimation of the attitude, velocity and position using sliding windows. The architecture is shown in Figure 2.6. An extended Kalman filter is used for data fusion. Inertial measurements are used for dynamical propagation part of the filter. Updates are based on the difference between the predicted image coordinates of a tracked point, and those measured in the online image. Image features can be either absolute if they are matched with a map or relative if they are unidentified. However, the processing of relative features need to solve for their depth. Sibley et al. (2010) have an approach similar to Mourikis and Roumeliotis (2007) solve this depth problem by initializing the 3D coordinates of the features through stereo imaging, namely achieving a triangulation using two on-board cameras.

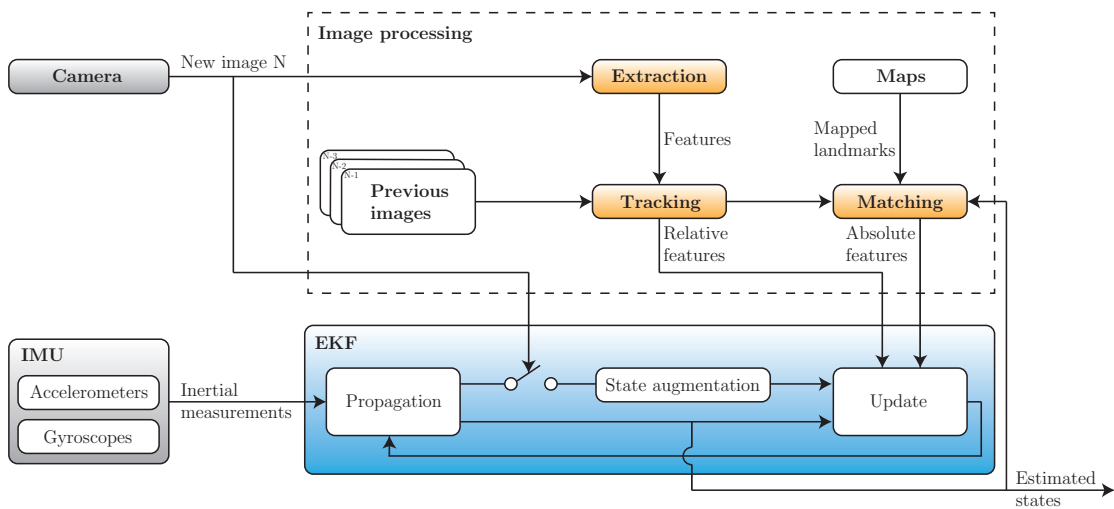


Figure 2.6 – Sliding windows visual-inertial EKF scheme used in Mourikis et al. (2009). Pose parameters of the camera at the times of acquisition of a number k of previous images are added to the state vector to process absolute or relative measurements.

2.2.3 Discussion

Table 2.3 draws a list of the advantages and drawbacks of the loose and tight fusion schemes reviewed in this section.

Loose fusion is simple to implement, requires hardly any more computer processing than the navigation subsystems already do, and having several stand-alone solutions for navigation enable back-ups and double-checks between them. However, loose architectures can meet performance and stability issues when several filters are cascaded and independent landmark-based visual pose estimation, like that selected in Section 2.1, can suffer from degenerate surface feature configurations.

Tight fusion enables a finer noise modeling and thus better precision in the filter as the sensor observation function is the measurement equation of the filter. In degraded observation conditions, tight-coupled visual filters can also update the state even with only one measured image landmark contrary to loose schemes. Though, tight systems have a higher computational cost.

Table 2.3 – *Data fusion scheme trade-off*

Fusion type	Advantages	Drawbacks
Loose	Simple implementation Low computational cost Parallel solutions	Performance and stability issues Degenerate configurations
Tight	Precision Robustness to degraded measurements	Higher computational cost

The finer accuracy of tight filters is especially relevant to pinpoint landing. In addition, the absence of degenerate configurations and the robustness to degraded observation conditions loosens the constraints on image processing. The computational cost, even though higher than for loose schemes, should still fit the planetary landing requirements and we decided to go with a tight filter for *Lion*.

Between SLAM and sliding-windows tight filters, SLAM is not adapted to absolute visual navigation except if 3D reconstruction is needed for absolute image matching. However even in this case, it has a significantly higher cost than sliding-windows filter, and it cannot handle image processing time delays. Furthermore all the absolute image matching methods reviewed in Section 2.1 which made no topography assumption needed no triangulation, thus no SLAM, for recognizing mapped landmarks. We thus decided to set up a sliding-windows tight absolute visual-inertial navigation filter for *Lion*, fully compatible with any terrain topography, and which is described in Chapter 3 along with the evaluation of its performance on a lunar landing trajectory.

2.3 PERFORMANCE EVALUATION

Design of vision-based navigation algorithms for planetary landing is constrained by the difficulty to access flight-representative data with accurate ground truth for validation in various conditions. Most landing sites are exploration targets which, like the Lunar south pole, have never been visited before. For this reason, the absolute image matching solutions reviewed in Section 2.1 employ either *software simulation* or *hardware experiment* in terrestrial planetary analogue environment to test the performance and robustness of their approach. Table 2.4 classifies different performance evaluation setups of the literature.

2.3.1 Software simulation

The cheapest way to get planetary descent images is to render them virtually from a DEM³, camera, surface reflection and illumination models. This is equivalent to what Adams et al. (2008) plan to do online in *APLNav*. Such images can be generated with tools like *PANGU* developed by Parkes et al. (2004). *PANGU* includes a planetary surface generation software based on observed topography, crater and boulder distribution models. It was used to evaluate landmark constellation matching in lunar environment by Pham et al. (2012) and DEM reconstruction and matching for a Mercury landing mission by Janschek et al. (2006). These simulations allow to thoroughly

³Not necessarily a true one.

Table 2.4 – *Performance evaluation setups*

Evaluation	Data type	Reference
Software simulation	True planetary images	Cheng and Ansar (2005) Mourikis et al. (2009) Li (2008) Singh and Lim (2008)
	Rendered planetary images	Adams et al. (2008) Pham et al. (2012) Janschek et al. (2006)
Hardware experiment	Indoor test bench images	de Lafontaine et al. (2008) Sibley et al. (2010)
	Helicopter flight images	Conte and Doherty (2009) Trawny et al. (2007)
	Sounding rocket parachute flight images	Mourikis et al. (2009)

test the robustness of navigation methods in different conditions. However, as the difference between a true image and a virtual one is still noticeable with naked eye, one can question whether all the optical phenomena happening in a real space camera or all the planetary terrain properties are well modeled yet.

Another to simulate descent data is to project true planetary images on a virtual surface, usually a plane or the associated DEM when available, inside a 3D simulation environment which also includes a camera model to render the descent images. Cheng and Ansar (2005) used orbital images from Mars Odyssey probe to test their crater recognition system. Mourikis et al. (2009) analysed the sensitivity of their correlation-based matching module using imagery of Mars, Europe and the Moon. Real images from asteroid 433 Eros were employed by Li (2008) to evaluate relative terrain navigation performance. And eventually, Singh and Lim (2008) tested their absolute navigation system for lunar orbit and landing also using the real image database from the lunar probe *Clementine*. If orbital images are available, this type of simulation is the simplest way to obtain realistic descent images with the associated ground truth to estimate the absolute navigation error. However, it does not allow to vary observation parameters like terrain illumination as usually there is only one or a few images available of the same area. The realism also becomes limited over complex surfaces as the resolution of the true DEMs is often less than that of the images.

2.3.2 Hardware experiment

Sibley et al. (2010) set up a stereo rig on a rail track facing a planetary surface poster to evaluate their relative visual navigation algorithm performances. A similar robotic test facility featuring a camera facing a 3D martian planetary mock-up is planned is discussed by de Lafontaine et al. (2008). These approaches can offer a very representative environment while demonstrating the performances with actual camera hardware. However, indoor test benches do not include flight perturbations like vibrations or atmospheric effects.

Conte and Doherty (2009) and Trawny et al. (2007) both validated their algorithms with data acquired from a helicopter Unmanned Air Vehicle (UAV) equipped with a

camera and a differential GPS antenna offering submeter precision as a reference for error measurement. Likewise, Mourikis et al. (2009) validated their navigation filter with IMU and camera data from a sounding rocket flight over desert in New Mexico. These approaches provide representative flight data and trajectories, however visual data are representative of the Earth and not of another planet.

2.3.3 Discussion

Software simulation is intensively used in early navigation design as it provides a cheap and handy interface to test a vision-based system in various planetary-like conditions. However, hardware experimentation is the next step after this to prove the performance of the system in a physical environment. Only an indoor experiment configuration with a camera facing a 3D terrain mock-up illuminated in some way is able to provide various lunar-representative real images and accurate ground truth with respect to the reference scenario described in Section 1.3. Chapter 5 describes the design of such a lunar-analogue indoor optical test bench.

CHAPTER CONCLUSION

This chapter traded off filtering, image processing, and performance evaluation techniques reviewed in the aerospace literature in order to develop a navigation system fusing inertial and absolute visual measurements to solve the relief-independent pin-point planetary landing problem raised in this thesis.

Since navigation is above all an estimation problem, Chapter 3 develops a tight visual-inertial navigation filter based on a sliding-windows architecture which is compatible with any topography. The performance is evaluated during the approach phase of a lunar landing trajectory assuming ideal image measurements over various terrain topographies. These results set the upper bound of the accuracy which can be reached later with image processing in the loop.

Chapter 4 details the design of various matching methods using geometric descriptors to match generic landmarks while making no assumption about the terrain relief and using efficiently the a priori information propagated by the navigation filter to keep the computational cost down. The performance of the whole navigation system with image processing in the loop is evaluated with computer-generated lunar images.

To push testing one step further, Chapter 5 explains the design of a lunar-analogue indoor optical test bench which was used to simulate key image sequences of the whole lunar descent. Hardware results in accuracy and robustness with this setup are presented in Chapter 6.

3

A TIGHT VISUAL-INERTIAL FILTER FOR LANDING ON ANY RELIEF

CONTENTS

3.1	LION FILTER OVERVIEW	33
3.1.1	Inertial propagation	34
3.1.2	Tight image measurements	34
3.2	FILTER IMPLEMENTATION	35
3.2.1	State propagation	35
3.2.2	Covariance propagation	36
3.2.3	Filter update	37
3.2.4	State management	38
3.3	POINT-BASED SOFTWARE SIMULATION TESTS	38
3.3.1	Simulation environment	38
3.3.2	Performance	40

TIGHT absolute visual-inertial data fusion schemes measure the image locations of landmarks recognized on a map of the planet in order to update the estimate of the navigation state $x_g = [q_g^b{}^T \ v_{gb}^g{}^T \ p_{gb}^g{}^T]^T$ propagated with data coming at high rate from an IMU. Section 2.2 reviewed and traded off tight versus loose fusion architectures. The finer noise modeling of tight schemes makes them more adapted to get the most accuracy out of a landing. In addition, the absence of degenerate configurations and the robustness to degraded observation conditions, when the number of matched landmarks is low, are strong advantages with respect to loose fusion. Among those tight schemes, sliding-windows architectures have a lower computational cost than SLAM ones and integrate image processing delays in the estimation.

Section 3.1 gives an overview of the extended Kalman filter which has been implemented in *Lion* to provide an estimation structure for visual-inertial pinpoint planetary landing navigation, without making any assumption about the terrain relief. The EKF equations are derived from the propagation and measurement models in Section 3.2. And eventually Section 3.3 verifies that this navigation filter reaches a performance compatible with the 100-m accuracy pinpoint landing requirement independently of terrain relief using point-based simulation during the approach phase of a lunar descent over different terrain topographical conditions.

3.1 LION FILTER OVERVIEW

The global architecture of the *Lion* navigation system is illustrated in Figure 3.1.

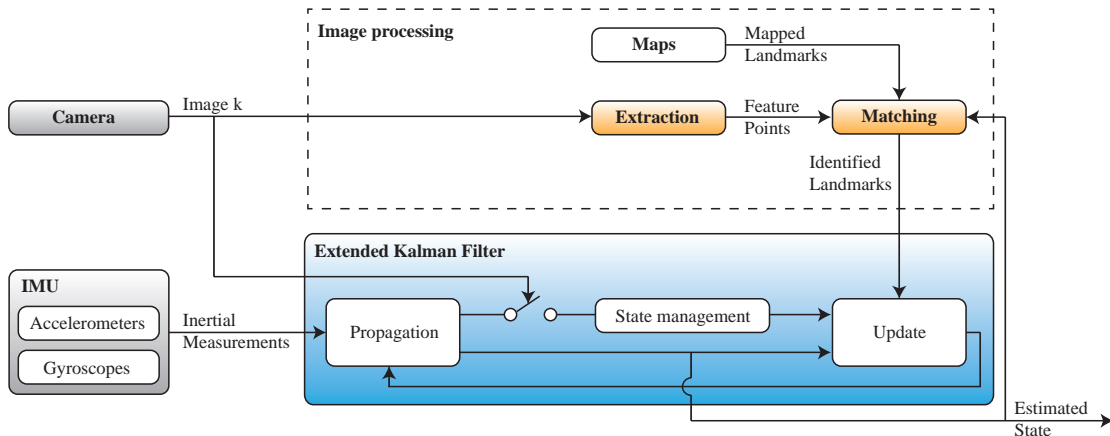


Figure 3.1 – Navigation system architecture. The estimate of the state and its error covariance matrix are propagated using inertial measurements in an EKF. The camera pose associated to the last image is appended to the state to handle time delays while it is being processed for feature points which are ultimately matched with the map in order to provide the EKF measurements.

An extended Kalman filter is employed, which is the simplest extension of the standard Kalman filter to handle non-linearities in the propagation and measurement models. Kalman filters provide a maximum-likelihood estimate of the state of linear systems assuming Gaussian noise. Section C.1 recapitulates the general equations of Kalman and extended Kalman filtering. The EKF in *Lion* uses a continuous-time propagation with discrete-time updates for which the models are provided now.

3.1.1 Inertial propagation

The non-gravitational acceleration and angular rate measurements of the IMU are used in the propagation part of the filter in Figure 3.1. Inertial navigation equations allow to propagate the attitude, velocity and position states of the spacecraft in frame $\{g\}$ by successive integrations, along with their error covariance matrix. The propagated estimates are assumed to be initialized during the orbital phase using a star tracker in attitude and Earth-based range-rate radar measurements for position and velocity.

The system model of the landing vehicle equipped with an IMU was put into equations as

$$\begin{cases} \dot{q}_g^b &= \frac{1}{2}\Omega(\omega_{gb}^b)q_g^b \\ \dot{\mathbf{b}}_{gyr} &= \mathbf{n}_{bgyr} \\ \dot{\mathbf{v}}_{gb}^g &= \mathbf{a}_{gb}^g \\ \dot{\mathbf{b}}_{acc} &= \mathbf{n}_{bacc} \\ \dot{\mathbf{p}}_{gb}^g &= \mathbf{v}_{gb}^g \end{cases} . \quad (3.1)$$

The dynamic navigation state is thus $\mathbf{x}_V = \left[q_g^b{}^T \quad \mathbf{b}_{gyr}{}^T \quad \mathbf{v}_{gb}^g{}^T \quad \mathbf{b}_{acc}{}^T \quad \mathbf{p}_{gb}^g{}^T \right]^T$. q_g^b is the quaternion describing the rotation from the planet frame $\{g\}$ to the vehicle body frame $\{b\}$. \mathbf{p}_{gb}^g , \mathbf{v}_{gb}^g and \mathbf{a}_{gb}^g are respectively the position, velocity and acceleration of $\{b\}$ with respect to $\{g\}$, projected in the axes of $\{g\}$. In addition to the standard navigation state, \mathbf{x}_V also includes the gyroscope bias \mathbf{b}_{gyr} and the accelerometer bias \mathbf{b}_{acc} projected $\{b\}$, which is assumed to be the same as the IMU frame. Those two quantities are modeled by a random walk with centered Gaussian white noise, respectively \mathbf{n}_{bgyr} and \mathbf{n}_{bacc} . ω_{gb}^b is the angular velocity vector with respect to the planet frame and projected on the vehicle frame. The operator Ω defined by

$$\Omega(\omega) = \begin{bmatrix} 0 & -\omega^T \\ \omega & -[\omega\wedge] \end{bmatrix}, \quad (3.2)$$

$$\text{where } [\omega\wedge] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (3.3)$$

The definition of Ω in Equation (3.2) depends on the convention presented in Appendix A and used here for the quaternions: the first component is the scalar part and the three others are the imaginary parts. The angular velocity and acceleration data are inferred from the IMU measurements ω_{IMU} and \mathbf{a}_{IMU} using

$$\omega_{IMU} = \omega_{gb}^b + \mathbf{C}(q_g^b)\omega_{ig}^g + \mathbf{b}_{gyr} + \mathbf{n}_{gyr}, \quad (3.4)$$

$$\mathbf{a}_{IMU} = \mathbf{C}(q_g^b)(\mathbf{a}_{gb}^g - \mathbf{g}_b^g + 2[\omega_{ig}^g\wedge]\mathbf{v}_{gb}^g + [\omega_{ig}^g\wedge]^2\mathbf{p}_{gb}^g) + \mathbf{b}_{acc} + \mathbf{n}_{acc}. \quad (3.5)$$

$\mathbf{C}(\cdot)$ is the coordinate change matrix associated to a quaternion, ω_{ig}^g is the angular velocity vector of the planet frame with respect to the inertial frame, \mathbf{g}_b^g is the local gravity vector, \mathbf{n}_{gyr} and \mathbf{n}_{acc} are centered Gaussian white noises.

3.1.2 Tight image measurements

Each time an image is acquired, the current dynamic pose is added as a static state in the filter as indicated by the state management block in Figure 3.1. Unlike the

closest approach in the literature depicted in Figure 2.6 (Mourikis et al., 2009), only one static pose is added to the dynamic state here: that of the last image taken in order to deal with image-processing time delays. Indeed, there is no need for *Lion* to add more than one pose if the accuracy is high enough after evaluation. The image is processed for feature points which are matched with the map. As shown in Figure 3.1, the absolute matching algorithm provides the 3D coordinates in global frame $\mathbf{p}_{gl_j}^s$ of the point j identified at normalized image coordinates $\mathbf{z}_j = [u_j \ v_j]^T = [u'_j/f \ v'_j/f]^T$, with $\mathbf{z}'_j = [u'_j \ v'_j]^T$ the pixel image coordinates and f the focal length in pixels. The EKF measurement model is then

$$\mathbf{z}_j = h_j(\mathbf{x}) + \mathbf{n}_j = \frac{1}{z_{cl_j}^c} \begin{bmatrix} x_{cl_j}^c \\ y_{cl_j}^c \end{bmatrix} + \mathbf{n}_j, \quad (3.6)$$

where $\mathbf{p}_{cl_j}^c = [x_{cl_j}^c \ y_{cl_j}^c \ z_{cl_j}^c]^T = \mathbf{C}(\mathbf{q}_g^c)(\mathbf{p}_{gl_j}^s - \mathbf{p}_{gc}^s)$ is the 3D position of the point in the camera frame and \mathbf{n}_j is the normalized image measurement noise of covariance matrix $\mathbf{R}_j = \sigma_{im}^2 \mathbf{I}_2$ due to the uncertainties in the feature extraction process. The EKF measurements are thus the image positions of the matched landmarks which can be predicted using their world coordinates recovered from the orbital image coupled to a DEM of the area, the a priori state estimate and the calibrated camera model. These landmark measurements are used to update the corresponding static pose and the dynamic state through correlation. The static state is ultimately removed by the state management function and a new image is taken to repeat this iterative process.

Equation (3.6) is a simple pinhole projection model but it makes no assumption at all about the terrain topography. Coupled with the 3D knowledge of the DEM use to build the map, it is the key to enable *Lion* to work over any type of terrain relief.

3.2 FILTER IMPLEMENTATION

This section derives the actual EKF equations implemented in *Lion* from the inertial propagation and tight image measurement models. Some parts of the complete demonstration are included in Section C.2 and are referred to in the following.

3.2.1 State propagation

The estimate of the state $\hat{\mathbf{x}}_V = E[\mathbf{x}_V]$ is the expected value of \mathbf{x}_V and can be propagated through time by applying the expectation operator E to the terms on each side of System (3.1):

$$\begin{cases} \dot{\hat{\mathbf{q}}}_g^b &= \frac{1}{2} \boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}) \hat{\mathbf{q}}_g^b \\ \dot{\hat{\mathbf{b}}}_{gyr} &= \mathbf{0}_{3 \times 1} \\ \dot{\hat{\mathbf{v}}}_{gb}^s &= \mathbf{C}(\hat{\mathbf{q}}_g^b)^T \hat{\mathbf{a}} - 2 \left[\boldsymbol{\omega}_{ig}^s \wedge \right] \hat{\mathbf{v}}_{gb}^s - \left[\boldsymbol{\omega}_{ig}^s \wedge \right]^2 \hat{\mathbf{p}}_{gb}^s + \mathbf{g}_b^s, \\ \dot{\hat{\mathbf{b}}}_{acc} &= \mathbf{0}_{3 \times 1} \\ \dot{\hat{\mathbf{p}}}_{gb}^s &= \hat{\mathbf{v}}_{gb}^s \end{cases}, \quad (3.7)$$

where $\hat{\mathbf{a}} = \mathbf{a}_{IMU} - \hat{\mathbf{b}}_{acc}$ and $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_{IMU} - \hat{\mathbf{b}}_{gyr} - \mathbf{C}(\hat{\mathbf{q}}_g^b) \boldsymbol{\omega}_{ig}^s$.

These equations are simply integrated using the first-order trapezoidal rule given the high data rate $f_{IMU} = 100$ Hz of the IMU.

3.2.2 Covariance propagation

The System of equations (3.1) can be modeled by a nonlinear function f_V such that

$$\dot{\mathbf{x}}_V = \mathbf{f}_V(\mathbf{x}_V, \mathbf{n}_{IMU}) \quad (3.8)$$

with $\mathbf{n}_{IMU} = \left[\mathbf{n}_{gyr}^T \quad \mathbf{n}_{bgyr}^T \quad \mathbf{n}_{acc}^T \quad \mathbf{n}_{bacc}^T \right]^T$ the IMU process noise of covariance matrix \mathbf{Q}_{IMU} . The function \mathbf{f}_V depends on inertial measurements that change through time, thus it is not stationary.

In an EKF scheme, \mathbf{f}_V is linearized with respect to the estimated state vector $\hat{\mathbf{x}}_V$ and one can write the state estimation error $\delta\mathbf{x}_V = \mathbf{x}_V - \hat{\mathbf{x}}_V$ as

$$\dot{\delta\mathbf{x}}_V = \mathbf{F}_V \delta\mathbf{x}_V + \mathbf{G}_V \mathbf{n}_{IMU}, \quad (3.9)$$

$$\text{where } \mathbf{F}_V = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}\wedge] & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\mathbf{C}(\hat{\mathbf{q}}_g^b)^T [\hat{\mathbf{a}}\wedge] & \mathbf{0}_3 & -2[\boldsymbol{\omega}_{ig}^g\wedge] & -\mathbf{C}(\hat{\mathbf{q}}_g^b)^T & -[\boldsymbol{\omega}_{ig}^g\wedge]^2 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \quad (3.10)$$

$$\text{and } \mathbf{G}_V = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}(\hat{\mathbf{q}}_g^b)^T & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}. \quad (3.11)$$

The derivation of \mathbf{F}_V and \mathbf{G}_V is detailed in Subsection C.2.1. The dynamic error state vector $\delta\mathbf{x}_V = \left[\delta\boldsymbol{\theta}_g^b{}^T \quad \delta\mathbf{b}_{gyr}^T \quad \delta\mathbf{v}_{gb}^g{}^T \quad \delta\mathbf{b}_{acc}^T \quad \delta\mathbf{p}_{gb}^g{}^T \right]^T$ includes 15 components. Indeed, the error quaternion $\delta\mathbf{q}$ is defined by $\mathbf{q} = \hat{\mathbf{q}} \otimes \delta\mathbf{q}$, with \otimes the quaternion product. Using the small angle approximation, one can write $\delta\mathbf{q} \simeq \left[1 \quad \frac{1}{2}\delta\boldsymbol{\theta}^T \right]^T$ and thus $\delta\boldsymbol{\theta}$ is a correct minimal representation of the error quaternion.

In practice for a sliding-windows architecture with only one window, the static pose error of the camera for the last image $\delta\mathbf{x}_S = \left[\delta\boldsymbol{\theta}_g^{c_{k-1}}{}^T \quad \delta\mathbf{p}_{g^{c_{k-1}}}^g{}^T \right]^T$ is appended to the error state vector $\delta\mathbf{x} = \left[\delta\mathbf{x}_V^T \quad \delta\mathbf{x}_S^T \right]^T$ which then contains 21 components. Frame $\{c_{k-1}\}$ is that attached to the camera at the acquisition time t_{k-1} of image $k-1$. The addition of this static pose allows to deal with image processing delays.

The error covariance matrix at time t_{k-1} can be divided in four blocks

$$\mathbf{P}(t_{k-1}) = \begin{bmatrix} \mathbf{P}_{VV}(t_{k-1}) & \mathbf{P}_{VC}(t_{k-1}) \\ \mathbf{P}_{VC}(t_{k-1})^T & \mathbf{P}_{CC}(t_{k-1}) \end{bmatrix}. \quad (3.12)$$

$\mathbf{P}_{VV}(t_{k-1}) \in \mathbb{R}^{15 \times 15}$ represents the covariance of the dynamic error state vector, $\mathbf{P}_{CC}(t_{k-1}) \in \mathbb{R}^{6 \times 6}$ that of the pose of $\{c_{k-1}\}$, and $\mathbf{P}_{VC}(t_{k-1}) \in \mathbb{R}^{15 \times 6}$ is the matrix of the covariances between the dynamic and static error states. The creation of $\mathbf{P}_{CC}(t_{k-1})$ and $\mathbf{P}_{VC}(t_{k-1})$ during state augmentation is detailed in Subsection 3.2.4.

$P_{VV}(t)$ is propagated through the standard continuous-time Kalman covariance propagation of Equation (C.5) :

$$\dot{P}_{VV}(t) = F_V P_{VV}(t) + P_{VV}(t) F_V^T + G_V Q_{IMU} G_V^T. \quad (3.13)$$

$P_{CC}(t)$ remains constant through time as it represents the static states. The propagation of $P_{VC}(t)$ between t_{k-1} and t is derived as

$$P_{VC}(t) = E \left[\delta x_V(t) \delta x_S(t)^T \right], \quad (3.14)$$

where $\delta x_S(t) = \delta x_S(t_{k-1})$ and, from Equation (C.2),

$$P_{VC}(t) = e^{F_V(t-t_{k-1})} E[\delta x_V(t_{k-1}) \delta x_S(t_{k-1})^T] \quad (3.15)$$

$$P_{VC}(t) = e^{F_V(t-t_{k-1})} P_{VC}(t_{k-1}). \quad (3.16)$$

3.2.3 Filter update

The measurement can be predicted from this model, the knowledge of $p_{gl_j}^g$ and the a priori state estimate \hat{x}_V as

$$\hat{z}_j = h_j(\hat{x}) = \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} \hat{x}_{cl_j}^c \\ \hat{y}_{cl_j}^c \end{bmatrix} \quad (3.17)$$

with $\hat{p}_{cl_j}^c = \begin{bmatrix} \hat{x}_{cl_j}^c & \hat{y}_{cl_j}^c & \hat{z}_{cl_j}^c \end{bmatrix}^T = C(\hat{q}_g^c)(p_{gl_j}^g - \hat{p}_{gc}^g)$.

The innovation $\delta z_j = z_j - \hat{z}_j$ can then be linearized as shown in Subsection C.2.2 to update the EKF

$$\delta z_j \simeq H_j \delta x + n_j = H_{j,\theta} \delta \theta_g^c + H_{j,p} \delta p_{gc}^g + n_j, \quad (3.18)$$

$$\text{with } H_{j,\theta} = \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} I_2 & -\hat{z}_j \end{bmatrix} \left[C(\hat{q}_g^c)(p_{gl_j}^g - \hat{p}_{gc}^g) \wedge \right], \quad (3.19)$$

$$H_{j,p} = \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} I_2 & -\hat{z}_j \end{bmatrix} C(\hat{q}_g^c), \quad (3.20)$$

$$\text{and } H_j = \begin{bmatrix} \mathbf{0}_{2 \times 15} & H_{j,\theta} & H_{j,p} \end{bmatrix}. \quad (3.21)$$

By concatenating in columns all the innovations δz_j and the Jacobian matrices of the measurements H_j for each point j , one can work out the overall innovation $\delta z = H \delta x + n$. After computing the Kalman gain, this innovation allows to update the state and covariance estimated by the EKF.

From Equation (3.18), one can note that the innovation δz_j only depends on the static state error $\delta x_S = \begin{bmatrix} \delta \theta_g^{cT} & \delta p_{gc}^{gT} \end{bmatrix}^T$. The ability of the measurement to update the dynamic state x_V is only due to the fact that the correlation matrix $P_{VC}(t)$ is not null.

3.2.4 State management

After each filter update, the old camera error attitude estimate $\delta\hat{\theta}_g^{c_{k-1}}$ and error position estimate $\delta\hat{p}_{g_{c_{k-1}}}^g$ are removed from the EKF error state vector, thus bringing it back to 15 components. The corresponding lines and columns of the error covariance matrix are also removed. Immediately, a new image is acquired and the new static camera error pose estimates $\delta\hat{\theta}_g^{c_k}$ and $\delta\hat{p}_{g_{c_k}}^g$ are appended to form the state vector with 21 components again.

At each image acquisition, the computation of the new camera pose estimate from the vehicle pose estimate is done using the known position and orientation of the camera frame $\{c\}$ with respect to $\{b\}$:

$$\hat{q}_g^c = \hat{q}_g^b \otimes q_b^c \quad (3.22)$$

$$\hat{p}_{gc}^g = \hat{p}_{gb}^g + C(q_g^b)^T p_{bc}^b \quad (3.23)$$

Based on these two equations and as demonstrated in Subsection C.2.3, the Jacobian matrix J allows to augment the error covariance matrix from $\mathbb{R}^{15 \times 15}$ to $\mathbb{R}^{21 \times 21}$

$$P(t) \leftarrow \begin{bmatrix} I_{15} \\ J \end{bmatrix} P(t) \begin{bmatrix} I_{15} \\ J \end{bmatrix}^T \quad (3.24)$$

$$J = \begin{bmatrix} C(q_g^c) & \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 3} \\ -C(q_g^b)^T [p_{bc}^b \wedge] & \mathbf{0}_{3 \times 9} & I_3 \end{bmatrix}. \quad (3.25)$$

3.3 POINT-BASED SOFTWARE SIMULATION TESTS

This section investigates the performance of the *Lion* navigation filter in a point-based software simulation where ideal image-to-map matching is assumed: measurements are still corrupted by the extraction noise n_j from Equation (3.6) but there are no outlier matches. This flawless environment allows to verify if the tight absolute visual-inertial estimation structure of *Lion* can meet the 100-m pinpoint landing accuracy requirement over any terrain topographical conditions, before implementing image processing in the loop.

3.3.1 Simulation environment

The trajectory generated corresponds to that of an approach phase for a Moon landing. It lasts 80 seconds, starts at a 2-km altitude and ends 10 m above the ground (De-laune et al., 2010). Guidance is based on that of Apollo lunar module (Sostaric and Rea, 2005). Inertial data are generated through an IMU model calibrated to match the aerospace-class performances expected on planetary landing missions of this kind and presented in Subsection B.2.1. The lunar surface is modeled by a cloud of points, which are the assumed visual navigation landmarks used here. These points are distributed randomly according to a continuous uniform distribution within a 2D sampling grid horizontally, and vertically within a specified elevation range. To test robustness of the filter to various topographical conditions, three surfaces with elevation ranges of 0, 100 or 1000 m were created. Figure 3.2 shows the final part of the descent camera trajectory in this point-based simulation environment.

1024 × 1024 descent images were generated by projecting the surface points onto the focal plane of a pinhole camera model covering a 70-deg field of view and then

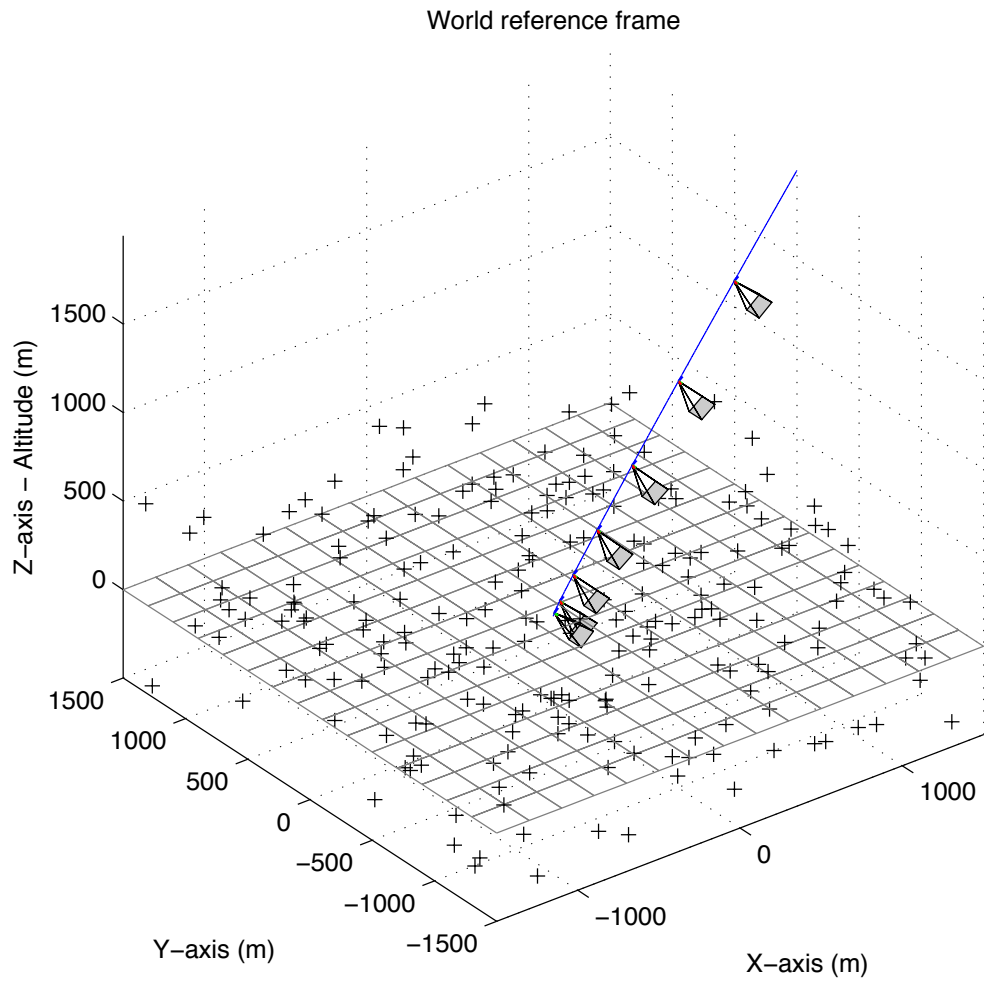


Figure 3.2 – Trajectory of the camera in the point-based simulation environment. The lunar surface is modeled by a cloud of points randomly sampled within a 2D horizontal grid and over various elevation ranges.

adding the Gaussian normalized image measurement noise n_j of Equation (3.6) with a standard deviation equivalent to 1 pixel. The acquisition rates are 1 Hz for the camera and 100 Hz for the IMU

3.3.2 Performance

100 Monte Carlo simulations were run over each of the three simulated terrain topographies. The error on the state of the system is initialized on each axis using zero-mean Gaussian distributions with 3σ values of 1 deg in attitude, 10 m.s⁻¹ in velocity, and 100 m in position at the 2-km start altitude, where σ is the standard deviation. These values are regarded as representative of the error distribution at this stage of a pinpoint landing attempt.

Table 3.1 shows the results in terms of dispersion error at touchdown. The total dispersion of a vector estimate can be evaluated by the Root Mean Square (RMS) value of the vector difference with respect to the mean vector estimate. Complete details about navigation error statistics can be found in Section B.4. Whatever the terrain topography, the 3-RMS value for total dispersion is always below 22 m in position, 1.4 m.s⁻¹ in velocity and 0.4 deg in attitude. The mean errors are below 0.8 m, 0.05 m.s⁻¹ and 0.04 deg respectively, which are negligible with respect to the corresponding dispersions. These dispersions are of the same order of magnitude with the flat surface and the one with 100-m elevation range, about 20 m in position (3 RMS). They actually improve with the 1000-m elevation range surface down to about 2 m (3 RMS). The better performance in this latter case can directly be related to the number landmarks visible during the descent and plotted in Figure 3.3. In the point-based simulation, the landmark altitudes were scattered between -500 m and +500 m. Thus when touchdown happens at 0 m, there are still many landmarks visible with 1000-m elevation range while there might be only a few for the 100-m case and none for the flat terrain. Eventually, all those performances are largely below the 100-m pinpoint landing requirement and show the compatibility of the *Lion* EKF to handle any terrain topography.

Figure 3.4, Figure 3.5, and Figure 3.6 show the evolution of the estimation error and the 3σ values propagated by the filter for each axis over the terrain with the 100-m elevation range, respectively for the attitude, velocity and position of frame $\{b\}$ tied to the vehicle with respect to the planet frame $\{g\}$. Note that on these plots the 3σ envelope is that of the run which had the largest final standard deviation, it is displayed for illustration purposes only. These plots show that the dispersion of the error is actually minimal around $t_V = 4030$ s, thus about 30 s before touchdown, and later starts to rise up again. The explanation is that because of the limited landmark density, less and less of them are visible as the lander progresses down its trajectory, as illustrated in Figure 3.3. After t_V , there are not enough absolute image measurements to correct the inertial drift, this is the end of the so-called *visual phase* and the error keeps increasing until touchdown. This phenomenon is somewhat limited with the 1000-m elevation range because enough landmarks remain visible in our point-based simulation as mentioned earlier, but it is very representative of what happens during a real mission as the limited resolution of orbital images also limits the maximum landmark surface density and less and less of them are visible when the lander goes down. The actual performance at the end of the visual phase at t_V are presented in Table 3.2. The performances before the inertial drift is significantly better than the performance at touchdown, with 3-RMS value for total dispersion below 2.2 m in

Table 3.1 – Dispersion of the navigation error at touchdown with various point-based terrain elevation ranges. The dispersion over each axis is measured with the 3σ value for scalar components, where σ is the standard deviation. The total dispersion of the vector quantities is measured with the 3-RMS value.

Elevation range		0 m	100 m	1000 m
$\delta\theta_{g,x}^b$	(roll, deg)	0.2	0.3	0.1
$\delta\theta_{g,y}^b$	(pitch, deg)	0.2	0.2	0.1
$\delta\theta_{g,z}^b$	(yaw, deg)	0.1	0.1	0.03
$\delta\theta_g^b$	(total, deg)	0.3	0.4	0.2
$\delta v_{g,b,x}^g$	(downrange, m.s ⁻¹)	0.8	1.0	0.3
$\delta v_{g,b,y}^g$	(crossrange, m.s ⁻¹)	0.6	0.9	0.2
$\delta v_{g,b,z}^g$	(height, m.s ⁻¹)	0.5	0.4	0.1
δv_{gb}^g	(total, m.s ⁻¹)	1.1	1.4	0.4
$\delta p_{g,b,x}^g$	(downrange, m)	13.4	16.3	2.2
$\delta p_{g,b,y}^g$	(crossrange, m)	10.3	13.4	0.9
$\delta p_{g,b,z}^g$	(height, m)	6.9	6.1	1.4
δp_{gb}^g	(total, m)	18.3	22.0	2.7

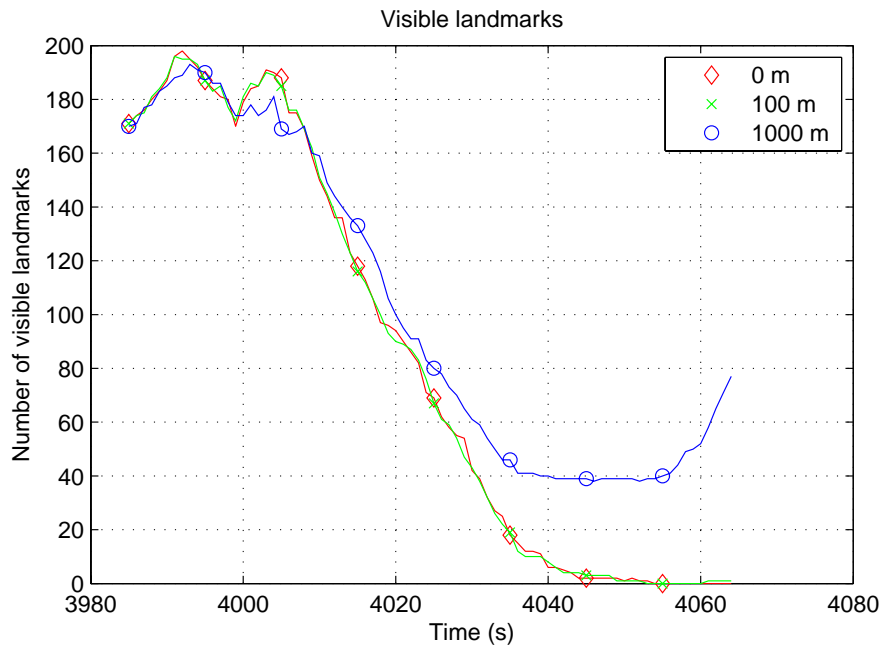


Figure 3.3 – Evolution of the number of absolute measurements over the point-based terrain for the different surface elevation ranges.

position, $0.7 \text{ m}\cdot\text{s}^{-1}$ in velocity and 0.4 deg in attitude. Only in the case with the 1000-m elevation range is the inertial drift negligible less important.

Table 3.2 – Dispersion of the navigation error at the end of the visual phase with various point-based terrain elevation ranges. The dispersion over each axis is measured with the 3σ value for scalar components, where σ is the standard deviation. The total dispersion of the vector quantities is measured with the 3-RMS value.

Elevation range		0 m	100 m	1000 m
$\delta\theta_{g,x}^b$	(roll, deg)	0.2	0.3	0.2
$\delta\theta_{g,y}^b$	(pitch, deg)	0.2	0.2	0.1
$\delta\theta_{g,z}^b$	(yaw, deg)	0.1	0.1	0.04
$\delta\theta_g^b$	(total, deg)	0.3	0.4	0.2
$\delta v_{gb,x}^g$	(downrange, $\text{m}\cdot\text{s}^{-1}$)	0.3	0.4	0.3
$\delta v_{gb,y}^g$	(crossrange, $\text{m}\cdot\text{s}^{-1}$)	0.3	0.5	0.5
$\delta v_{gb,z}^g$	(height, $\text{m}\cdot\text{s}^{-1}$)	0.3	0.3	0.3
δv_{gb}^g	(total, $\text{m}\cdot\text{s}^{-1}$)	0.5	0.7	0.6
$\delta p_{gb,x}^g$	(downrange, m)	1.1	1.4	1.5
$\delta p_{gb,y}^g$	(crossrange, m)	0.9	1.1	1.3
$\delta p_{gb,z}^g$	(height, m)	0.6	0.8	0.9
δp_{gb}^g	(total, m)	1.6	1.9	2.2

CHAPTER CONCLUSION

This chapter developed the implementation of the tight absolute visual-inertial EKF used in the *Lion* navigation system. It keeps the pose estimate at the acquisition time of the last image as an additional static state to take into account time delays associated with absolute image matching. We implemented it in a point-based simulation assuming ideal absolute measurements, and testing the final part of the trajectory over surface elevation ranges from 0 to 1000 m. The dispersion was below 22 m (3-RMS) at touchdown on all topographies and there was no apparent link between the surface elevation range and the quality of the estimation, which appears more affected by the number of landmark measurements. Indeed, the dispersion actually fell down to 2.2 m (3-RMS) at the end of a so-called visual phase, before fewer if any mapped landmarks were left visible and a final inertial drift started to increase the navigation error. The most important result is that the *Lion* EKF estimation architecture is capable to reach pinpoint landing accuracy whatever the terrain structure.

From these ideal results, Chapter 4 now tackles the issue of processing actual images to recognize mapped landmarks in them and intends to keep the same level of performance. It details the design of several new absolute geometric image matching methods and compares their performances with image processing in the loop.

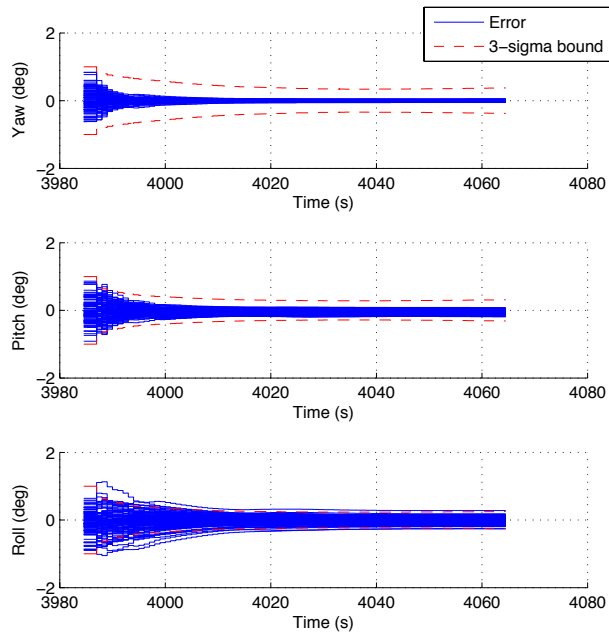


Figure 3.4 – Attitude estimation error during 100 Monte Carlo runs over the point-based terrain with 100-m elevation range. The red dotted line shows the 3σ bounds propagated by the EKF for the run which had the largest final standard deviation.

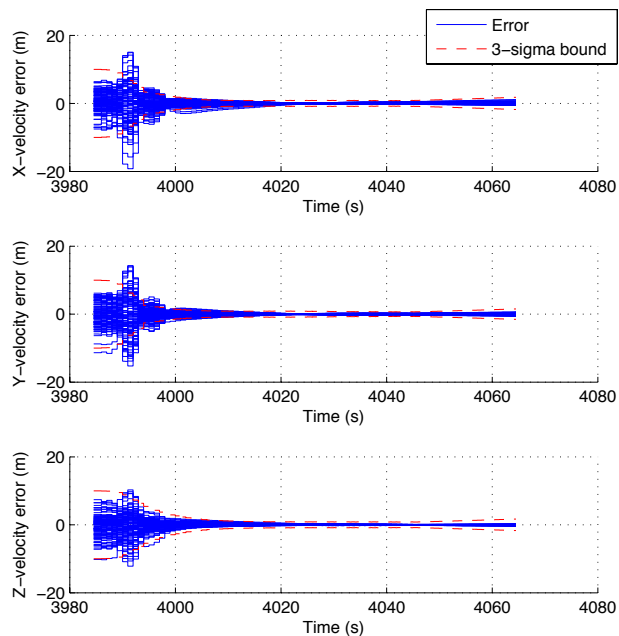


Figure 3.5 – Velocity estimation error during 100 Monte Carlo runs over the point-based terrain with 100-m elevation range. The red dotted line shows the 3σ bounds propagated by the EKF for the run which had the largest final standard deviation.

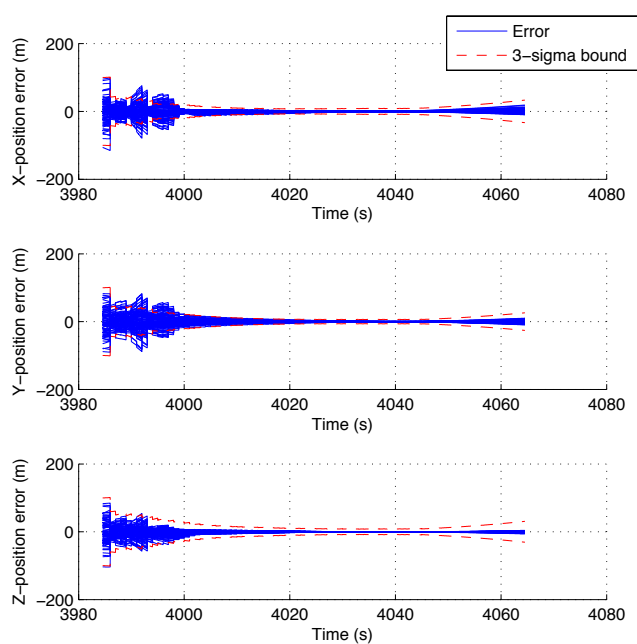


Figure 3.6 – Position estimation error during 100 Monte Carlo runs over the point-based terrain with 100-m elevation range. The red dotted line shows the 3σ bounds propagated by the EKF for the run which had the largest final standard deviation.

4

2D/3D GEOMETRIC VISUAL LANDMARK RECOGNITION

CONTENTS

4.1	IMAGE-BASED SOFTWARE SIMULATION ENVIRONMENT	47
4.2	LANDMARK RECOGNITION USING IMAGE REPROJECTION	48
4.2.1	Offline map	48
4.2.2	Online image processing	49
4.2.3	Navigation performance	53
4.2.4	Discussion: the repeatability issue	58
4.3	REPEATABILITY IMPROVEMENT THROUGH LANDMARK SCALE MANAGEMENT	59
4.3.1	Scale-space model	59
4.3.2	Scale-invariant extractor selection	60
4.3.3	Scale reprojection formula	61
4.3.4	Comparison of repeatability rates	63
4.4	LANDMARK RECOGNITION USING SCALE-AUGMENTED IMAGE REPROJECTION	64
4.4.1	Offline map	65
4.4.2	Online image processing	65
4.4.3	Navigation performance	66

VISUAL landmark recognition aims at matching surface features imaged by a descent camera with a map stored in the computer memory on board the spacecraft. In *Lion*, no assumption about the topography of the terrain shall be made in this recognition step either so that the 2D/3D absolute pairings between the image and the physical world can be used to update the filter independently of the terrain relief. Absolute visual navigation should also be performant at low altitude to delay the blindness phenomenon and the associated inertial error drift noticed in Chapter 3 and imposed by the limited resolution of the orbital images. After reviewing the state of the art in Section 2.1, we narrowed down our focus on matching methods using geometric descriptors to identify generic landmarks. These approaches are very promising indeed in terms of simplicity, cost efficiency, and enhanced illumination robustness.

Section 4.1 describes the software simulator using computer-generated lunar images which was designed to assess the performance of the whole navigation system with image processing in the loop. Two geometric matching methods using standard Harris feature points as landmarks are then described and evaluated in Section 4.2. By using a landmark reprojection step aided by the navigation filter, they manage to avoid the planar rectification often met in literature and the associated flat-world assumption. Though, both these methods suffer from a low landmark repeatability problem highlighted in the analysis of performance. We thus evaluate in Section 4.3 the benefit of using a scale-invariant extractor instead of the classical Harris operator in terms of repeatability rate. More remarkably we also show that the scale of a landmark can be used as an additional descriptor to match it with the map, in similar way to the radius for craters. Based on these results, Section 4.4 designs a new geometric landmark matching method based on scale-invariant Harris-Laplace orbital image landmarks. The navigation filter is efficiently used to predict the image region and scale in which to search for each landmark, followed by a selection process to identify the non-ambiguous ones suitable for matching.

4.1 IMAGE-BASED SOFTWARE SIMULATION ENVIRONMENT

Realistic descent and orbital lunar images have been rendered using the *PANGU* planetary scene generator described in Subsection 2.3.1. The DEMs made with the altimetry data collected by the NASA Lunar Reconnaissance Orbiter (LRO) spacecraft which has been orbiting the Moon since 2009 were used as a DEM base in *PANGU* (Smith et al., 2010). Their horizontal resolution was scaled down to avoid seeing the DEM artefacts and the elevation range was scaled up to obtain two height variation cases of 100 m and 500 m so as to perform a preliminary robustness test to 3D terrain topography. These DEM bases were augmented with higher-resolution artificial DEMs through a system of layers to keep the images realistic when approaching the surface. On each additional layer, new craters are scattered using lunar distribution models. An image extracted from the generated descent sequence is shown in Figure 4.1. It covers a field of view of 70 degrees with 1024×1024 -pixels on an 8-bit sensor. Each image pixel is disrupted with a zero-mean Gaussian noise of 1 intensity level. These parameters match those of planetary landing cameras discussed in Subsection B.2.3. The simulated orbital images were taken from a 50-km altitude with 5064×5064 sensor behind a lens providing a 2.85-degree field of view to simulate the Narrow Angle Camera (NAC) on-board LRO which images all areas at polar latitudes. The same DEM used for image generation is used for map-building, with a resolution of 1 m,

and there is no misalignment error between the orbital image and the DEM yet at this stage for preliminary testing. The elevation of the Sun above the horizon at the targeted landing site was 15 degrees, and a 20-deg azimuth difference was implemented between the orbital and descent conditions as a preliminary test to illumination robustness.



Figure 4.1 – Descent image generated with PANGU

An 80-second lunar landing approach trajectory starting at 2 km of altitude and based on the Apollo guidance scheme was implemented, like in Section 3.3. Likewise too, inertial data were simulated by software through an IMU model calibrated to match the aerospace-class performances discussed in Subsection B.2.1. The acquisition rates are 1 Hz for the camera and 100 Hz for the IMU.

4.2 LANDMARK RECOGNITION USING IMAGE REPROJECTION

In this section we propose two robust image-to-map matching algorithms based on a common core architecture for generic landmarks using geometric descriptors and making no assumption about the topography of the terrain. Both methods employ Harris corner features to process both the descent and the orbital images. Unlike craters, these corners are generic enough to be detected on any planetary terrain with textures. Furthermore, they have already shown robust performance for a wide spread of computer vision problems (Szeliski, 2011).

4.2.1 Offline map

Both algorithms make use of a map stored in the on-board memory and which consists in the 3D lunar coordinates $\{p_{gl_j}^g\}_{j \in \llbracket 1; N \rrbracket}$ in $\{g\}$ of the N Harris landmarks ex-

tracted in the orbital image. Knowing the position and attitude of the satellite which took the images in $\{g\}$ as well, along with the calibration matrix of its camera, one can determine the 3D rays going from the optical center of the camera to each landmark (Hartley and Zisserman, 2003). The intersections of these rays with the DEM mesh is recovered through 2D interpolation. The overall map-building process is illustrated in Figure 4.2. The map is thus simply a $N \times 3$ matrix which contains only the 3D coordinates of each landmark.

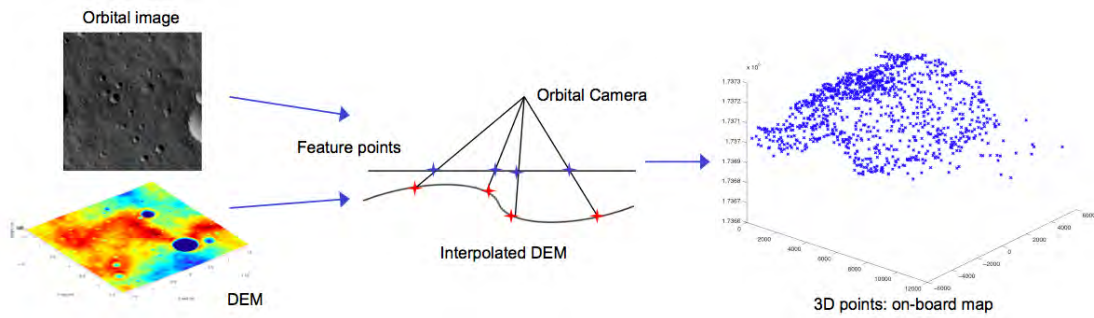


Figure 4.2 – Map generation process. The 3D landmark coordinates are recovered by tracing the rays back-projected from the Harris features extracted in the orbital image, and then interpolating them with the DEM.

4.2.2 Online image processing

The online part of each algorithm goes in four steps:

1. Online extraction of Harris corners,
2. Projection of the 3D map points onto the focal plane predicted by the filter,
3. Putative matching process,
4. Robust matching based on the RANdom Sample Consensus algorithm (RANSAC).

There are actually two steps of matching, 3 and 4. Step 4 is called *robust* because a RANSAC scheme is implemented and aims at detecting the false putative matches of step 3, based on the calibrated camera model (Fischler and Bolles, 1981). Those putative matches are first obtained by searching for similarities between two sets of image features. The first set is obtained in step 1 through the extraction of Harris corners in the actual descent image. The second set is an image prediction aided by the filter through a reprojection of the map landmarks onto the focal plane of the camera using the a priori pose estimate in the EKF. The difference between the two algorithms happens in step 3, where putative matches can be obtained either through a projected shape context or a generalized Hough transform.

Compared to the homography-based rectification made in Pham et al. (2012) which is based on a planar assumption, the landmark reprojection of step 2 uses the topography description contained in the DEM of the terrain and thus brings the robustness to 3D surfaces.

3D model projection

Using the a priori camera pose estimate $(\hat{q}_g^c, \hat{p}_{g^c}^g)$ propagated with the inertial measurements in the filter, and the known camera calibration model, the 3D map landmarks $\{p_{gl_j}^g\}$ stored in memory can be projected onto the expected focal plane of the image to get a prediction of the online image feature point locations $\{\hat{z}_j\}$. Along with the Harris features $\{z_k\}$ extracted in the actual descent image, this eventually leads to two sets of image features, one measurement and one prediction, which shall be matched. The complexity of the matching can be imagined from Figure 4.3, which yet shows the two sets in the ideal case of a null estimation error. Some points are not repeated, i.e. they were detected in the orbital image but not in the descent image. Oppositely, some other are false alarms which appear in the descent image where nothing was detected in the orbital data. Even for the landmarks which seem successfully repeated, the measured and predicted locations appear slightly different due to image extraction noise. This low repeatability issue is the most significant difference compared to the ideal point-based simulation of Section 3.3 the algorithms have to adapt to.

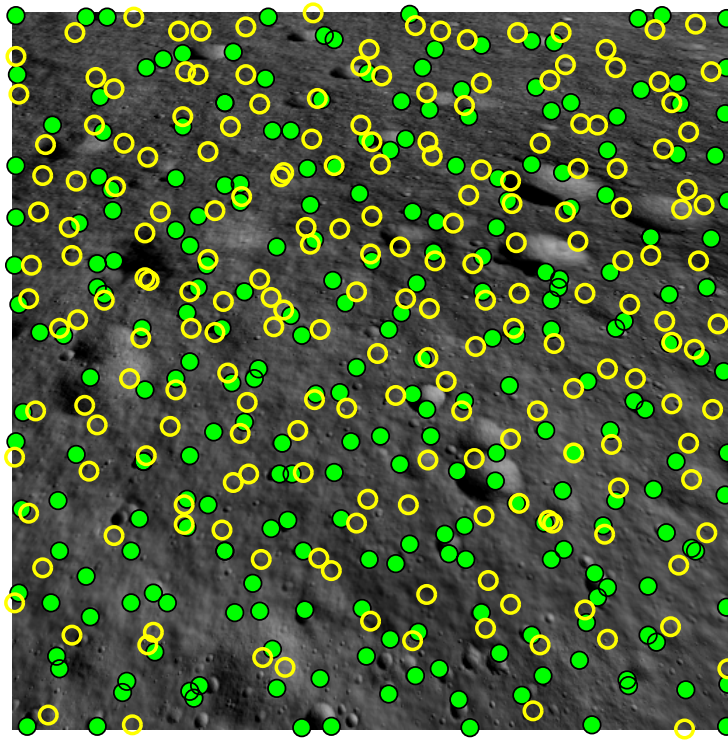


Figure 4.3 – Harris image features extracted online and predicted by the filter. The descent feature measurements are the green dots and the reprojected filter predictions are the yellow circles.

Putative matching with projected shape context

Out of all the possible match combinations between the image features and the map, the putative matching process aims at identifying the most promising ones. Following the Shape Context (SC) description for object recognition introduced by Belongie et al. (2000) and first applied to planetary landing by Pham et al. (2009), we characterize

each feature point by the geometric distribution of its neighbours in terms of distance and polar angle, as shown in Figure 4.4. The number of neighbours in each quadrant is counted within a region comprised between a minimum radius br and maximum radius pr , and stored into an histogram.

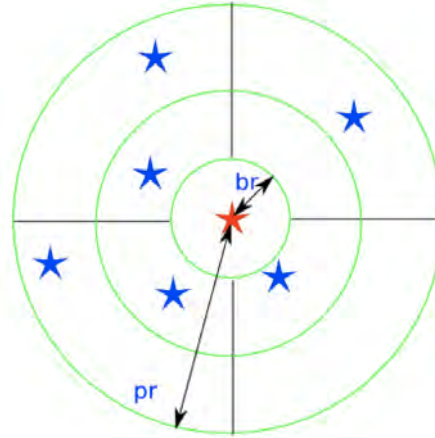


Figure 4.4 – Shape context signature (Pham et al., 2009)

Signatures of the descent features are compared one to one with that of the re-projected feature predictions within an image uncertainty area derived from the state covariance in the filter. The minimized criterion for comparison is the χ^2 distance. All pairs of features between the descent and predicted set for which the χ^2 distance is below a certain threshold are considered as putative matches.

Putative matching with generalized Hough transform

As an alternative to SC, we can approximate the image error between the descent features and their predictions by a single global translation within the image plane. This is not rigorously true because of perspective distortion effects in the image but should the translation be searched within quantized grid with a quantization step large enough, a voting accumulation effect will happen to highlight the correct translation. This approach is known as Generalized Hough Transform (GHT) in the object recognition field (Eric and Grimson, 1991) and makes the basis of operational real-time infrared missile guidance systems (Duclos et al., 2009).

The matching process is designed as follows:

1. Each possible match between a measured descent feature and a predicted one defines a possible 2D translation;
2. Accumulate all translations:
 - for which the norm is below a certain threshold r derived from the camera pose covariance,
 - after quantization with a step length s derived from the expected perspective distortion effects;
3. Select the peak of the accumulator: it yields the estimated discrete translation;
4. Shift the landmark predictions according to the estimated translation and match them with the closest descent point.

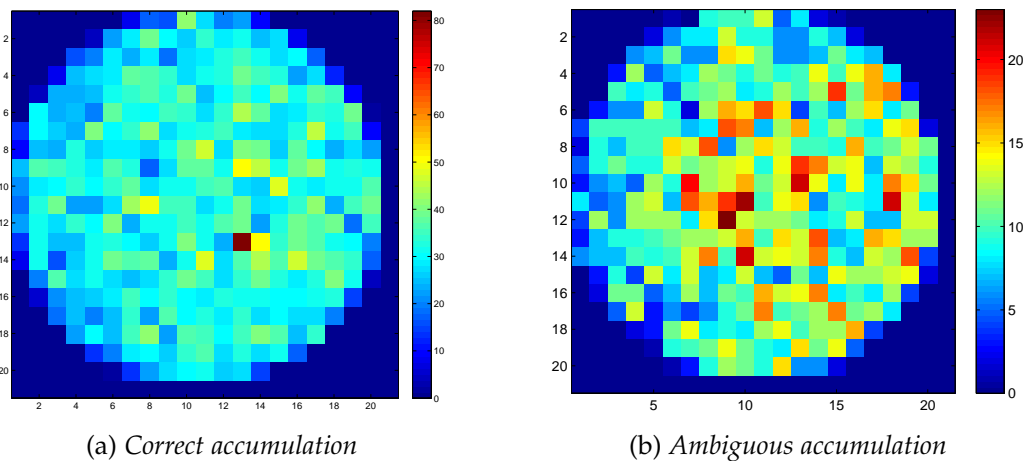


Figure 4.5 – GHT accumulation grid in both the correct and ambiguous cases. Ambiguity happens when very close to the ground and not enough map landmark information is available to create the accumulation effect.

5. Accept the match if the distance to this closest point is not above a threshold.

The quantized grid in which translation votes are accumulated can lead to two cases shown in Figure 4.5: a correct or an ambiguous accumulation. The ambiguous case happens when the camera is so close to the ground that not enough orbital landmark information is available in the field of view to create the accumulation effect, or when the parameters r and s are badly chosen.

RANSAC-based robust matching

Fischler and Bolles (1981) introduced the RANSAC algorithm which aims at fitting a model to experimental data and providing the associated set of inliers and outliers to this model. It is now a proven technique in real-time visual odometry for terrestrial robotics. In our case, the model used in RANSAC is the full 3×4 camera projection matrix (Hartley and Zisserman, 2003), and the experimental data taken as inputs are the potential 2D/3D putative matches. The calibrated camera projection matrix can be computed in a closed form from a minimal set of 3 matches as shown in Fischler and Bolles (Fischler and Bolles, 1981) too. Three correspondences actually bring multiple solutions, but we select that for which camera position is the closest to the a priori position estimate in the filter. The aim of this final step is to get the set of matches which correspond to camera projection matrix with the highest number of inliers, and exclude the outliers from the filter update. Let us emphasize that with the tight visual-inertial integration scheme described in Chapter 3, *Lion* does not use the estimate of the state which is computed within RANSAC but only the associated set of inliers for the EKF update.

The RANSAC-based implementation in *Lion* operates as follows from a set S of N 2D/3D putative matches.

1. A set s of 3 matches is randomly selected from S .
2. If there is any degenerated configuration, for instance if the three points are aligned (Hartley and Zisserman, 2003), in the 2D points of s , then do step 1 again.
3. Compute the possible camera projection matrices associated to s .

4. Select the matrix for which the position is closest to the filter estimate. One additional benefit of filtering is to help choosing among the multiple analytical camera solutions.
5. Determine the inliers for the projection matrix selected in step 4. Namely, inliers among putative matches are those for which the 3D landmark image reprojection is within a certain image distance threshold of the associated descent feature. The image threshold is selected as $3\sigma_{im}$, where σ_{im} is the standard deviation of image noise.
6. If the number of inliers is greater than the previous reference, then store the associated inlier vector as the new reference.
7. Back to step 1 until the maximum set of iteration is reached.

The maximum number of iterations in step 7 is determined adaptively to ensure with a probability $p = 0.99$ that at least one of the set s is free from outliers (Hartley and Zisserman, 2003). Once the maximum number of iterations is reached, the last inlier vector stored in step 6 is outputted as final matches to update the navigation filter. It corresponds to the camera projection matrix having the largest number of inliers.

4.2.3 Navigation performance

Putative matching method comparison

Both the SC and GHT methods introduced earlier to get putative matches between the descent image features and the map were compared to select the most performant one to work with the *Lion* EKF scheme described in Chapter 3. Based on the software simulation environment presented in Section 4.1, the on-board map was generated from a mosaic of orbital images covering the $16 \times 16 \text{ km}^2$ area seen during the descent trajectory. 4000 landmarks were extracted in the whole orbital mosaic, and 4000 others in the orbital image covering the surroundings of the landing site which will be visible during the last part of the trajectory. 250 features are extracted per descent image. The parameters of the SC signature, shown in Figure 4.4, are a minimum radius $br = 10$ pixels, a maximum radius $pr = 100$ pixels, 10 rings and 20 wedges. For the GHT, a circular grid of radius = 100 pixels and discretized in squares with a step length $s = 10$ pixels was used.

Even though the performance of SC method appeared promising in the early point-based simulation tests (Delaune et al., 2011), it turned up to be very complex to tune its parameters to process successfully actual images. Such difficulty can only be due to the low landmark repeatability issue which locally affects the SC signatures of all landmarks within the radius pr from a false alarm or a non-repeated landmark. Figure 4.6 actually compares the number of landmarks outputted by the RANSAC robust matching step for SC and GHT, with the best achievable tuning for SC. The GHT method clearly achieves more matches than SC with 45 against 15 in average along the descent. These matches are also more distributed across the image, which improves the accuracy of the pose measurement process done in the filter. As a consequence, the convergence of the filter was empirically noticed to be a lot more stable and fast with the GHT rather than with SC. Thus, the GHT method is chosen for further statistical performance evaluation in the following.

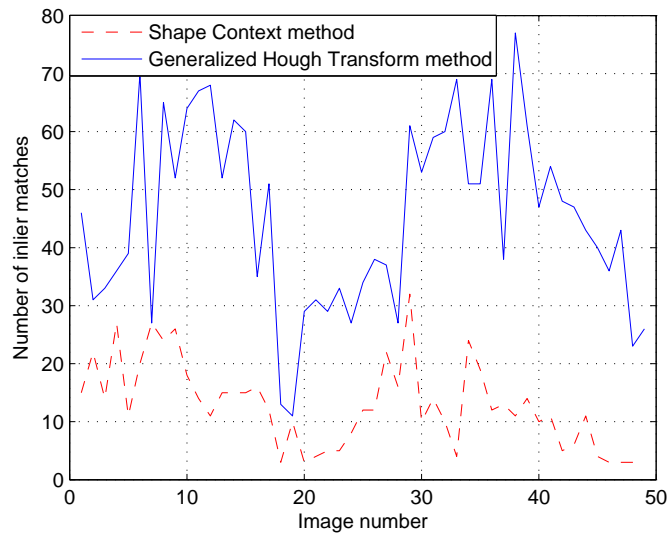


Figure 4.6 – Putative matching method comparison. The GHT method achieves more matches than SC with 45 against 15 in average.

GHT navigation performance

The results are obtained through a Monte Carlo performance analysis of the whole navigation filter with GHT putative matching. 200 Monte Carlo runs were performed first over the terrain with the 100-m relief variation. Like in Section 3.3, the initial errors were sampled according to independent zero-mean Gaussian distribution on each axis, with 3σ values equal to 1 degree in attitude, $10 \text{ m}\cdot\text{s}^{-1}$ in velocity, and 100 m in position at the 2-km start altitude.

Figure 4.7, Figure 4.8, and Figure 4.9 show the Monte Carlo results for the estimation error and the 3σ bound from the filter over each axis respectively for attitude, velocity and position of frame $\{b\}$ tied to the vehicle with respect to the global frame $\{g\}$ tied to the planetary surface. Figure 4.10 shows the evolution of matched landmarks with time.

Because of the limited resolution of orbital images, the visual measurement phase ends at $t = t_V = 4040 \text{ s}$ when not enough landmarks are visible to update the filter, and the inertial drift begins until touchdown at $t = t_{TD} = 4065 \text{ s}$.

A run is said to be convergent when the navigation error for any state is within the 3σ boundaries computed by the filter for this state. Convergence can be observed on the figures for 93.5 % of the runs. The performance was found quite sensitive to parameter choices for r and s . The overall statistics, including divergent runs, are given in Table 4.1. Position dispersion at the end of the visual phase falls below 67 meters on each axis and 89 m in total. The position error increases then until touchdown up to 156 m accuracy due to inertial drift. Like in Section 3.3, the mean errors are negligible. These results are not compliant with the 100-m accuracy requirement for pinpoint landing but they include the runs which diverge. That led to position errors of up to 400 m in one case, which alter the performance statistics strongly. Considering only the 93.5 % of runs which did converge, the statistics tremendously improve as shown in Table 4.2. Position errors falling below 8 meters on each axis at the end of the visual phase, 10 m total, and below 22 meters at touchdown, 26 m total. That means that if

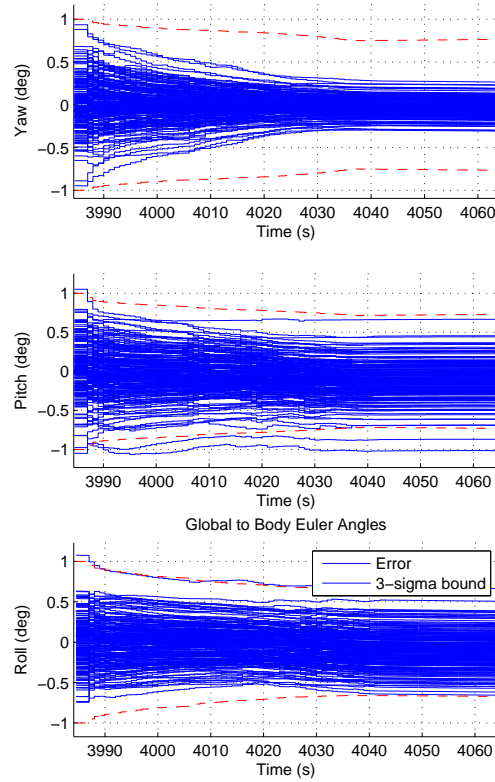


Figure 4.7 – GHT attitude estimation error during 200 Monte Carlo runs in image-based simulations. The red dotted line shows the 3σ bounds propagated by the EKF for the run which had the largest final standard deviation.

Table 4.1 – Dispersion of the navigation error with GHT at the end of the visual phase (V) and at touchdown (TD) for all runs on image-based simulations. The dispersion over each axis is measured with the 3σ value for scalar components, where σ is the standard deviation. The total dispersion of the vector quantities is measured with the 3-RMS value.

Time		t_V	t_{TD}
$\delta\theta_{g,x}^b$	(roll, deg)	0.7	0.3
$\delta\theta_{g,y}^b$	(pitch, deg)	0.7	0.4
$\delta\theta_{g,z}^b$	(yaw, deg)	0.3	0.2
$\delta\theta_g^b$	(total, deg)	1.0	0.5
$\delta v_{gb,x}^g$	(downrange, $\text{m}\cdot\text{s}^{-1}$)	1.7	1.7
$\delta v_{gb,y}^g$	(crossrange, $\text{m}\cdot\text{s}^{-1}$)	0.7	0.9
$\delta v_{gb,z}^g$	(height, $\text{m}\cdot\text{s}^{-1}$)	1.3	1.1
δv_{gb}^g	(total, $\text{m}\cdot\text{s}^{-1}$)	2.2	2.2
$\delta p_{gb,x}^g$	(downrange, m)	66.3	120.5
$\delta p_{gb,y}^g$	(crossrange, m)	26.8	50.4
$\delta p_{gb,z}^g$	(height, m)	52.0	85.2
δp_{gb}^g	(total, m)	88.4	155.9

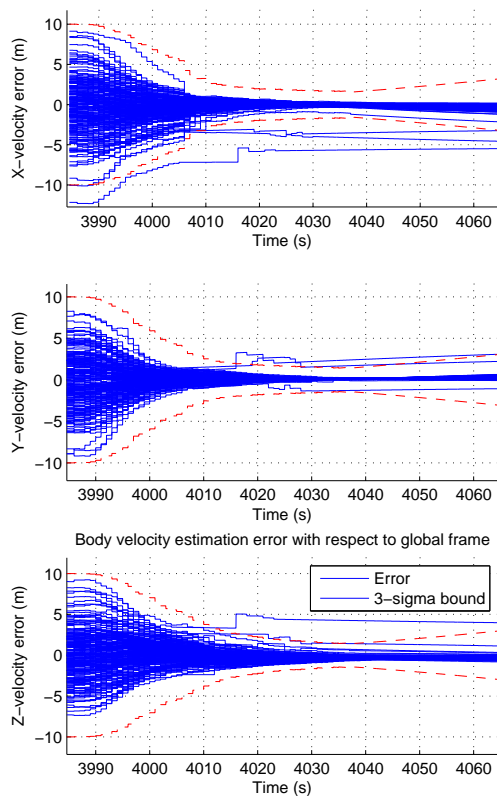


Figure 4.8 – GHT velocity estimation error during 200 Monte Carlo runs in image-based simulations. The red dotted line shows the 3σ bounds propagated by the EKF for the run which had the largest final standard deviation.

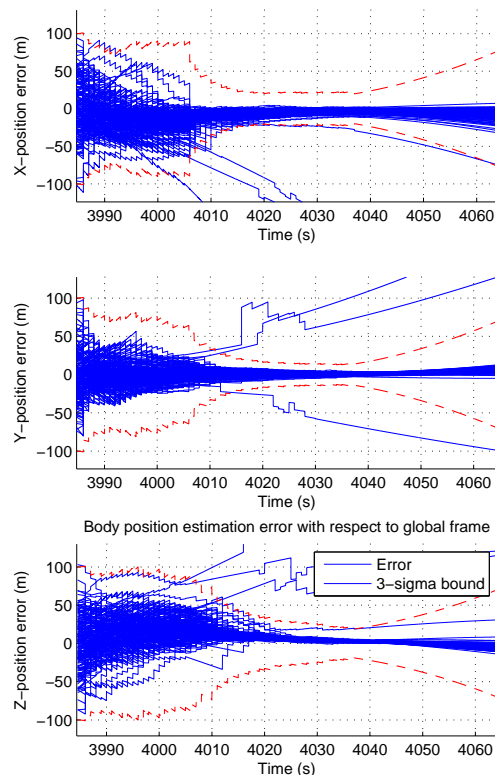


Figure 4.9 – GHT position estimation error during 200 Monte Carlo runs in image-based simulations. The red dotted line shows the 3σ bounds propagated by the EKF for the run which had the largest final standard deviation.

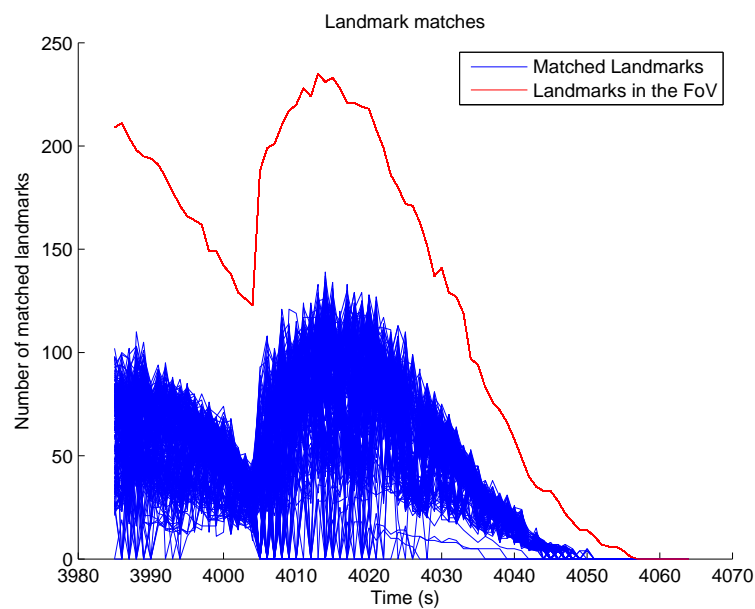


Figure 4.10 – Evolution of the number of absolute measurements during 200 Monte Carlo runs in image-based simulations. The red line shows the total number of landmarks in the field of view.

we manage to find the origin of the diverging runs and solve the related issue, the *Lion* navigation system can provide very promising performances for pinpoint landing.

Table 4.2 – Dispersion of the navigation error with GHT at the end of the visual phase (V) and at touchdown (TD) for converging runs only on image-based simulations. The dispersion over each axis is measured with the 3σ value for scalar components, where σ is the standard deviation. The total dispersion of the vector quantities is measured with the 3-RMS value.

Time		t_V	t_{TD}
$\delta\theta_{g,x}^b$	(roll, deg)	0.6	0.3
$\delta\theta_{g,y}^b$	(pitch, deg)	0.6	0.3
$\delta\theta_{g,z}^b$	(yaw, deg)	0.3	0.2
$\delta\theta_g^b$	(total, deg)	0.9	0.5
$\delta v_{gb,x}^s$	(downrange, m.s ⁻¹)	0.4	0.7
$\delta v_{gb,y}^s$	(crossrange, m.s ⁻¹)	0.2	0.3
$\delta v_{gb,z}^s$	(height, m.s ⁻¹)	0.7	0.4
δv_{gb}^s	(total, m.s ⁻¹)	0.8	0.9
$\delta p_{gb,x}^s$	(downrange, m)	7.4	21.8
$\delta p_{gb,y}^s$	(crossrange, m)	4.5	7.0
$\delta p_{gb,z}^s$	(height, m)	4.6	10.8
δp_{gb}^s	(total, m)	9.8	25.3

Robustness to 3D topography

Lion was tested over rugged terrains with stronger 3D topographies using the DEM with the 500-m terrain elevation amplitude by comparing the performance of the version of *Lion* presented here versus an alternative version where a flat world hypothesis was made like in Conte and Doherty (2009), Mourikis et al. (2009), and Pham et al. (2012). In the flat-hypothesis version, the 3D map projection was not used but instead a homography transformed the feature points extracted in the orbital image geometry onto the expected descent camera focal plane (Hartley and Zisserman, 2003). This transformation assumes the observed scene is planar. Convergence only occurred in 16.6 percents of a 700-run Monte Carlo analysis whereas *Lion* could still maintain a 79.3-percent convergence rate, thus showing a significantly increased robustness to 3D terrains.

4.2.4 Discussion: the repeatability issue

As highlighted in Figure 4.3, both methods presented in this section have to face one major issue which is that the feature points extracted in the descent image are not necessarily also detected, or repeated, in the orbital image. There are some non-detections, but also some false alarms with descent points which had not been detected in the orbital image. The fact that geometric descriptors are relying on no other description than the spatial distribution of landmarks makes the accuracy and the reliability of the extraction step crucial in matching, and the repeatability issue affects them strongly. The problem appears even under similar illumination conditions and can only be due

to differences between the images in distance to the ground, viewpoint, sensor resolution or noise. In image-based software simulation, these differences led SC to fail to give enough correct matches even only for the filter to converge. GHT does a lot better because it is a voting process and it is designed to cope with lower feature point repeatability rate. In the 200-run Monte Carlo simulation, 93.5 % of the runs are converging and have a landing accuracy compatible with pinpoint landing. Nevertheless, 6.5 % of the GHT runs are still diverging, even after the outlier removal step. Paradoxically, the same voting process which is the greatest strength of the GHT method in low repeatability environments can also turn out to be its main weakness with respect to diverging runs. Indeed when an incorrect global image translation is voted for, not only one landmark measurement is wrong but all of them are. That causes the estimation error leaps visible in Figure 4.9 and leading to divergence.

Our educated guess to make this convergence rate 100 % was to improve the repeatability rate of landmarks by testing different extraction methods than the simple Harris operator. Structured features like craters are interesting because they are not only characterized by their terrain location but also by other geometric parameters like their radiuses for instance. This provides additional information to detect and match them. But craters are ruled out here as they cannot be reliably detected anywhere, even on the Moon. We nevertheless investigated if no useful additional information, equivalent to crater radius, could also be found for generic feature points like Harris or SIFT. A deeper look at their extraction process highlights a fundamental extraction parameter which appears relevant for landmark matching purposes: the image scale.

4.3 REPEATABILITY IMPROVEMENT THROUGH LANDMARK SCALE MANAGEMENT

The image scale of an object in the observed scene can be defined as its apparent size in pixels. For a given object size, it is determined by the distance between the camera and the object, the focal length of the lens, and the sensor resolution. Scale may thus be predicted from camera calibration and the estimation of the pose parameters done by the filter. Figure 4.11 shows a typical scaling issue with a crater being imaged by the same camera but at two different altitudes. It illustrates the fact that from an image point of view, a change in scale is equivalent to a change in pixel resolution. The same crater has a scale of 70 pixels in the higher image shot at 260 km of altitude, compared to 350 pixels at 50 km. The level of detail which can be perceived is totally different from one image to the other. The small-scale structures appearing in the 50-km image on the right cannot be seen any more in the higher image. This phenomenon may cause repeatability issues similar to those face by the GHT method in Section 4.2 if it is not properly handled by the image processing navigation software. During a lunar landing, the descent camera images the terrain at altitudes ranging from 100 km down to a few meters. This section investigates how to properly process image scale in order to improve the landmark repeatability all along the trajectory.

4.3.1 Scale-space model

Figure 4.11 illustrates the fact that changing the image resolution is equivalent to smoothing it. A smoothing can be modeled by a convolution operation $*$ of the reference image array I with a Gaussian kernel G as shown in Equation (4.1). Lindeberg formalized the set of images L obtained by smoothing I under the notion of scale-

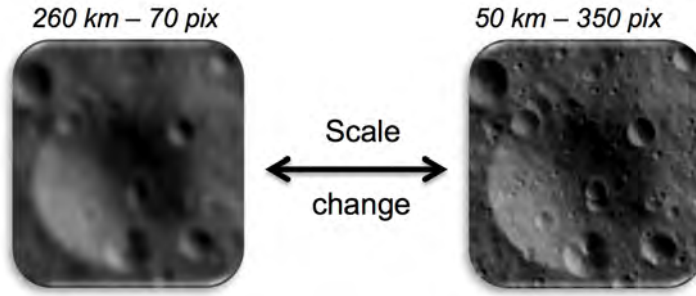


Figure 4.11 – Scale change illustration: the same crater is imaged by the same camera at 260 km of altitude (left), or 50 km (right).

space (Lindeberg, 1994).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (4.1)$$

x and y are the 2D image coordinates. The Gaussian kernel can be written

$$G(x, y, \sigma) = \frac{e^{-\frac{1}{2}(x^2+y^2)/\sigma^2}}{2\pi\sigma^2} . \quad (4.2)$$

The new image scale σ is defined as the standard deviation of the Gaussian function.

4.3.2 Scale-invariant extractor selection

Scale-invariant feature detectors select not only a location but also a characteristic scale for each image feature. By browsing the scale space for a pair of images of the same scene, they aim at matching common features between them, even if their apparent sizes in pixels are significantly different.

Lowe uses a Difference-of-Gaussian (DoG) detector to select SIFT features (Lowe, 2004). The DoG operator basically computes the difference between two smoothed images at nearby scales separated by a multiplicative constant k :

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) . \quad (4.3)$$

SIFT features are the local extrema of the D function within its 3-dimensional definition domain.

Mikolajczyk and Schmid (2004) proposed the Harris-Laplace detector, inspired from the classical Harris detector, to detect image corners at different scales. It is based on the scale-adapted second moment matrix μ . If $L_i(x, y, \sigma_D)$ is the image derivative in direction i after a Gaussian filtering at scale σ_D , then this matrix has the form shown on Equation (4.4). It represents the local gradient distribution after averaging the derivatives through another Gaussian kernel at scale¹ σ_I .

$$\mu(x, y, \sigma_I, \sigma_D) = \sigma_D^2 G(x, y, \sigma_I) * \begin{bmatrix} L_x(x, y, \sigma_D)^2 & L_x(x, y, \sigma_D)L_y(x, y, \sigma_D) \\ L_x(x, y, \sigma_D)L_y(x, y, \sigma_D) & L_y(x, y, \sigma_D)^2 \end{bmatrix} \quad (4.4)$$

A corner is defined as a strong intensity change in all spatial image directions. From Equation (4.4), the image point of coordinates (x, y) is thus said to be a corner at scale

¹ $\sigma_D = 0.7\sigma_I$ gives good results.

σ_I if μ has two large eigenvalues of the same order of magnitude. To avoid eigenvalue computation, the Harris detector actually selects the local maxima of the cornerness measure of Equation (4.5) in an image neighborhood sized according to scale.

$$C(x, y, \sigma_I, \sigma_D) = \det(\mu(x, y, \sigma_I, \sigma_D)) - \alpha \operatorname{tr}(\mu(x, y, \sigma_I, \sigma_D))^2 \quad (4.5)$$

α is usually experimentally set in the interval [0.04; 0.15]. Note that in the original Harris extractor, σ_I is simply chosen so as to smooth image noise (Harris and Stephens, 1988). The Harris-Laplace extractor computes Harris corners over a chosen range of scales but subsequently keeps only those corresponding to an extremum of the Laplacian-of-Gaussian (LoG) operator shown in Equation (4.6) over scale space. The identified maximum sets the characteristic image scale of the feature when several Harris corners are identified within the same image neighborhood at different scales.

$$| \operatorname{LoG}(x, y, \sigma_I) | = \sigma_I^2 | L_{xx}(x, y, \sigma_I) + L_{yy}(x, y, \sigma_I) | \quad (4.6)$$

Mikolajczyk and Schmid (2004) tested the repeatability rates of both DoG and Harris-Laplace extractors under scale change. The Harris-Laplace operator was shown to have better performance with a computation time of the same order of magnitude. Based on this result, we chose it as the baseline scale-invariant extractor for this thesis.

4.3.3 Scale reprojection formula

Image features can be associated to actual terrain features in the scene that the camera is observing. In the case of planetary landing, the scene features are called landmarks and they are selected from the orbital image. With scale-invariant extractors, a characteristic terrain scale σ can thus be defined for any landmark based on its characteristic scale on the orbital image σ_{orb} . This orbital scale can be reprojected onto the image plane of the descent camera to predict the scale of the landmark in the descent image σ_{reproj} , for instance using a priori pose estimates.

The geometry of this reprojection is illustrated in Figure 4.12. For camera i , d_i is the physical distance between the optical center C_i and the landmark, β_i is the field of view, s_i is the sensor resolution in pixel, α_i is the offset angle between the landmark direction and the optical axis.

By applying Thales' theorem for each camera within the plane containing the landmark and the optical axis, one can write

$$\frac{\sigma_{orb}}{\sigma} = \frac{f_{orb} / \cos(\alpha_{orb})}{d_{orb}} \quad (4.7)$$

$$\frac{\sigma_{reproj}}{\sigma} = \frac{f_{dsc} / \cos(\alpha_{dsc})}{d_{dsc}} \quad (4.8)$$

where

$$f_{orb} = \frac{s_{orb}}{2 \tan(\beta_{orb}/2)} \quad (4.9)$$

$$f_{dsc} = \frac{s_{dsc}}{2 \tan(\beta_{dsc}/2)} \quad (4.10)$$

are the focal lengths, each equal to the distance from the optical center to the image plane. From this, one gets

$$\sigma_{reproj} = \sigma_{orb} \frac{d_{orb} f_{dsc} \cos(\alpha_{orb})}{d_{dsc} f_{orb} \cos(\alpha_{dsc})} \quad (4.11)$$

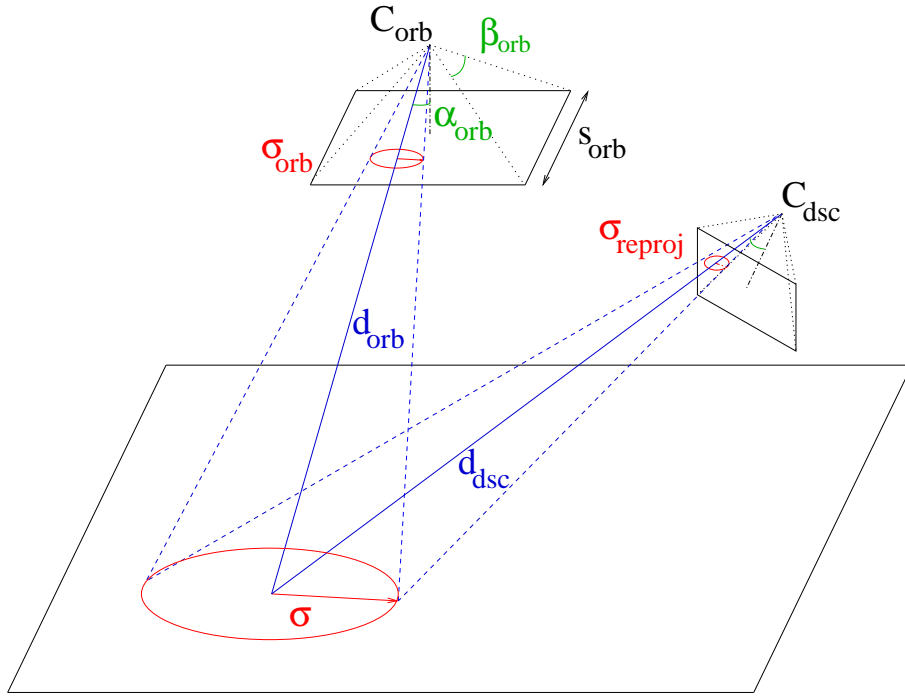


Figure 4.12 – Scale reprojection geometry

and eventually

$$\sigma_{reproj} = \sigma_{orb} \frac{d_{orb} \tan(\beta_{orb}/2) s_{dsc} \cos(\alpha_{orb})}{d_{dsc} \tan(\beta_{dsc}/2) s_{orb} \cos(\alpha_{dsc})}. \quad (4.12)$$

This reprojection model is based on a few assumptions which are discussed below. None of them is mandatory to derive a scale reprojection formula, they just make the formula simpler.

Isotropic orbital feature gradient

A corner is defined as locally strong intensity changes in all spatial image directions. We assume the intensity gradient is the same in every direction so that it can be simply modeled by a circle of radius σ_{orb} in the orbital image.

Flat level world at feature scale

The terrain is assumed to be flat and level within the back-projected landmark scale footprint. This flatness assumption is only local around the landmark and did not happen to be a problem during our tests over terrains with strong 3D topography. Should it become one in other tests, knowledge of the DEM could still be used to get rid of this assumption, but at the cost of extra complexity.

Orbital camera orientation

The orbital camera is assumed to be pointing nadir since most orbital images are taken this way. The landmark scale circle on the orbital image can then be projected onto a characteristic terrain circle of radius σ .

No affine distortion on descent camera at feature scale

Affine distortion is a simpler model of image perspective distortion often valid at feature level in computer vision. An affine transform turns the characteristic landmark terrain circle into an ellipse in the descent image. Equation (4.12) actually gives the semi-major axis length σ_{reproj} of this ellipse, which defines the reprojected descent scale of the landmark. The affine transform may alter the intensity gradient distribution around the landmark in such a way that a corner in the orbital image is not necessarily one in the descent image, even with scale correction. That is actually another source of repeatability issues. Affine-invariant extractors do exist but usually have worse robustness to scale-only change, which is our main concern here (Mikolajczyk and Schmid, 2004). Since a significant part of the descent image is seen with a low incidence angle within the large field of view employed for descent cameras, affine distortion is neglected and we assume a corner in the orbital image corresponds to one in the descent image. This is equivalent to assuming the characteristic landmark terrain circle is projected not into an ellipse but into a characteristic circle of radius σ_{reproj} in the descent image.

4.3.4 Comparison of repeatability rates

The repeatability rates between the descent and the orbital images are compared with and without using a scale-invariant technique for image feature extraction.

Test setup

Within the image-based software simulation environment of Section 4.1, two navigation maps were created. The first one was made by extracting regular Harris corners in the orbital image, and interpolating the back-projected rays with the DEM to get the 3D landmark coordinates. This is the same map as that used for the methods described in Section 4.2. The second map is made the same way but with Harris-Laplace features instead, and it includes their characteristic scales too. With the Harris map, descent images are processed with the same Harris extractor as for the orbital image. With the Harris-Laplace map, each descent image is processed with a multi-scale Harris extractor. It extracts Harris features over the whole reprojected scale range of landmarks computed from the scale reprojection formula of Equation (4.12). The scale range is discretized by a multiplicative constant $\sigma_{step} = 1.4$, such that $\sigma_n = \sigma_0 \sigma_{step}^n$.

Definitions

In each simulated descent image sequence, since the true camera trajectory is known, the exact image positions of mapped landmarks are computed to assess the repeatability of the descent image feature extractor. One landmark, be it Harris or Harris-Laplace, is said to be *repeated* if a descent image feature is found within 10 pixels. This 10-pixel value was empirically set for 1024×1024 -pixel descent images. One Harris-Laplace landmark is said to be *repeated with scale* if a descent image feature is found within 10 pixels and at a similar scale. A reprojected landmark scale σ_{reproj} is said to be similar to a descent feature scale σ_{dsc} if it satisfies Inequation (4.13) which characterizes the discretization domains in scale space.

$$Similarity \Leftrightarrow \frac{\max(\sigma_{reproj}, \sigma_{dsc})}{\min(\sigma_{reproj}, \sigma_{dsc})} \leq \sqrt{\sigma_{step}} \quad (4.13)$$

Lastly, the orbital repeatability rate is defined as the ratio of the number of repeated landmarks over the total number of landmarks within the field of view.

Results

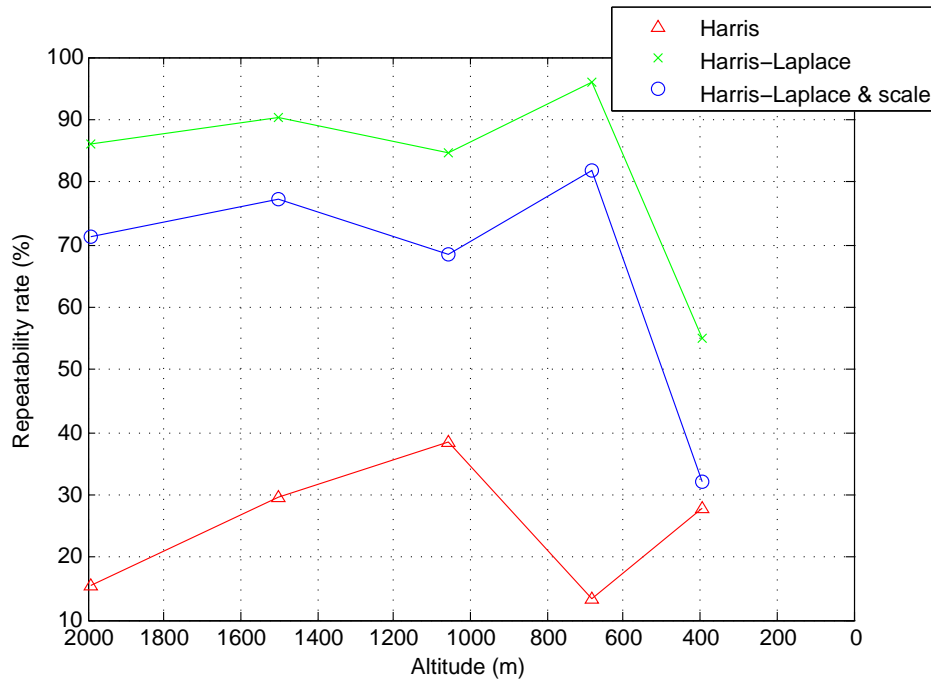


Figure 4.13 – Repeatability rate comparison between the single-scale feature extractor Harris and the multi-scale Harris-Laplace.

Figure 4.13 compares the orbital repeatability rates between the standard Harris and the Harris-Laplace approaches at various altitudes during the visual measurement phase, namely at altitudes where absolute image measurements were available in Subsection 4.2.3. The Harris-Laplace operator achieves a repeatability rate significantly better than Harris all along the descent. At 685 m of altitude for instance, repeatability is 95.8 % with Harris-Laplace, while it is only 13.5 % with Harris. Yet, the most interesting result in this test is that more than half of the repeated landmarks are so at the scale computed by reprojecting that from the orbital image with Equation (4.12), and always at a better rate than regular Harris. We will now try to integrate these enhanced repeatability performances in *Lion* by using the a priori state estimate from the filter to predict the descent image scales of landmarks and use them as an additional description for matching. The next section introduces a new landmark matching method which was designed to take this scale information into account.

4.4 LANDMARK RECOGNITION USING SCALE-AUGMENTED IMAGE REPROJECTION

The new matching algorithm uses the a priori EKF state and covariance estimates to predict the scale and the research area for each landmark in the descent image. It then runs a selection process to retains only landmarks which are considered non-ambiguous for matching.

4.4.1 Offline map

The map building process is identical to that described in Subsection 4.2.1 except landmarks are extracted using the Harris-Laplace method in the orbital image instead of the standard Harris operator. The map is eventually a $N \times 5$ matrix which contains for each landmark its 3D position in $\{g\}$, its characteristic scale σ_{orb} in the orbital image, and its Harris cornerness score at that scale defined in Equation (4.5).

4.4.2 Online image processing

The online part of the algorithm is divided in four main steps which are detailed in the following:

1. Landmark prediction,
2. Landmark selection,
3. Descent image measurement,
4. RANSAC-based outlier removal.

Landmark prediction

The position of a landmark in the descent image, its image scale, and search area, are computed from the EKF estimates of the state of the lander.

The image position z_j of landmark j is predicted using the associated projection function h_j , an a priori estimate of the state \hat{x} , and its global coordinates $p_{gl_j}^g$ like in Equation (3.17).

The predicted descent scale σ_{reproj} is computed from Equation (4.12) using the known orbital image parameters and the descent state predictions propagated by the navigation filter.

Finally at first order, the descent image position covariance matrix P_j^I of the landmark is derived from the state covariance matrix P propagated in the filter by

$$P_j^I = H_j P H_j^T + R, \quad (4.14)$$

where H_j is the Jacobian matrix of the landmark projection function from Equation (3.21) and R is the covariance of the image extraction noise. Equation (4.14) allows to compute a 3σ elliptical area in the image within which the landmark is located with a Gaussian probability of more than 97 % (Gura and Gersten, 1970). Equation (4.14) and the resulting Gaussian distribution of the image position of the landmark are based on a linearization of the projection function.

Landmark selection

Not all the landmarks visible in the descent image are considered suitable for matching. If the elliptical research area of a landmark is not overlapped by another, then it is directly selected for matching. However when several ellipses cross each other, only that corresponding to the landmark with the higher Harris characteristic score is selected. The characteristic score of a Harris-Laplace feature is defined as the cornerness score at its characteristic scale. This selection process is illustrated in Figure 4.14.

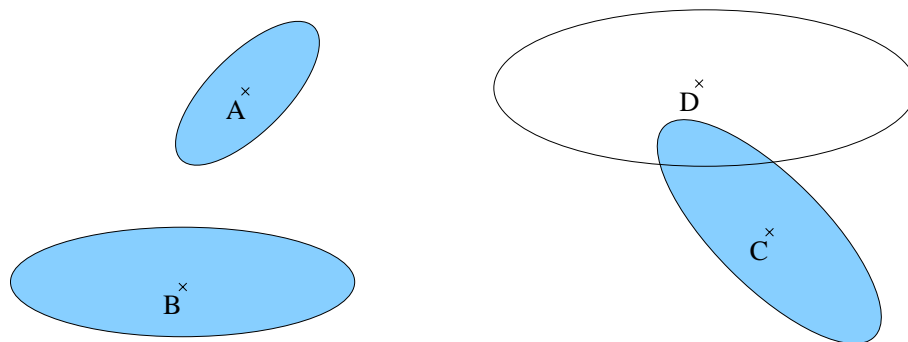


Figure 4.14 – Landmark selection process. Research ellipses of landmarks A and B are not overlapped, they are both selected. Landmark C and D overlap each other, only C is selected since it has the highest Harris characteristic score.

Descent image measurement

For each landmark selected, the Harris corneriness criterion of Equation (4.5) is calculated within the research ellipse with $\sigma_I = \sigma_{reproj}$ the predicted reprojection scale. The position of the pixel with the maximum value is chosen as the 2D image match for that landmark.

Outlier removal

The set of image-to-map landmark matches is verified to eliminate outliers, i.e. mismatches, using the same RANSAC implementation as that described in Subsection 4.2.2. The associated inliers form the final set of matches which is fed directly as a measurement to the tight *Lion* navigation filter of Chapter 3.

4.4.3 Navigation performance

A Monte Carlo analysis over 100 runs has been performed to evaluate the performance of the whole navigation system using the absolute landmark matching method presented in this section. We tested the most challenging DEM with the terrain relief variation of 500 m. Like previous simulations, the initial navigation errors were according to independent zero-mean Gaussian distribution on each axis, with 3σ values equal to 1 degree in attitude, $10 \text{ m}\cdot\text{s}^{-1}$ in velocity and 100 m in position at the 2-km start altitude. The Harris-Laplace extraction in the orbital images browsed through 25 different scales between 1 and 80 pixels with a multiplicative factor of 1.2 between each. Only Harris corners with scores above 0.2 times that of the image maximum were selected at each scale.

Figure 4.15, Figure 4.16, Figure 4.17 show the results of the estimation error and the largest 3σ bound from the filter over each axis respectively for attitude, velocity and position. The most significant result is that all the runs are now converging and the 7 % divergent cases previously found with the GHT method are now eliminated.

Figure 4.18 shows the evolution of the number of matches fed to the filter with respect to time. The switch between the visual measurement phase and the inertial drift occurs at $t = t_V = 4040 \text{ s}$ with the current map extraction density settings, which corresponds to an altitude of 80 m. The statistics of all the Monte Carlo runs are provided in Table 4.3.

The 3-RMS dispersion of the position error at touchdown is 53.8 m which largely meets the 100-m accuracy requirement for pinpoint landing. This value is even smaller

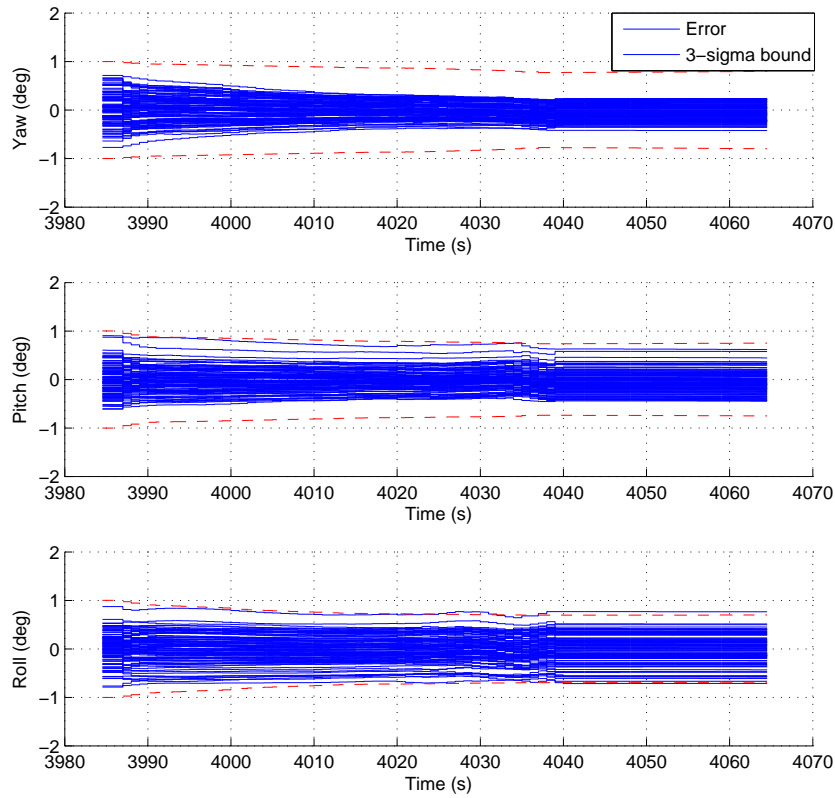


Figure 4.15 – Lion attitude estimation error during 100 Monte Carlo runs in image-based simulations. The red dotted line shows the 3σ bounds propagated by the EKF for the run which had the largest final standard deviation.

Table 4.3 – Dispersion of Lion navigation error at the end of the visual phase (V) and at touchdown (TD) for all runs on image-based simulations. The dispersion over each axis is measured with the 3σ value for scalar components, where σ is the standard deviation. The total dispersion of the vector quantities is measured with the 3 -RMS value.

Time		t_V	t_{TD}
$\delta\theta_{g,x}^b$	(roll, deg)	0.9	0.9
$\delta\theta_{g,y}^b$	(pitch, deg)	0.7	0.4
$\delta\theta_{g,z}^b$	(yaw, deg)	0.5	0.5
$\delta\theta_g^b$	(total, deg)	1.2	1.1
$\delta v_{gb,x}^g$	(downrange, m.s ⁻¹)	0.7	1.2
$\delta v_{gb,y}^g$	(crossrange, m.s ⁻¹)	0.8	1.2
$\delta v_{gb,z}^g$	(height, m.s ⁻¹)	0.9	0.9
δv_{gb}^g	(total, m.s ⁻¹)	1.4	1.9
$\delta p_{gb,x}^g$	(downrange, m)	12.9	35.3
$\delta p_{gb,y}^g$	(crossrange, m)	11.8	32.9
$\delta p_{gb,z}^g$	(height, m)	10.1	23.9
δp_{gb}^g	(total, m)	20.2	53.8

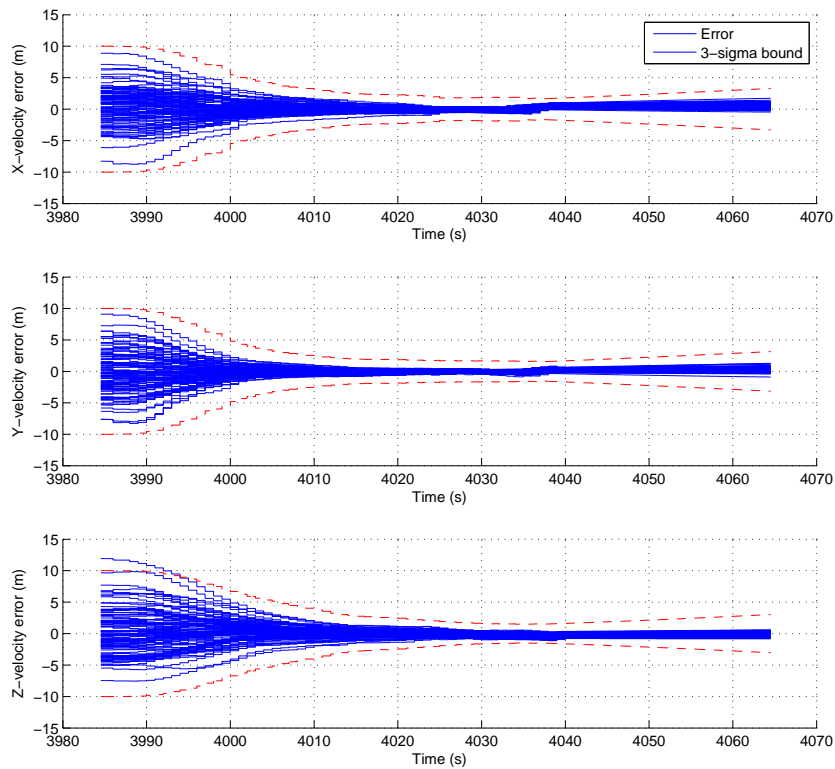


Figure 4.16 – Lion velocity estimation error during 100 Monte Carlo runs in image-based simulations. The red dotted line shows the 3σ bounds propagated by the EKF for the run which had the largest final standard deviation.

at the moment of the last landmark matching since it falls down to 20.2 m before inertial drift, more than four times more accurate than GHT in Table 4.1. Here again, the mean errors are negligible. Based on these promising results, we can proceed to further testing of *Lion* for pinpoint landing.

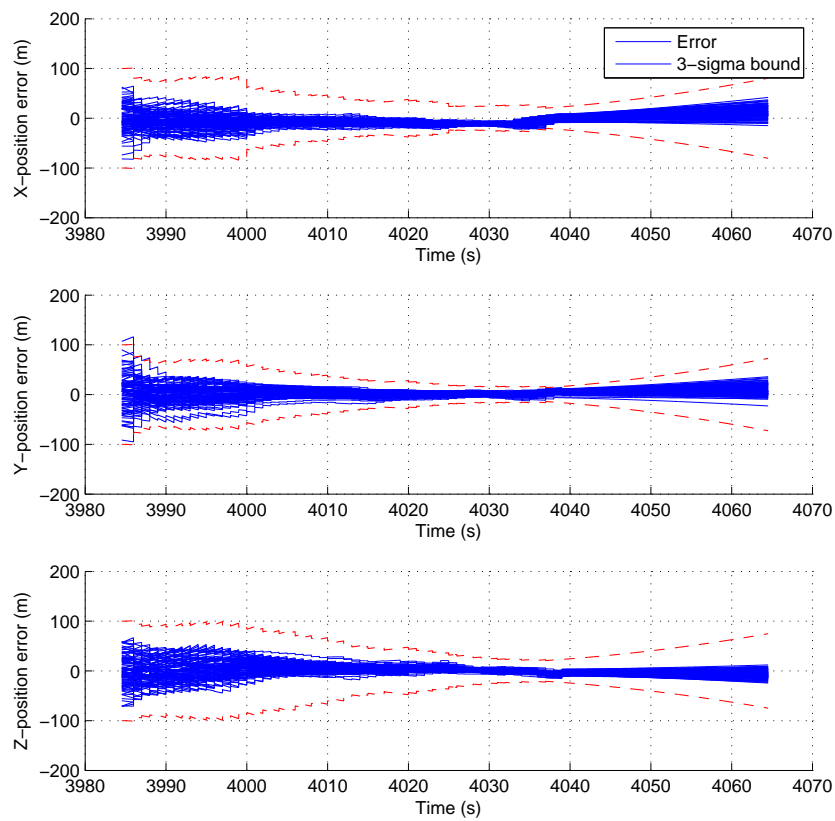


Figure 4.17 – Lion position estimation error during 100 Monte Carlo runs in image-based simulations. The red dotted line shows the 3σ bounds propagated by the EKF for the run which had the largest final standard deviation.

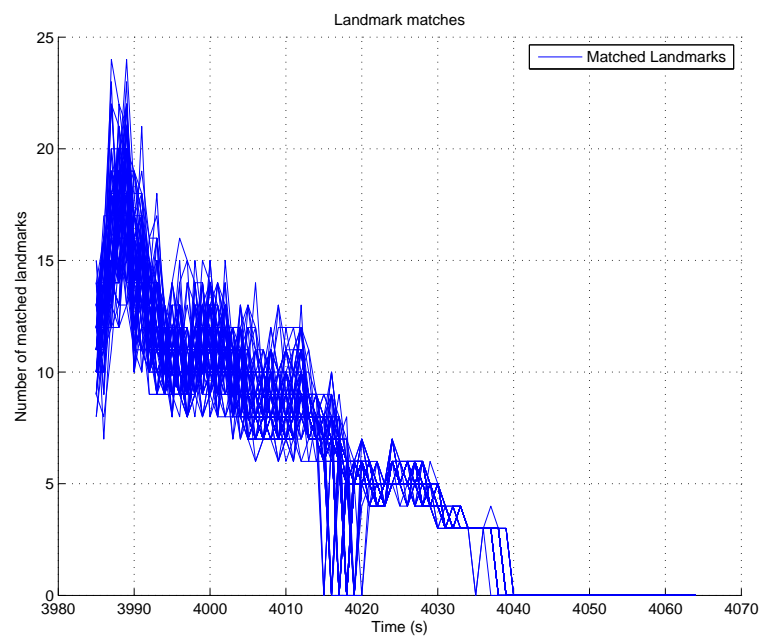


Figure 4.18 – Evolution of the number of absolute measurements during 100 Monte Carlo runs in image-based simulations.

CHAPTER CONCLUSION

In this chapter, three geometric image-to-map matching techniques for generic landmarks have been proposed, implemented along with the tight navigation filter, and evaluated in image-based software simulations representative of the approach phase of a lunar landing. All of them are compatible with any terrain topography, whether it is flat or not, through an efficient use of the camera projection function and a priori information available from a Bayesian filter instead of a planar rectification. Landmark repeatability issues met by most geometric methods, and troublesome on landing trajectories due to significant altitude change, led to completely rethink the way image scale is used in the algorithm. While other algorithms in the literature only proceed to global image scale corrections if ever, we show scale can be efficiently used as an individual descriptor for each landmark in a new method which was selected as the baseline for *Lion*. Based on Harris-Laplace orbital landmarks, it raises the repeatability rates significantly and improves the accuracy down to 20.2 m at the end of the visual phase and 53.8 m at touchdown (3 RMS), with a 20-deg illumination change. These performances are largely below the 100-m accuracy requirement for pinpoint landing and were obtained with significant 3D terrain relief, namely surface height variations of 500 m on a lunar landing approach phase trajectory starting at 2 km.

The next chapter will focus on the design of a lunar-analogue indoor optical test bench, in order to prove that *Lion* can estimate the pose of real camera in a physical environment representative of the Moon.

5

HARDWARE TEST BENCH DESIGN & ANALYSIS

CONTENTS

5.1	LUNAR-REPRESENTATIVE REQUIREMENTS ANALYSIS	73
5.2	TEST BENCH DESIGN	75
5.2.1	Camera and lens	75
5.2.2	Planetary surface model	77
5.2.3	Mock-up platform	80
5.2.4	Camera platform	82
5.2.5	Illumination system	84
5.3	TEST BENCH OPERATIONS	86
5.3.1	Image acquisition	87
5.3.2	True pose computation	88
5.4	ABSOLUTE NAVIGATION ERROR OBSERVABILITY	89
5.4.1	Manual ground truth accuracy	89
5.4.2	Observable absolute navigation error model	93
5.5	VISUAL GROUND TRUTH REFINEMENT	95
5.5.1	Method	95
5.5.2	Performance	95

VISILAB is a hardware test bench for vision-based navigation algorithms in conditions representative of lunar descent and landing. It has been designed, built, and integrated indoor in a lab at ESA-ESTEC. After software simulation in Chapter 4, hardware experimentation aims at demonstrating the performance of *Lion* on real images of a physical environment. *Visilab* includes a lunar-analogue planetary mock-up, a camera mounted on a robotic arm, a sun-representative illumination system, and a set of methods to determine the ground truth pose of the camera with respect to the reference mock-up frame.

Section 5.1 derives the list of hardware requirements which was used to ensure *Visilab* is mission-representative. Based on these requirements, Section 5.2 presents the full design process of the lunar mock-up, its support platform in the lab, the camera support platform, and the selection of the camera and illumination hardware. The operations of *Visilab* are described in Section 5.3, in terms of how to acquire images at a position and orientation specified by their coordinates with respect to the reference mock-up frame. Section 5.4 computes the camera pose uncertainties, or ground truth accuracy, in the operation of the bench and proposes a new model to evaluate the observability of the absolute navigation error in *Visilab*. A visual refinement process for ground truth is tested in Section 5.5 to improve the observability of the absolute navigation error below the 100-m pinpoint landing requirement at scale.

5.1 LUNAR-REPRESENTATIVE REQUIREMENTS ANALYSIS

Table 5.1 shows a list of preliminary hardware requirements which ensures that *Visilab* provides mission-representative lunar flight images and inertial data to test *Lion* on. Hardware aspects to be accounted for includes a camera, an IMU, a planetary surface model, motion capability of the camera support, and an illumination system.

Table 5.1 – *Visilab* hardware requirements

Hardware	Requirements
Camera	1024 × 1024-pixel sensor 70-deg field of view
IMU	Software-simulated
Planetary surface model	Lunar DEM from NASA LRO mission Real data only
Motion capability	4 degrees of freedom: 3 translations, pitch rotation
Illumination	White light Parallel rays Uniform flux

The 1024 × 1024 camera sensor resolution and the 70-degree Field of View (FoV) were chosen since they are representative of current state of the art in European hardware discussed in Subsection B.2.3.

No flight-representative inertial data can be obtained by fixing an IMU to the camera support in *Visilab* because the signal-to-noise ratio cannot be scaled up correctly to a lunar mission. Indeed, specific forces and angular rates sensed in the lab would be

much smaller than in the actual flight while the noise and bias would be of the same order of magnitude. Actually, noise and bias might even be larger than in a real space flight because a much lower-class IMU would be used in the lab. The inertial data thus had to be generated by software. Noise and bias representative of aerospace-class IMU performances discussed in Subsection B.2.1 were added to the ground truth trajectory.

Planetary surface mock-ups can be manufactured from a DEM at input. A machine mills the DEM profile line by line to get the final 3D terrain out of a block of reinforced resin. In *Visilab*, the DEM was selected from the altimetry dataset of the NASA LRO spacecraft. It was decided to only use DEMs with no or little need for post-processing so as to obtain a terrain model as representative of the true Moon as possible.

The lunar descent trajectory shown in Section 1.3 is almost in a vertical plane with respect to a surface reference frame. Two degrees of freedom in translation are thus needed for the camera in *Visilab* to simulate lunar descent: altitude and downrange. A third translation axis, equivalent to crossrange, is included in the requirement so the camera can cover the whole plan parallel to the mock-up for the acquisition of orbital images. Attitude mostly varies about the pitch axis during the descent so *Visilab* requires only one degree of freedom in rotation. In total, the requirement thus asks for 4 degrees of freedom: 3 translations, and 1 rotation about the pitch axis. A camera platform had to be designed to provide this motion capability. This design could rely on the initial equipment available in the lab and shown in Figure 5.1, namely a motorized translation table with a 92-cm course. The translation table was designed within ESA and can be controlled to a 0.1-mm position accuracy along the axis. It is mounted on an optical table which is 2.5 m long and 1.2 m wide (Melles Griot, 2012).

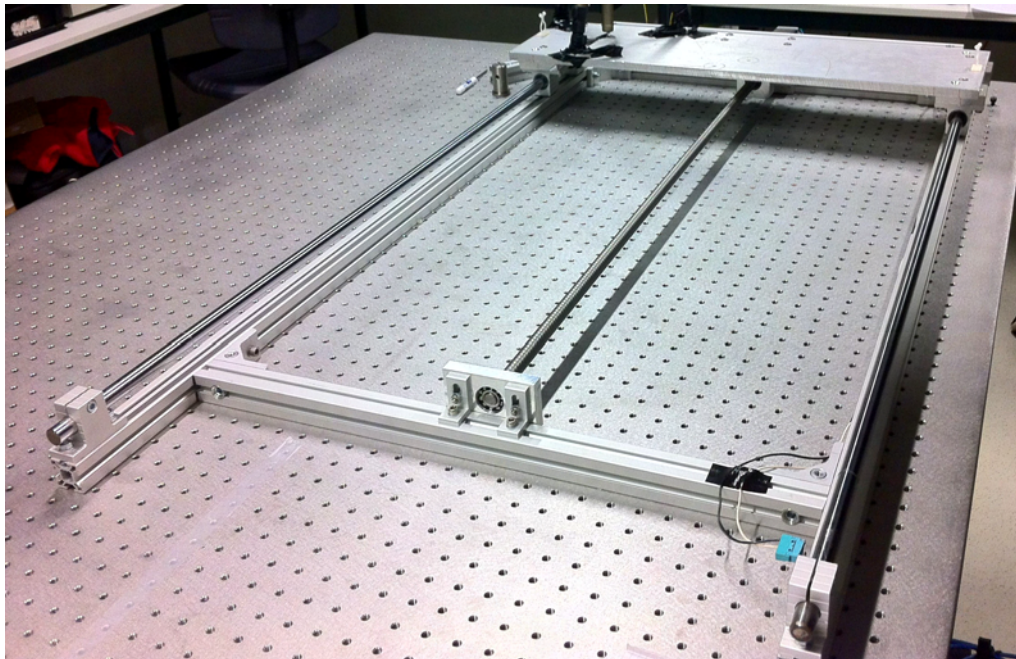


Figure 5.1 – Initial equipment available in the lab

The illumination system for *Visilab* should be as representative of the Sun as possible. This converts into requirements for white light, parallel rays, and a uniform light flux density over the mock-up surface.

5.2 TEST BENCH DESIGN

After analyzing the requirements, this section describes how *Visilab* was designed to meet them. Except for the planetary model which was tailored to our request by the German Aerospace Center (DLR), all the equipment is commercial off the shelf.

5.2.1 Camera and lens

The camera selected for *Visilab* has a 1280×1024 monochromatic CMOS sensor, which is enough for the 1024×1024 image resolution requirement. It offers a global-shutter mode, an 8-bit resolution depth and it works with standard C-mount lenses (Imaging Development Systems, 2012).

The dimensions $l \times w$ of the sensor specified on the camera data sheet are $l = 6.784$ mm and $w = 5.427$ mm. The width, smaller, is the driving parameter to obtain the FoV $\alpha = 70$ deg required for *Visilab* images. Figure 5.2 shows the pinhole camera model with the optical center C . The optical distortions are not considered since they will be estimated during camera calibration and corrected by software. The principal point P is the intersection of the camera optical axis Z with the sensor plane. With short focal lengths used to obtain a large FoV, the lens is focused at infinity during the experiment. Thus the focal plane of the lens will be located on the sensor plane, and the distance f between C and P is the focal length. The choice of the focal length f of the lens thus drives the field of view according to

$$f = \frac{w/2}{\tan(\alpha/2)}. \quad (5.1)$$

To observe a field of view $\alpha \geq 70$ deg, f then must be such that $f \leq 3.875$ mm.

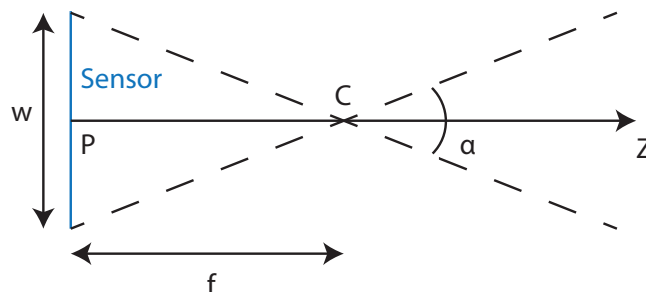


Figure 5.2 – Optical lens sizing

A 3.5-mm lens by Goyo Optical Inc. (2011) was selected to offer a FoV $\alpha_c = 75.6$ deg which is slightly above the 70-degree requirement for *Visilab* and thus considered compliant. Table 5.2 sums up the main characteristics of the {camera+lens} set which is shown in Figure 5.3.

An additional 4.5-mm lens by Goyo Optical Inc. (2013) was also used for data acquisition on some tests. It only gives a FoV of 62.2 deg but this is tolerated as on the other hand optical distortions are also visibly lower than on the 3.5-mm lens and we were initially concerned this might be an issue for *Lion*. The camera and lenses in *Visilab* have been preferred to other equivalent models meeting the requirements since they were cheap and available for preliminary testing at ONERA.

Table 5.2 – Main parameters of the camera mounted with the 3.5-mm C-mount lens. The minimum distance of focus was measured manually.

Resolution	1280 x 1024
Image acquisition	Global shutter
Resolution depth	8 bits
Sensor size	6.784 x 5.427 mm ²
Field of view	78.4 x 75.6 deg ²
Minimum distance of focus	6 cm



Figure 5.3 – Visilab camera and lens mounted on a tripod head

5.2.2 Planetary surface model

Dimensions and milling resolution

Within the budget allowed for *Visilab*, several manufacturing options were available for the mock-up and are shown in Table 5.3.

Table 5.3 – *Mock-up manufacturing options*

Option	Size (m × m)	Milling strip resolution (mm)	Maximum depth (mm)
1	1 × 1	0.5	100
2	2 × 1	0.5	50
3	2 × 2	1	50

Option 1 offers the largest depth range of 100 mm. However, the lunar DEMs we scaled down for the mock-up always led to height ranges below 5 cm at scale. Thus the depth gain of option 1 was not necessary and it has been discarded right away.

As discussed in Section 1.3, the lunar landing trajectory is mostly planar. Thus the longer the mock-up, the longer part of the trajectory can be simulated. Both options 2 and 3 offer a maximum length of 2 m, but option 2 has a finer horizontal milling resolution and was eventually selected. Machining was achieved with a 6-mm milling tool. The 2 × 1 m² mock-up was manufactured as two independent tiles of 1 m² each.

DEM selection

Source The NASA LRO mission and its laser altimeter *LOLA* have been providing measurements which NASA used to create DEMs of the Moon with vertical accuracy of about 1 m due to the combination of uncertainties in laser measurements and spacecraft navigation (Mazarico et al., 2011). All data could be downloaded from the *LOLA* archive website <http://imbrium.mit.edu/LOLA.html> at the time of writing. We focused on those DEMs centered on the lunar south pole. They come up as height matrices using a polar-stereographic projection with horizontal resolution ranging from 5 to 400 meters per pixel (Smith et al., 2010).

Scaling issues Several approaches have been considered for scaling the DEM of the mock-up in *Visilab*. First, terrain data were scaled by matching the minimum altitude during the last part of the braking phase going from 14 km down to 2 km above the surface with the minimum distance of focus of 6 cm in *Visilab* from Table 5.2. Unfortunately at scale, the 0.5-mm mock-up horizontal milling strip resolution required to use the 20-m resolution DEM provided by NASA and on which many artefacts are visible. These artefacts are trenches, peaks and holes due to false measurements of the laser altimeter and which are thus not representative of the true terrain. Image processing techniques such as Gaussian, median and mean filtering were applied on the DEM image to try and smooth out the artefacts. Unfortunately though, the level of filtering required for the artefacts not to be visible any more also caused a huge loss of real terrain details. To make sure the mock-up remained realistic, we thus decided not to use any such post-processing filtering techniques but only raw NASA data. Not seeing the artefacts thus became the sizing driver for the DEM resolution. The 400-m DEM resolution is the one at which artefacts are least visible. It is shown in Figure 5.4 and was used as a basis for the final mock-up DEM to be prepared from.

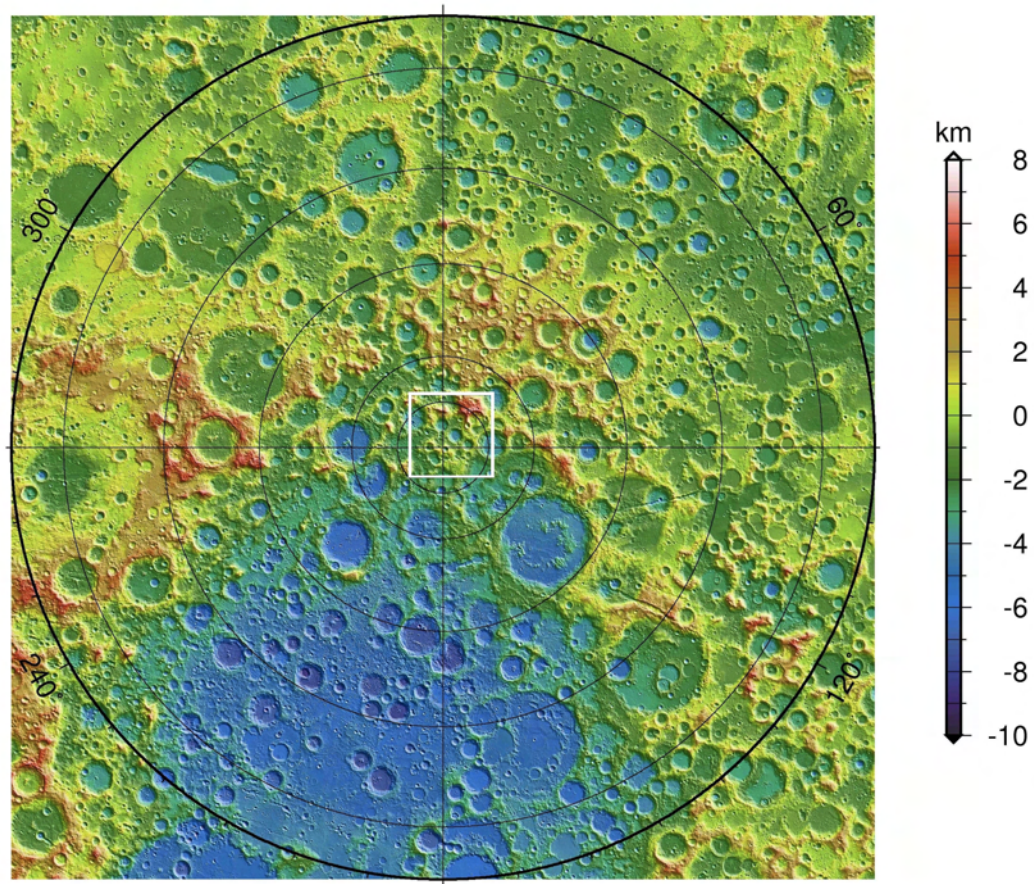


Figure 5.4 – NASA DEM of the lunar south pole selected as a basis to prepare the final mock-up DEM in Visilab. The white rectangle show the area covers by Figure 5.5.

Selection of the lunar area Candidate sites for the ESA lunar lander mission are all in the vicinity of the south pole. They are shown in Figure 5.5, which covers the area within the white rectangle at the center of Figure 5.4. It is interesting for *Visilab* to have topography representative of planetary exploration targets by including these landing sites in the mock-up DEM. As a consequence, a requirement is to pin the south pole down in the middle of one of the two 1 m² tiles forming the mock-up, 0.5 m away from the closest edges. The problem was then to orient and scale the rectangle of the mock-up DEM on the basis NASA DEM.

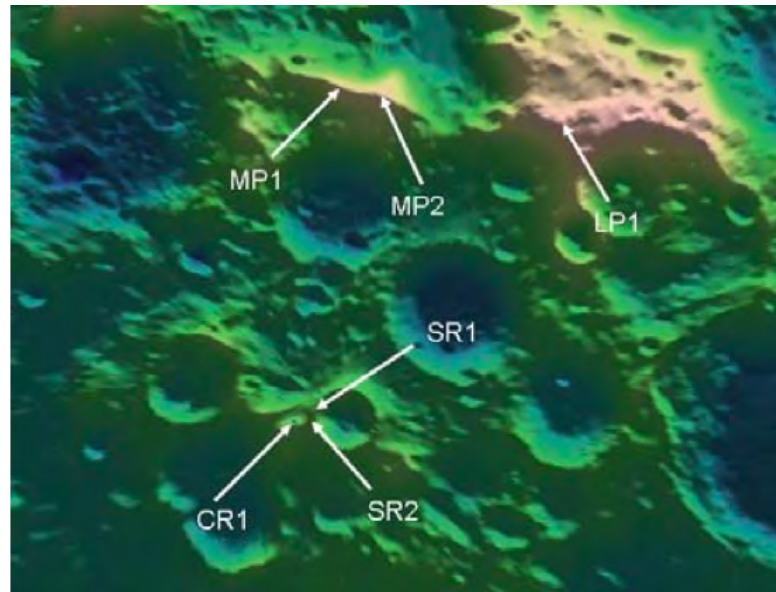


Figure 5.5 – Candidate landing sites near the lunar south pole for the ESA lunar lander mission (Vanoutryve et al., 2010).

A lunar approach will most likely be made above the near-side of the Moon to enjoy direct-to-Earth communication. Near-side corresponds to the longitude interval [270 deg → 0 deg → 90 deg], i.e. the top half of Figure 5.4. Orientation was thus constrained to have the mock-up DEM rectangle within this part of the image. Eventually, we decided to constrain the orientation of the main axis of the mock-up along the 0-deg longitude, since it appears to have the highest crater density and the most various craters in diameter. That could indeed be of interest if *Visilab* is to be used for crater detection algorithms in the future.

With the orientation constrained and the position of the South Pole pinned down on the mock-up, only the scaling problem remains. It should be clear that due to the length of a complete lunar landing trajectory from the de-orbit burn to touchdown, fitting it within the 2-m-long mock-up would result in an extremely low altitude at scale and the limited 0.5-mm horizontal milling resolution would be apparent and the images not realistic. Thus in *Visilab*, the images do not show exactly what a true lunar lander camera would see but only something representative of it. Scaling was first tackled by sizing the data so that the maximum elevation difference fits within the 5-cm depth of the mock-up. This approach was called *vertical sizing*. Unfortunately here again artefacts were visible. An example of artefacts rendered is shown in Figure 5.6, they appear as trenches mainly at the bottom left. The image was rendered with the *PANGU* planetary image generator.

Since the data artefacts appeared to threaten so much the realism of the mock-up,

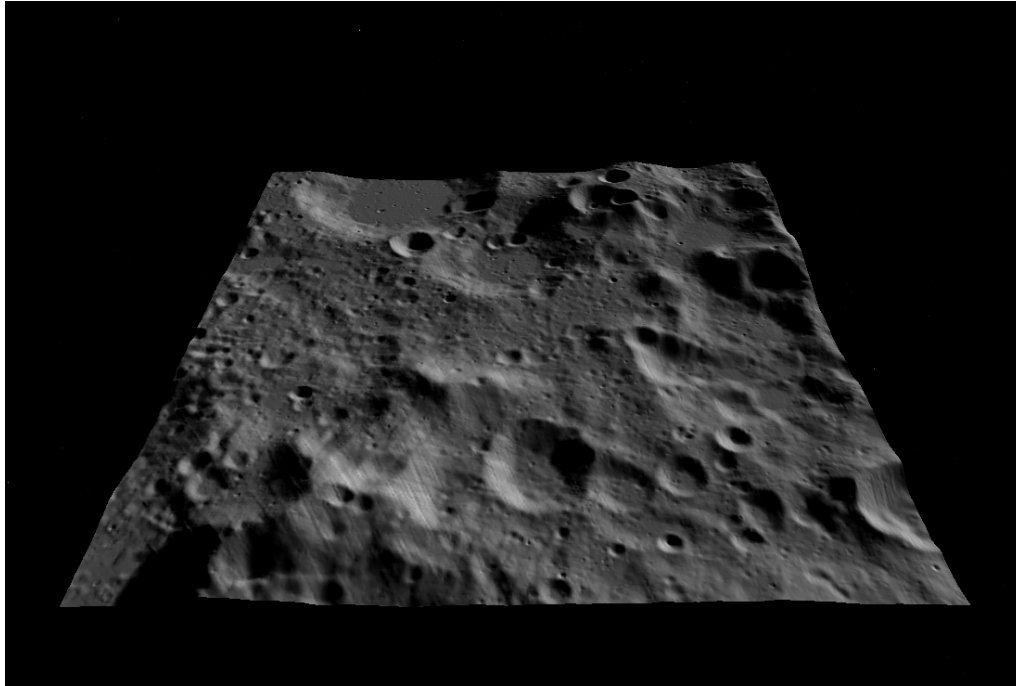


Figure 5.6 – DEM artefacts are visible at the bottom left with the vertical sizing approach

it was eventually decided it should have the largest possible terrain footprint on the original NASA DEM so as to make those artefacts least visible. The final cropped DEM selection goes from below the south pole to the top of the basis DEM in Figure 5.4. This sizing limits the artefact enhancement effect associated to the vertical sizing approach where DEM data were being oversampled. However with this new scaling, the maximum elevation difference of the mock-up was of only 1.5 cm. We decided to distort the mock-up in vertical dimension to make it fit the 5-cm height range. Although this makes the terrain different from the true Moon, it prevents the detail smoothing effect that would happen if the height range was too small because of the discrete altitude step of the milling tool. As shown in Figure 5.7, the terrain keeps looking lunar-like with this distortion. This figure is rendered in *PANGU* only using the input DEM for *Visilab* manufacturing and does not take into account milling tool effects. It was used to verify that when putting the camera at a 6-cm height above it, no major artefacts was observed. 6 cm is the minimum focus distance of the camera mentioned in Table 5.2.

Mock-up summary

Table 5.4 recaps the characteristics of the *Visilab* mock-up and of its DEM. It also shows the dimensions of the associated true terrain on the Moon.

Figure 5.8 shows the manufactured planetary mock-up in its container box when it was delivered in ESA-ESTEC premises.

5.2.3 Mock-up platform

The most practical configuration to accommodate the planetary surface model in *Visilab* was found to have it stand vertically with the camera mounted on a robotic arm and looking at it horizontally from the side.

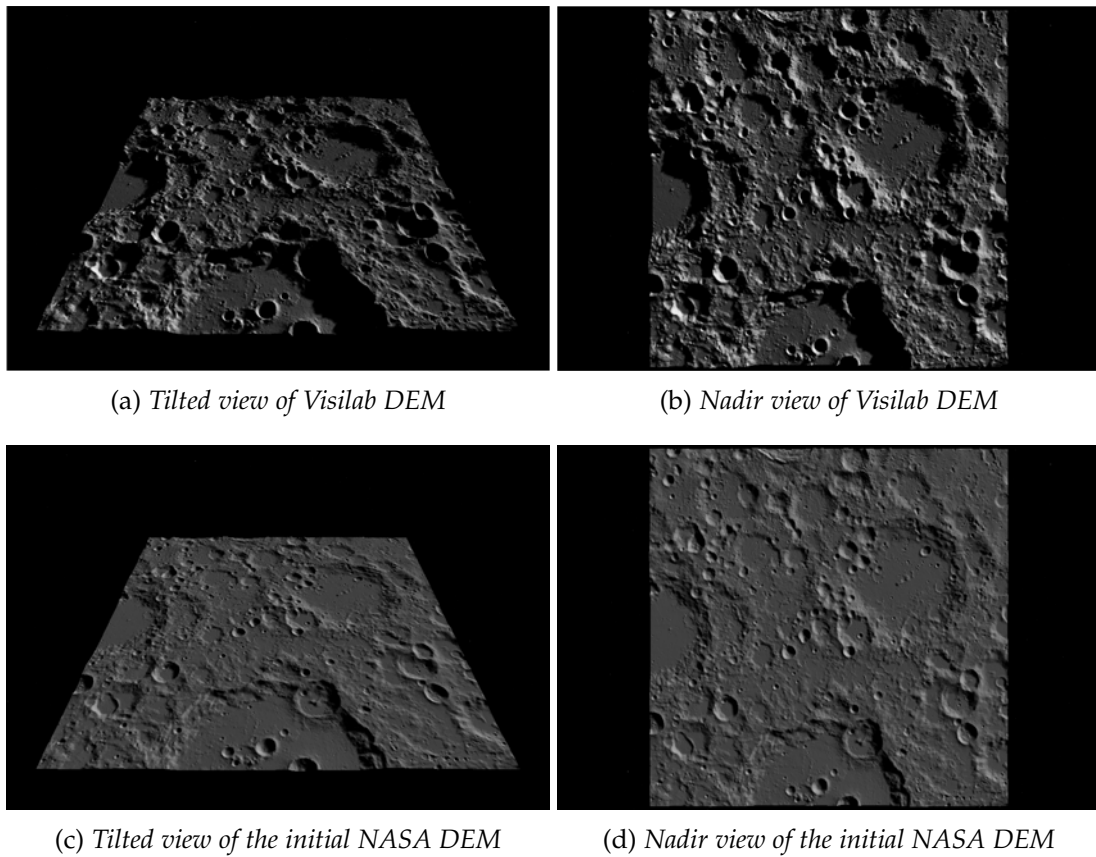


Figure 5.7 – PANGU-rendered views of the Visilab DEM versus the initial DEM of the same area in similar illumination conditions. Visilab DEM is enlarged about three times vertically to make it fill the 5-cm elevation range available for manufacturing the mock-up.

Table 5.4 – Characteristics of Visilab mock-up and DEM

Exact mock-up dimensions	980 × 1960 mm ²
Milling horizontal line step	0.5 mm
Height range	50 mm
DEM resolution	1960 × 3920
DEM resolution depth	16 bits
Lunar dimensions	960 × 1920 km ²
Lunar DEM pixel footprint	490 m

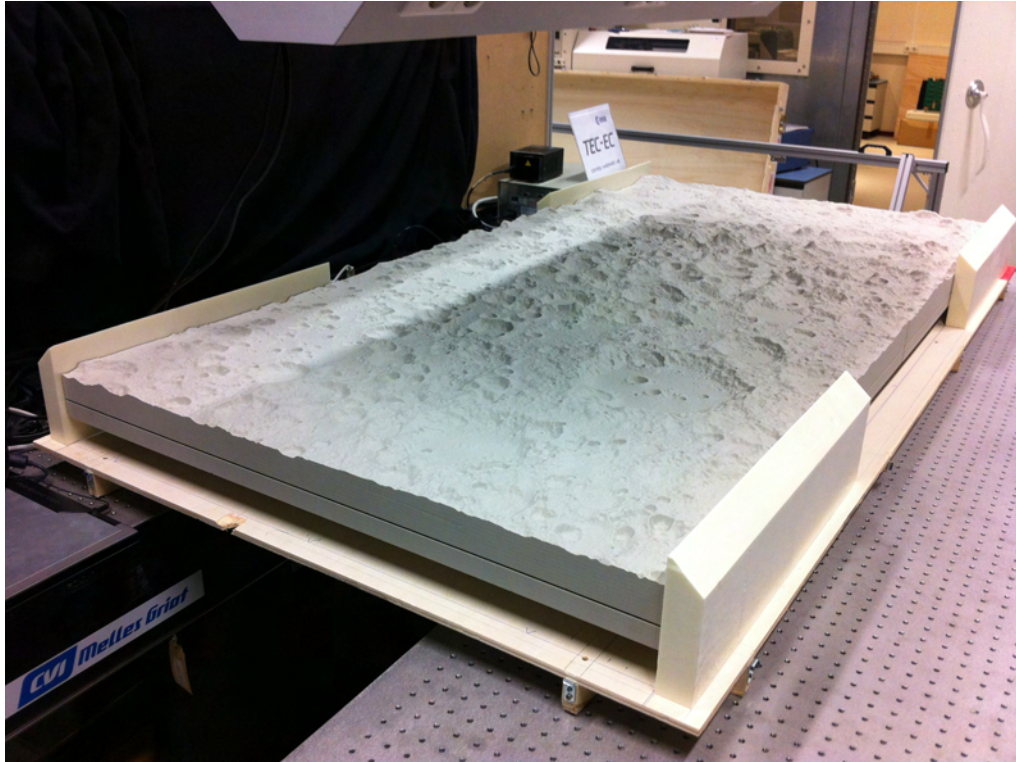


Figure 5.8 – Planetary surface mock-up being delivered in *Visilab*

A support platform thus had to be constructed to host the mock-up in the lab. The final design of this platform, its configuration in the lab and how the mock-up is assembled with it is shown in Figure 5.9. The support was built with a commercial aluminum extrusion system (KANYA, 2012). There is a 25-cm long extension of the platform on the left of the mock-up in order to accommodate an aluminum plate where to pin a camera calibration pattern to compute the camera pose later. The support platform is integrated between two optical tables in *Visilab*. The whole working area of the table in front of the vertical mock-up, i.e. $2.40 \times 1.15 \text{ m}^2$, will be used by the robotic arm supporting the camera so it must be left free. Horizontal beams are attached to the back panel of the mock-up to enforce the coplanarity constraint with the back of the calibration pattern plate. Three slanted beams prevent the assembly from falling forward or backward.

5.2.4 Camera platform

The motorized linear translation table initially present in *Visilab* and shown in Figure 5.1 was transformed into a simple robotic arm to support the camera. It was built with the same aluminum extrusion system as the mock-up platform and it is presented in Figure 5.10. This new camera robotic platform offers the 3 degrees of freedom required in translation, along with full rotation about the pitch axis for the camera thanks to the use of the tripod head of Figure 5.3 which is attached at the edge of the arm. Only the downrange translation axis is motorized. The positions on the altitude and crossrange axes are set manually using a ruler taped directly along the axis. The altitude axis is equipped with an endless screw. It can be upgraded by attaching it to an extra stepper motor to get both the downrange and altitude axes automatized. There is no direct reading for the pitch angle. It needs to be set approximately by hand



(a) Support platform

(b) Platform integrated on the optical tables



(c) Mock-up assembly (back view)

(d) Mock-up assembly (front view)

Figure 5.9 – Support platform and mock-up assembly in Visilab

first and then checked through optical pose estimation using the calibration pattern at the left of the mock-up. Limited motion is possible about the roll and yaw axes for the camera. Course ranges for this robotic platform are presented in Table 5.5.

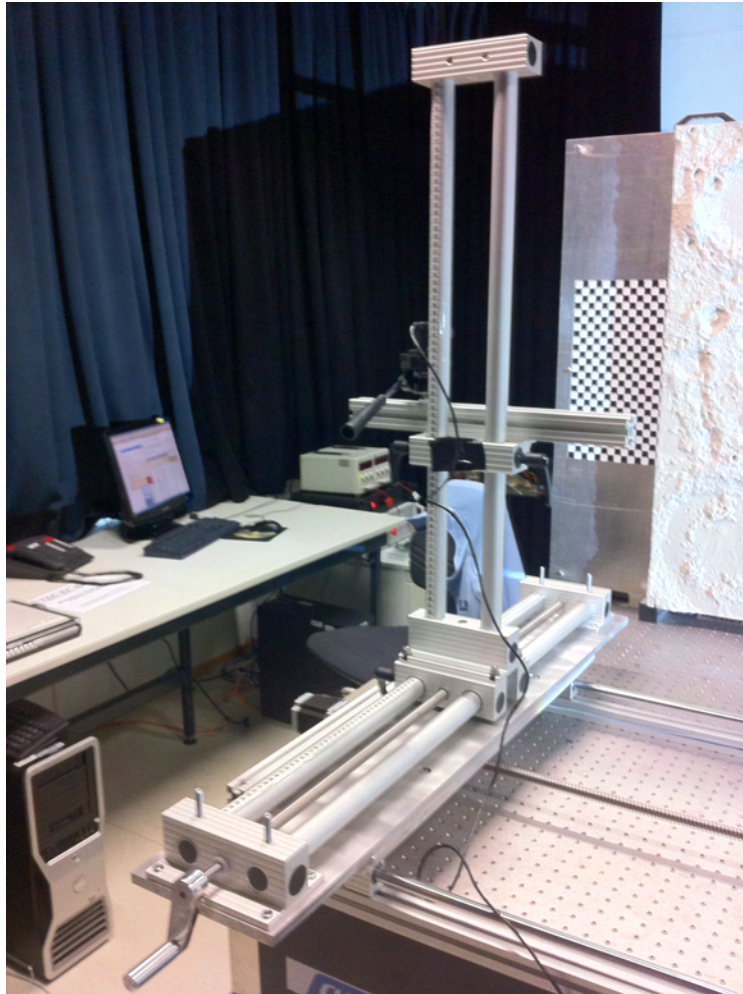


Figure 5.10 – Camera robotic platform in front of the camera calibration pattern at the left of the mock-up in Visilab

5.2.5 Illumination system

Strictly fulfilling the requirement calling for parallel rays of white light spread with a uniform flux over the surface of the mock-up would require to have at least 1-m-wide Fresnel lens in the lab along the shorter side of the mock-up to collimate a light beam with the appropriate flux pattern coming from a white source. Such an option would not fit in the lab though. In addition, it does not exist off-the-shelf from commercial illumination providers so a complete optical system would have to be designed from scratch which is out of the scope of this study and was thus discarded.

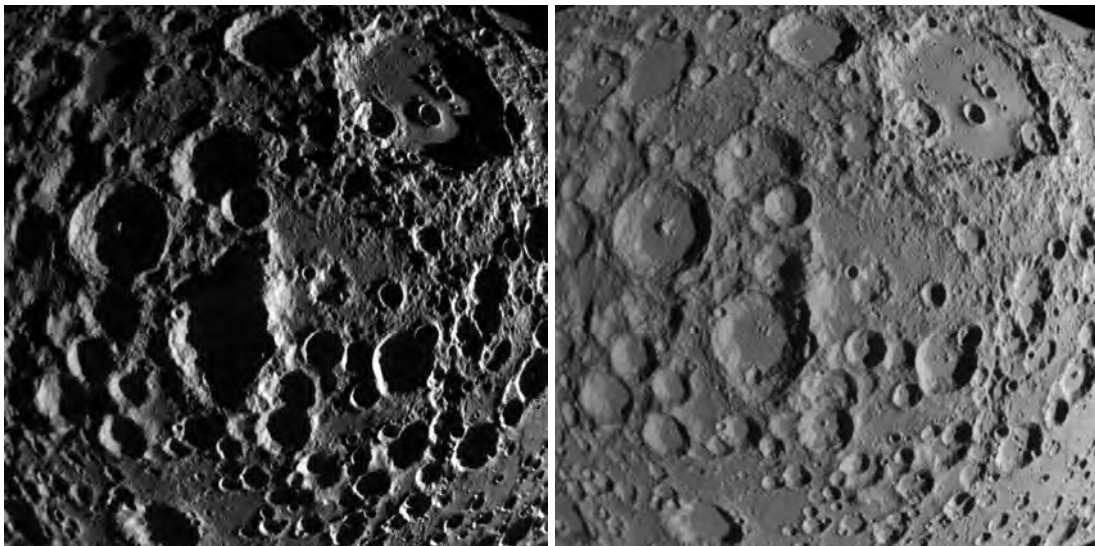
LED lamps were first considered because of their low-divergence beams and their nearly-uniform flux. However, the beamwidth appeared too small for our application: it was only about 30 cm wide at a 2-meter distance while the mock-up is 1-m wide. The solution consisting in assembling together several LED lamps in a grid network to

Table 5.5 – Course ranges of the camera robotic platform in *Visilab*

Axis	Course Range
Downrange	900 mm
Crossrange	658 mm
Altitude	717 mm
Pitch	170 deg

create a wider beamwidth appeared expensive, impractical, and would create higher-flux areas where the individual beams overlap.

The solution advised by the commercial illumination providers contacted was to look into cinema and theater spot lamps. The optical principle of those is simple: a halogen lamp is placed behind a Fresnel lens which outputs a low-divergence beam. However, we noticed that a simple retroprojector is based on the same optical scheme. In addition, the lower lamp power of those devices is more adapted to our use at short distance in *Visilab* than an expensive cinema-class lamp. As test images looked realistic, we decided to use a standard 500-W retroprojector as the illumination system in *Visilab*. Example images are shown in Figure 5.11.



(a) Sun elevation: 10 deg

(b) Sun elevation: 30 deg

Figure 5.11 – Example images of the same area in *Visilab* with different illumination conditions

Analysing the fitness of the retroprojector solution with respect to mission-representative requirements, we first reckon that the light rays are not strictly parallel in *Visilab* as they can be assumed to originate from a point source located close to the reflectors of the retroprojector which is not at an infinite distance like the Sun, but a low-divergence beam is as good as we can get in the lab. Furthermore, the light beam divergence in terms of shadow pointing directions is not noticeable at naked eye. As of the white light criterion, halogen lamps are incandescence devices and thus provide a continuous light spectrum which was judged white enough for *Visilab*. Eventually, the least representative aspect which can be pointed out in these images is the surface flux decrease from right to left due to the fact that light rays are not parallel. Actually with the point source model mentioned earlier for the retroprojector, the illumination

flux decreases as an inverse quadratic function of the distance to the point source, hence the surface flux variation on the mock-up plane. This default is obviously not representative of the true lunar conditions but it does not matter so much for *Lion* as it is based on feature points using local intensity gradient information whereas the non-uniformity of the flux is only noticeable at global image scale.

It shall be noted that the shadow cast by the camera and its support platform on the mock-up is another non-representative phenomenon in *Visilab*. However, it was only visible for light elevation angles above 45 deg while our test sequences were acquired with a maximum 30-deg angle, representative of lighting conditions for a lunar landing at the south pole.

5.3 TEST BENCH OPERATIONS

A test bench needs to be operated carefully in order to acquire workable experimental data. A view of *Visilab* during this data acquisition process is shown in Figure 5.12. Evaluating the performance of a vision-based navigation system requires to acquire an image sequence for which the pose of the camera with respect to a reference navigation frame attached to the observed scene is measured with other means at each acquisition time in order to evaluate the navigation error. This reference pose is called *ground truth* throughout this thesis. This section first describes the image acquisition process and then ground truth determination.

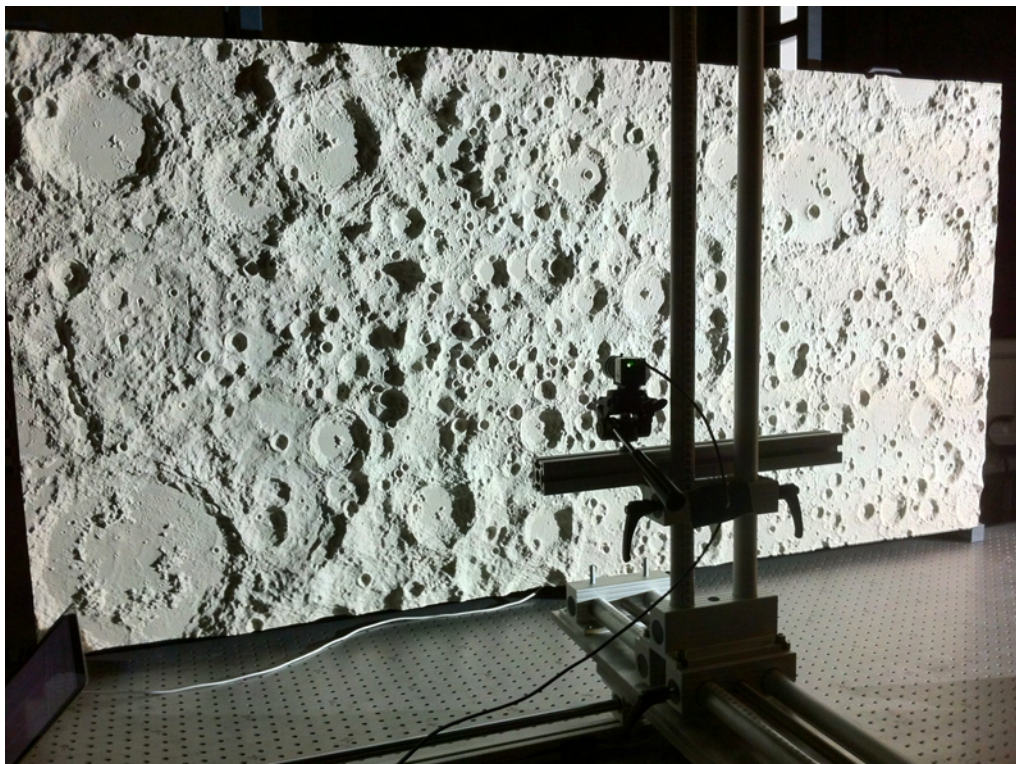


Figure 5.12 – The *Visilab* test bench during data acquisition at ESA-ESTEC

5.3.1 Image acquisition

1024 × 1024 grayscale images were obtained after cropping the 1280 × 1024 raw frames acquired through the USB interface of the camera connected to a laptop running *Linux* in the lab.

Optical lens setting

The optical aperture and the focus of the lens must be set before data acquisition. *Focus* determines the distance at which objects appear sharp in the image. In practice with the limited sensor resolution, this distance of focus is spread over an interval which is called the *depth of field*. *Aperture* sets the amount of light entering the optics and it needs to be set first as it impacts the depth of field: the smaller the aperture, the larger the depth of field.

Aperture In *Visilab*, the largest depth of field was preferred as it allows the camera to work over a greater range of altitudes. That led to set the aperture at the lowest possible value for which the image quality did not suffer from either light diffraction at the edge of the lens when the iris is too closed, or from the blur effect due to vibrations accumulating over the longer numerical exposure times needed with smaller apertures.

Focus With the aperture fixed, the focus ring can be adjusted to get sharp images. In practice with the lenses used in *Visilab*, setting the focus at ∞ was the best solution as it allowed to get sharp images from 6 cm to an infinite distance, and no images were acquired below 6 cm as no lunar-representative surface details are visible at such low altitudes.

Camera calibration

A camera images the world through a projection transformation which can be represented as a 3×4 matrix P in the pinhole camera model and decomposed as

$$P = K [R | t]. \quad (5.2)$$

P transforms a 3D physical world point into a 2D image point, which are characterized respectively by a 4-vector and a 3-vector of homogeneous coordinates (Hartley and Zisserman, 2003). The 3×3 matrix R and the 3-vector t respectively describe the rotation and translation from the world frame to the camera frame, they are called the *extrinsic* parameters of the camera for each image. Their proper estimation is the objective of vision-based navigation. The 3×3 matrix K contains the so-called *internal* parameters necessary to convert the direction of a ray in the camera frame into a pixel position on the sensor, or vice versa. There are five internal parameters: the focal length in terms of pixel dimensions in the x and y directions, the skew which denotes if the pixel angles are right or not, and the 2D pixel coordinates of the principal point.

Equation (5.2) is equivalent to the *Lion* EKF measurements model of Equation (3.6) with no noise. K transforms normalized image coordinates into pixel ones and must be known and determined before the mission in practice since navigation systems use the geometry of rays traced from pixel positions to infer back the pose of the camera. The estimation of K is called *calibration* and it must be performed before each data acquisition in *Visilab*. In practice, calibration also needs to estimate the optical

distortions which add non-linear effects into Equation (5.2). In *Visilab*, raw images are processed to remove the distortions before being sent to *Lion* thanks to the model estimated during calibration. As can be seen in Figure 5.11, radial distortions are indeed quite present in *Visilab* images with the 3.5-mm lens. The estimation of the 6th order of radial distortion and the skew had to be taken into account to get best calibration performances.

5.3.2 True pose computation

Ground truth, i.e. the true pose of the camera frame with respect to the reference navigation frame is computed using additional intermediary frames in *Visilab*.

Visilab frames

The mock-up reference navigation frame $\{r\}$ in *Visilab* is aligned with the bottom right corner at the back of the mock-up panel. It is equivalent to the lunar surface-fixed frame $\{g\}$ employed in planetary landing. To determine the pose of the camera, a dedicated calibration pattern was stuck on the aluminium plate at the left of the mock-up. The position and orientation of the frame $\{p\}$ tied to this pattern are known with respect to the reference frame. Before operating in *Visilab*, the user must initialize the position and orientation of the camera facing this pattern, corresponding to the pose of frame $\{c\}$, using a standard 2D/3D visual pose determination algorithm and the known pattern grid coordinates (Szeliski, 2011). The camera can then be translated to its final working pose $\{w\}$ for which the final position is measured relatively to $\{c\}$ directly onto the axes. The orientation cannot be changed after initialization between $\{c\}$ and $\{w\}$ during an acquisition sequence though. Figure 5.13 shows the configuration of *Visilab* and the associated frames when the camera is in initialization pose in front of the calibration pattern. The origin and axis orientation of each frame is defined in Appendix D.

Transformations between frames

The rigid frame transformation from the camera in working pose to the mock-up reference navigation frame provides the ground truth and can be written as the composition of three sub-transformations between the four *Visilab* frames:

1. Camera working frame $\{w\} \xrightarrow{f_1} \text{Camera calibration frame } \{c\}$,
2. Camera calibration frame $\{c\} \xrightarrow{f_2} \text{Calibration pattern frame } \{p\}$,
3. Calibration pattern frame $\{p\} \xrightarrow{f_3} \text{Mock-up frame } \{r\}$.

A rigid transformation can be represented minimally as 6-vector f such that $f = [r^T \ t^T]^T$ which transforms a point x into a point y in 3D Euclidean space according to

$$y = r \star x + t. \quad (5.3)$$

\star is the rotation operator. $r = \theta n$ is the rotation vector between the axes of the two frames, with θ the rotation angle and n the unit axis of the rotation. t is a translation vector. We choose to use this transformation as a coordinate change such that injecting $x = \mathbf{0}$ in Equation (5.3) makes y the position coordinates of the origin of the initial

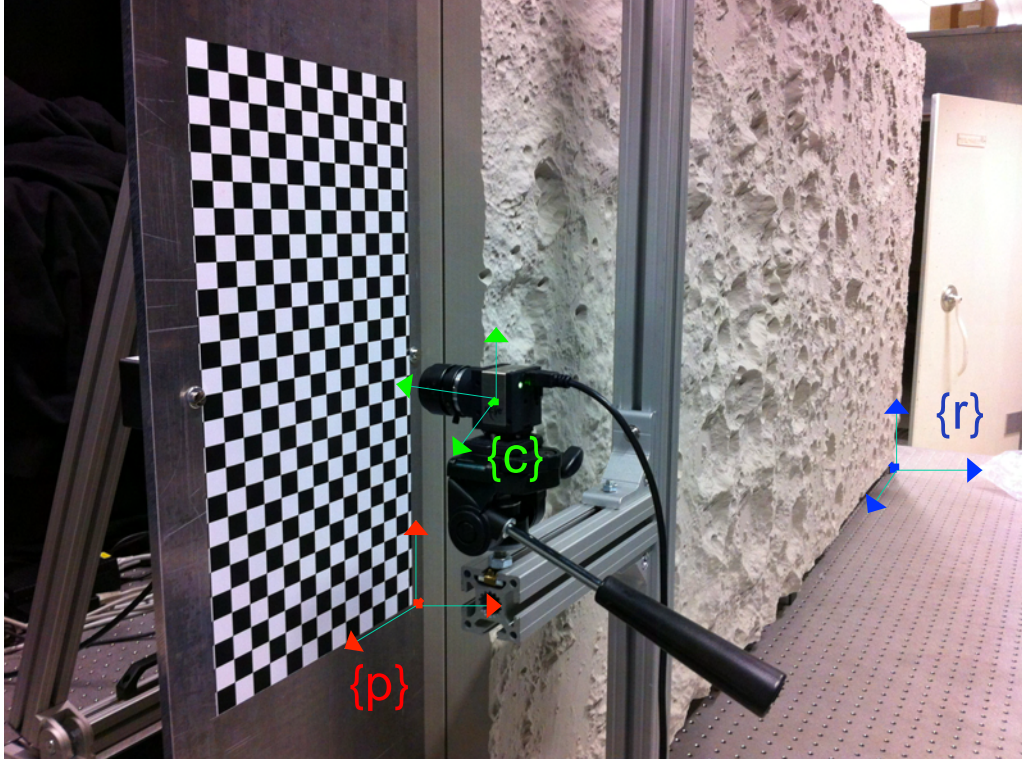


Figure 5.13 – *Visilab* frames with the camera in initialization pose facing the calibration pattern. Only the frame $\{w\}$ is not visible here, it represents the working pose of the camera when it is translated over the mock-up to acquire navigation test images after initialization.

frame in the axes the final frame, e.g. the coordinates of the optical center of the camera in the axes of the mock-up reference frame for ground truth.

Using this rigid body model, ground truth can thus be computed for each acquired image as the composed transformation f such that

$$f = f_3 \circ f_2 \circ f_1 \quad (5.4)$$

where the composition of $f_A = [r_A^T \ t_A^T]^T$ by $f_B = [r_B^T \ t_B^T]^T$ is defined as

$$g = f_B \circ f_A = \begin{bmatrix} r_B \circ r_A \\ r_B \star t_A + t_B \end{bmatrix} . \quad (5.5)$$

5.4 ABSOLUTE NAVIGATION ERROR OBSERVABILITY

The computation of the true pose of the camera frame relies on measurements which have a limited precision. In this section, we investigate the uncertainties in the determination of the ground truth and we propose a new model to evaluate how this limited accuracy impacts the absolute navigation error observable in *Visilab*.

5.4.1 Manual ground truth accuracy

The manual ground truth determination process proposed in Subsection 5.3.2 involves a composition of three frame-to-frame rigid transformations for which the uncertainties have been measured and are reported in Table 5.6. This subsection describes how

these measurements have been obtained and eventually derives the overall manual ground truth accuracy in *Visilab*.

Table 5.6 – *Frame-to-frame pose uncertainties in Visilab. The dispersion over each axis is measured with the 3σ value for scalar components, where σ is the standard deviation. The total dispersion of the vector quantities is measured with the 3-RMS value.*

Transformation	f_1	f_2	f_3
δr_x (deg)	0.1	0.1	0.05
δr_y (deg)	0.1	0.1	0.05
δr_z (deg)	0.2	0.2	0.05
Total δr (deg)	0.2	0.3	0.1
δt_x (mm)	1.4	0.9	0.3
δt_y (mm)	0.2	0.9	0.3
δt_z (mm)	0.3	0.3	0.7
Total δt (mm)	1.4	1.4	0.8

Transformation $f_1: \{w\} \rightarrow \{c\}$

Two phenomena introduce uncertainty in the transformation from the camera in working pose to camera in calibration pose: the internal deformations of the support platform structure when the camera is translated, and the misalignments between the optical table and the back panel the mock-up. Transformation f_1 can thus be decomposed into two virtual sub-transformations and Table 5.6 actually shows the uncertainty of their composition.

The first virtual transformation involves no motion but only the uncertainties associated to camera support platform structure deformation. To evaluate it, the camera was moved randomly from the calibration pose and brought back at the calibration pose 30 times. More samples would make the estimation more statically relevant but are practically time-consuming in *Visilab*. After each iteration, the new pose of the camera in front of the calibration pattern was computed using the same 2D/3D visual pose determination algorithm as in ground truth determination. Table 5.7 shows the uncertainties of the camera pose parameters due to structure deformations of the camera support platform.

The second virtual transformation only implies first-order uncertainty in translation which are due to the angular misalignments of the optical table with respect to the mock-up. These misalignments were measured within the 0.1-deg accuracy of the inclinometer available in *Visilab* for the vertical x_m axis, and within 0.25-mm ruler precision over the 2-m length of the optical table for the horizontal y_m axis, which is equivalent to $\text{atan}(0.25/2000) * 180/\pi \sim 0.01$ deg. Misalignments create translation uncertainty through a lever arm effect, which was assumed at 1-m over each axis due to camera translation range. The manufacturing flatness uncertainty of the table over the remaining z_m axis are neglected.

Transformation $f_2: \{c\} \rightarrow \{p\}$

The uncertainties of the rigid frame transformation from the camera in calibration pose to the calibration pattern which are reported in Table 5.6 were computed us-

Table 5.7 – *Uncertainty due to the deformation of the camera support platform in Visilab. The dispersion over each axis is measured with the 3σ value for scalar components, where σ is the standard deviation. The total dispersion of the vector quantities is measured with the 3-RMS value.*

δr_x (deg)	0.1
δr_y (deg)	0.1
δr_z (deg)	0.2
Total $\delta \mathbf{r}$ (deg)	0.2
δt_x (mm)	0.2
δt_y (mm)	0.2
δt_z (mm)	0.2
Total $\delta \mathbf{t}$ (mm)	0.3

ing the dispersion of the image reprojection errors of the 36 calibration pattern grid points in the 2D/3D visual pose determination algorithm used for ground truth determination and the Jacobian matrix of the camera projection function based on Equation (3.19) and Equation (3.21) (Hartley and Zisserman, 2003).

Transformation $f_3: \{p\} \rightarrow \{r\}$

A DEM of the mock-up was realized by DLR after it was milled from the input DEM. To create it, each of the two 1-m-wide square tiles forming the mock-up was laid down on a granite table for laser scanning. The granite table employed was flat within a 10^{-5} -m error tolerance off the plane. This can also be assumed to be the flatness accuracy of the back panel of the tiles, where the origin of the mock-up frame $\{r\}$ is located. After individual scanning, the two tiles were connected to each other while leaning on the granite table and terrain strips at the junction were re-scanned to determine alignment errors and construct the global DEM. A picture of the scanning process on the granite table is displayed in Figure 5.14.

The aluminum bars attached to the back panels of the mock-up and visible in Figure 5.9 were also employed to attach the aluminum plate on which the calibration pattern defining the origin of frame $\{p\}$ is. The bars are screwed into threaded holes in the back panels of the tiles. Because the mock-up assembly was not constrained during scanning and appears quite elastic, the screwing of the bars was achieved with the requirement that the mock-up shall not be deformed with respect to its resting-free shape so that the DEM remains intrinsically valid. After screwing, the bars appeared to lay flat in contact over the back-panel of the left edge of the mock-up, on the side where the calibration plate is mounted. A flatness deviation tolerance of 0.7 mm over 1 m is specified in the data sheet of the aluminum bars (KANYA, 2012). The tolerance was assumed equivalent to a 3σ dispersion and conservatively we kept it at the 0.7-mm value for the translation dispersion along the z_r axis over the 25 cm which can separate the origin of $\{p\}$ from the left edge of the mock-up. The flatness tolerance of the granite table is about tens of micrometers, thus order of magnitude less than the bars and its effect was consequently neglected here. The ruler precision used to position the calibration pattern is 0.3 mm both on x_r and y_r . The attitude uncertainties are also dominated by the flatness tolerance of the bar over the 1-m height of the mock-up leading to $\text{atan}(0.7/1000) * 180/\pi = 0.05$ degree about the axes.



Figure 5.14 – Mock-up lying on the granit table and being scanned for DEM at DLR

Frame transfer covariance composition

Pennec and Thirion (2004) studied the propagation of uncertainty for rigid motions. The uncertainty of a rigid transformation characterized by a 6-vector \mathbf{g} is described by 6×6 covariance matrix \mathbf{P} . When \mathbf{g} is a composition of frames $(\mathbf{f}_A, \mathbf{P}_A)$ by $(\mathbf{f}_B, \mathbf{P}_B)$ like in Equation (5.5), its covariance is determined at first order by

$$\mathbf{P} = \mathbf{J}_A \mathbf{P}_A \mathbf{J}_A^T + \mathbf{J}_B \mathbf{P}_B \mathbf{J}_B^T, \quad (5.6)$$

with the \mathbf{J}_A and \mathbf{J}_B Jacobian matrices defined as

$$\mathbf{J}_A = \frac{\partial(\mathbf{f}_B \circ \mathbf{f}_A)}{\partial(\mathbf{f}_A)} = \begin{bmatrix} \frac{\partial(\mathbf{r}_B \circ \mathbf{r}_A)}{\partial(\mathbf{r}_A)} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_B \end{bmatrix}, \quad (5.7)$$

$$\mathbf{J}_B = \frac{\partial(\mathbf{f}_B \circ \mathbf{f}_A)}{\partial(\mathbf{f}_B)} = \begin{bmatrix} \frac{\partial(\mathbf{r}_B \circ \mathbf{r}_A)}{\partial(\mathbf{r}_B)} & \mathbf{0} \\ \frac{\partial(\mathbf{r}_B^* \mathbf{t}_A)}{\partial(\mathbf{r}_B)} & \mathbf{I}_3 \end{bmatrix}, \quad (5.8)$$

and \mathbf{R}_B the rotation matrix associated to \mathbf{r}_B .

Alternatively, the error covariance matrix containing the uncertainties of a composed transformation can be approached without a first order assumption by running Monte Carlo simulations if the distribution model of random processes causing the uncertainties are known.

Both methods were tested in *Visilab* to determine the covariance matrix of the composed transformation \mathbf{f} in Equation (5.4) which represents the ground truth. They led to similar accuracy results of 4.5 mm in position and 0.5 deg in attitude (3 RMS). The results obtained with the covariance matrices \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 of the frame-to-frame transformations made from the measurements in Table 5.6.

Table 5.8 details the ground truth accuracy results for the camera in a test-representative pose in *Visilab* with respect to the mock-up axes. These figures correspond to 20,000 runs of Monte Carlo simulation and assuming Gaussian error distributions.

Table 5.8 – Manual ground truth accuracy in *Visilab* expressed along the mock-up axes. The dispersion over each axis is measured with the 3σ value for scalar components, where σ is the standard deviation. The total dispersion of the vector quantities is measured with the 3-RMS value.

δr_x (deg)	0.3
δr_y (deg)	0.3
δr_z (deg)	0.3
Total $\delta \mathbf{r}$ (deg)	0.5
δt_x (mm)	3.1
δt_y (mm)	2.7
δt_z (mm)	1.7
Total $\delta \mathbf{t}$ (mm)	4.5

5.4.2 Observable absolute navigation error model

Errors made in the determination of the true pose of the camera concern both descent and orbital images acquired in *Visilab*. As a consequence, the ground truth accuracy

evaluated in the previous subsection may eventually affect significantly the absolute navigation error observed during the hardware tests, even if the map-based state estimation performed in *Lion* works flawlessly. This effect needs to be modeled and evaluated as well.

Figure 5.15 illustrates the different frames employed to derive a new model of the absolute navigation error observable in *Visilab* with limited ground truth accuracy. In black is represented the true configuration of the lab. $\{r\}$ is the mock-up reference frame. $\{c_{dsc}\}$ and $\{c_{orb}\}$ are respectively the true descent and orbital camera frames, of which the knowledge is limited by ground truth accuracy. In blue, $\{\hat{c}_{dsc}\}$ is the descent camera frame estimated by *Lion*, which depends on absolute navigation accuracy. Eventually in red, $\{\bar{r}\}$ is the misaligned reference frame due to the ground truth error made on $\{c_{orb}\}$. Due to this misalignment, a perfect map-based navigation algorithm would estimate the camera pose not in $\{c_{dsc}\}$ but in $\{\bar{c}_{dsc}\}$. We assume the internal coherence of the map is not corrupted by this misalignment though, which is plausible with the ground truth accuracy determined earlier.

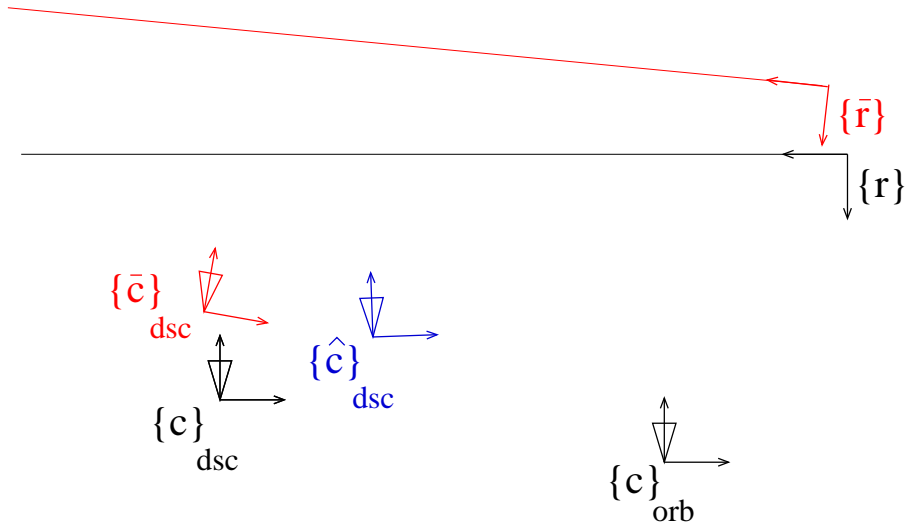


Figure 5.15 – True and estimated frames used to derive the absolute navigation error model in *Visilab*.

Using compositions of rigid body transformations like in Subsection 5.3.2, these frames enable to break down the absolute navigation error f_{obs} measured in *Visilab* defined as the rigid frame change from the estimated to the true descent camera frame, according to the following model

$$f_{obs} = f_{\hat{c}_{dsc}}^{c_{dsc}} = f_r^{c_{dsc}} \circ f_{c_{orb}}^r \circ f_{\bar{r}}^{c_{orb}} \circ f_{\hat{c}_{dsc}}^{\bar{r}} . \quad (5.9)$$

$f_r^{c_{dsc}}$ and $f_{c_{orb}}^r$ are respectively the descent and orbital camera ground truth poses, $f_{\bar{r}}^{c_{orb}}$ is a zero-uncertainty intermediary transformation, and $f_{\hat{c}_{dsc}}^{\bar{r}}$ is the map-based navigation estimate from *Lion*.

This model is a useful system engineering tool to size navigation error budgets, which can be analyzed at various levels of decomposition. For instance, $f_{\bar{r}}^r = f_{c_{orb}}^r \circ f_{\bar{r}}^{c_{orb}}$ represents the frame transfer from the misaligned reference frame back-projected from the orbiter camera to the true reference frame. This specific frame transfer correspond to the so-called *map-tie errors* in planetary landing literature.

By injecting the 4.5-mm and 0.5-deg ground truth accuracy determined earlier, we have been able to compute the *Visilab* contribution to the observable error, namely

the $f_r^{disc} \circ f_{Corb}^r \circ f_{\bar{r}}^{Corb}$ term in Equation (5.9): 8.6 mm in position, 1 deg in orientation (3 RMS). That corresponds to the error which would be measured if the tested navigation algorithm had a perfect output with a zero-mean error and null covariance matrix. With the scales defined for the test scenarios in Chapter 6, 8.6 mm is equivalent to at least 72.6 m on the Moon. This could be sufficient to prove a navigation system can converge within 100 m of pinpoint landing precision, but evaluation would gain from more observability of the proper navigation performance. Since this *Visilab* contribution to the observable error depends essentially on ground truth accuracy, we designed a new method to make it finer.

5.5 VISUAL GROUND TRUTH REFINEMENT

This section describes the visual ground truth refinement algorithm for *Visilab* and computes the new contribution to the observable error .

5.5.1 Method

Without external sensors available in the lab to measure the pose of the camera, we refined the ground truth pose estimation for *Visilab* images using a vision-based algorithm working as follows.

1. The image is virtually rendered using the pose of the camera measured with the manual ground truth procedure, the DEM of the mock-up, and similar illumination conditions.
2. SIFT feature points are matched between the virtual image and the real image acquired by the camera (Lowe, 2004).
3. The 3D surface coordinates of the 1000 best matches are computed by back-projecting the rays from the virtual image focal plane and interpolating them with the DEM.
4. The matches between these 3D points and the associated 2D features points on the *Visilab* image are used to solve a Perspective-n-Point problem (PnP) within a RANSAC framework (Fischler and Bolles, 1981). Camera poses are computed from random selections of 6 points using the EPnP algorithm (Moreno-Noguer et al., 2007).
5. The pose with most inliers is re-evaluated minimizing the reprojection error with the Gauss-Newton algorithm.

The image generation tool used is *PANGU*. SIFT was an easy choice for the matcher as the scene had similar illumination and orientation in the two images. The output of the algorithm is a refinement of the camera ground truth pose.

5.5.2 Performance

The accuracy of this new ground truth determination method was evaluated using the 2D reprojection errors of the 1000 points within the image plane. At first order, the covariance matrix of the pose can be estimated using the Jacobian matrix of the camera projection function (Hartley and Zisserman, 2003) and provides the visual ground truth accuracy results of Table 5.9. The new ground truth accuracy is 0.65 mm

in position and 0.05 deg in attitude (3 RMS). Table 5.10 compares the manual and visual ground truth accuracies in *Visilab*.

Table 5.9 – *Visual ground truth accuracy in Visilab expressed along the mock-up axes. The dispersion over each axis is measured with the 3σ value for scalar components, where σ is the standard deviation. The total dispersion of the vector quantities is measured with the 3-RMS value.*

δr_x (deg)	0.04
δr_y (deg)	0.04
δr_z (deg)	0.01
Total δr (deg)	0.05
δt_x (mm)	0.5
δt_y (mm)	0.5
δt_z (mm)	0.1
Total δt (mm)	0.65

Table 5.10 – *Summary of ground truth accuracies in Visilab (3 RMS)*

Parameter	Attitude (deg)	Position (mm)
Manual ground truth	0.5	4.5
Visual ground truth	0.05	0.65

After injecting this new ground truth accuracy in the observable absolute navigation error model of Subsection 5.4.2, the new *Visilab* contribution to the error is now 2.3 mm in position and 0.12 deg in attitude (3 RMS). This now scales up to 19.3 m on the Moon with the lowest test scenario scaling and makes the performance much more observable than with manual ground truth for the 100-m precision criterion.

CHAPTER CONCLUSION

In this chapter, we derived a set of lunar-representative requirements for the *Visilab* indoor navigation test bench. *Visilab* aims at assessing the performance of *Lion* in realistic hardware conditions. It may also be used to evaluate any type of optical navigation: absolute or relative, active or passive. We designed and built the hardware along with operation procedures. A key performance criterion of a navigation test bench is the accuracy of the ground truth. The manual operations of *Visilab* led to a ground truth accuracy of 4.5 mm in position (3 RMS), evaluated using a rigid body transformation framework. A new budget error model was derived to evaluate the contribution of the test bench itself to the measured absolute navigation error due to the limited ground truth accuracy. With manual ground truth, it showed that *Visilab* engenders a 72.6-m position estimation dispersion (3 RMS), even with perfect navigation. To provide better observability of the navigation performance with respect to the 100-m pinpoint accuracy criterion, a new vision-based ground truth determination system was developed and enhanced the ground truth accuracy down to 0.65 mm (3 RMS), with a *Visilab* contribution of now only 19.3 m (3 RMS) to the absolute navigation error.

The next chapter evaluates the performance of *Lion* in *Visilab* during a complete lunar descent, from orbit to touchdown. In addition, it verifies its robustness to tilt, illumination or sensor change between the orbital and descent data, and terrain topography.

6

HARDWARE PERFORMANCE EVALUATION

CONTENTS

6.1	ACCURACY TESTS	101
6.1.1	NASA LRO orbital data	101
6.1.2	Scalings	102
6.1.3	Nominal descent accuracy evaluation	103
6.2	ROBUSTNESS TESTS	103
6.2.1	Camera sensor change	106
6.2.2	Off-nadir descent camera inclination	107
6.2.3	Illumination change	110
6.2.4	Non-flat terrain	110

EVALUATION of the performance of *Lion* on the lunar-representative hardware test bench *Visilab* was achieved through two types of tests:

1. a dynamic scaled lunar landing sequence to test navigation accuracy,
2. static sequences to test the robustness to various parameters.

In the dynamic accuracy tests discussed in Section 6.1, visual and inertial data at scale for key sequences of the lunar descent flight were generated. Orbital data resolution and accuracy are representative of the NASA LRO mission. Section 6.2 deals with static robustness tests with respect to the parameters which might vary in a real mission: camera sensor change between descent and orbital conditions, off-nadir descent camera inclination, illumination change between descent and orbital conditions, and non-planar terrain topography. In all tests, the images were corrected for distortions prior to being processed in the standard way. Complete result tables for each test can be accessed in Appendix E.

6.1 ACCURACY TESTS

The accuracy of *Lion* absolute visual-inertial tight navigation filter was tested on five 30-second sequences at key altitudes of the lunar descent: 100km, 50 km, 15 km, 10 km and from 5 to 3 km. We assumed an image processing time of 1 second, thus a 1-Hz frame rate provides a new image each time the previous one has been processed. The 30-second duration of each sequence is approximately the convergence time observed in the software simulations of Subsection 4.4.3. Table 6.1 provides mission-representative navigation error requirements at each altitude, in terms of dispersion of the position estimate with respect to the true lander position.

Table 6.1 – *Position error dispersion requirements at lunar descent key altitudes*

Altitude (km)	3-RMS position error requirement (m)
100	1500
50	1500
15	500
10	500
5-3	100

6.1.1 NASA LRO orbital data

The onboard map used as reference in *Lion* is made from orbital images and DEMs acquired by an orbiter prior to the mission. The recent NASA LRO spacecraft has the *LROC* camera and the *LOLA* laser altimeter on board (Chin et al., 2007). *LROC* is made of a set of two Narrow-Angle Cameras (NAC) and one Wide-Angle Camera (WAC). The NACs provide 0.5-m-per-pixel image resolution of all areas above 85.5 degrees of latitudes, or equivalently within 136 km from each pole. The WAC provides a 100-m-per-pixel image resolution over the whole Moon.

Using Earth-based radio tracking, altimetric cross-overs from *LOLA* and a gravity model, Mazarico et al. (2011) managed to determine the orbit with a ground truth accuracy of about 60m (3 RMS). This map-tie error affects the co-registration of the

orbital images with respect to the DEM grid. Although current available DEMs still suffer from visible artefacts as discussed in Subsection 5.2.2, we assume possible to build DEMs with resolution of the order of the 3-RMS ground truth accuracy over the whole Moon, thus 60 m. Actually, due to the high density of altimetry measurements close to the poles, Mazarico et al. (2012) even managed to get consistent DEMs with 10-m pixel resolution. Table 6.2 summarizes the orbital data assumed available either globally on the whole Moon or at the poles.

Table 6.2 – NASA LRO orbital data summary

Coverage	Camera	Image resolution (m per pixel)	DEM resolution (m per pixel)	Ground truth accuracy (3 RMS, m)
Global	WAC	100	60	60
Polar	NAC	0.5	10	10

6.1.2 Scalings

Due to the small size of the mock-up in *Visilab*, three different lunar scalings based on altitude considerations had to be derived to realize the five sequences. Table 6.3 shows which sequence belongs to which scaling, and the associated parameters. A scaling is only representative of a lunar mission with respect to absolute navigation if the orbital image resolution, DEM resolution, and orbital ground truth accuracy altogether scale up to realistic mission values.

Table 6.3 – Scaled parameters for each sequence acquired in *Visilab*

Parameter	Visilab	Scaling 1	Scaling 2	Scaling 3
Altitude (km)	84.7×10^{-5} - 41.3×10^{-5}	100 - 50	15 - 10	$5 \rightarrow 3$
Scale	1	8.5×10^{-6}	4.1×10^{-5}	1.2×10^{-4}
Coverage	N/A	Global	Global - Polar	Polar
LR image resolution (m per pixel)	1.2×10^{-2} - 8.5×10^{-4}	100	100	100
HR image resolution (m per pixel)	5.9×10^{-5} - 2.0×10^{-5}	N/A	25.0	6.25
DEM resolution (m)	0.5×10^{-3}	59.0	12.1	4.2
GT accuracy (3 RMS, m)	0.65×10^{-3}	76.7	15.7	5.5

The coverage parameter describes if the sequences associated to a given scaling benefit from the global or polar LRO data specifications of Table 6.2. The resolution of the so-called Low Resolution (LR) orbital image was adapted to be 100 m like that LRO WAC for each scenario. Close to the pole where the LRO NAC is available, we used respectively a 25-m and 6.25-m High Resolution (HR) image for scenarios 2 and 3. This is worse than the actual 0.5-m resolution of the NAC but anyway even at the lowest altitude of 3 km, the descent camera resolution could not see such details. For DEM resolution and orbital Ground Truth (GT) accuracy, we simply scaled up the *Visilab* data specification since it led directly to LRO-representative values, as shown in Table 6.3.

The trajectory was scaled down for each sequence from a full computer-based lunar descent simulation to get the representative camera pose for each image. 100-Hz inertial data were virtually generated by interpolating between the refined ground truth pose estimates. Bias and noise were added representatively of the areospace-class IMU performance discussed in Subsection B.2.1. The descent camera is always pointing nadir. Illuminations of the descent image and the map are similar.

6.1.3 Nominal descent accuracy evaluation

100 Monte Carlo runs were performed for each of the five sequences, acquired with the 4.5-mm lens in *Visilab*. The initial state errors with respect to the *Visilab* reference frame were randomly sampled over each axis according to zero-mean Gaussian distribution and 3σ values per axis of

- 5 % of the start altitude in position, e.g. 5 km on the 100-km orbit,
- 5 m.s⁻¹ in velocity,
- 0.5 deg in attitude,

with σ the standard deviation. The position uncertainty at initialization is rather large and more a proof of robustness as Earth-based systems would likely be able to initialize the position within 1 km in lunar orbit. Star trackers can initialize very accurately the orientation with respect an inertial frame. This is negligible compared to the orientation error of 0.5 deg with respect to a terrain frame due to the position uncertainty itself, hence the attitude error initialization.

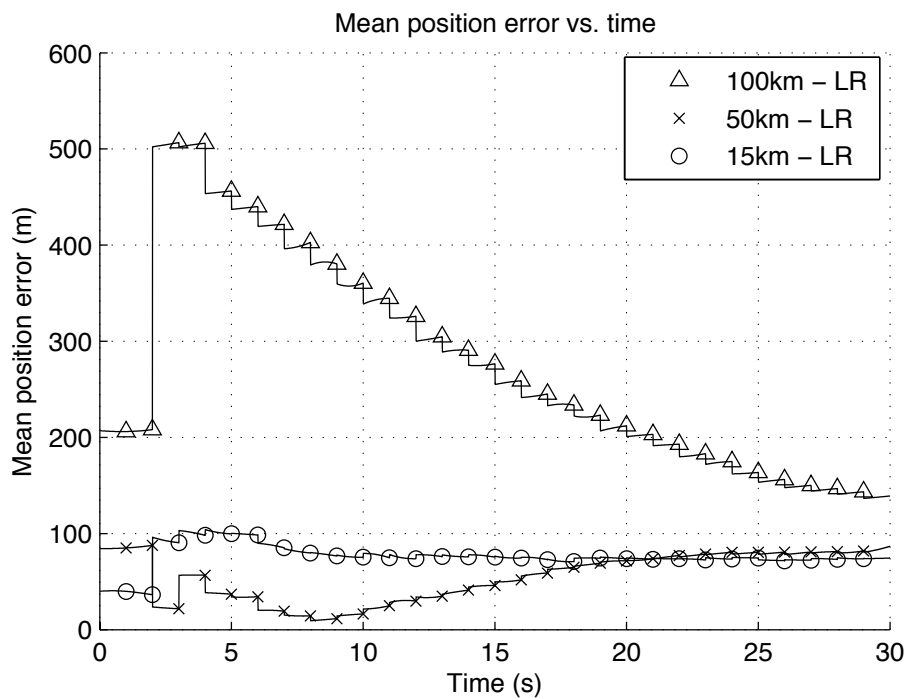
Figure 6.1 and Figure 6.2 show the performance in terms of mean and dispersion of the error versus time for each sequence. For the 10-km sequence and the 5-to-3 km one in Figure 6.2, results are provided both for the LR map and the HR one.

A general result is that the dispersion of the error decreases within each sequence, showing the ability of the filter to converge towards a value which is always less than 150 m in mean, whatever the altitude. At 100 km from an initial error dispersed over 8660 m (3 RMS), the final error mean is 139.1 m for a 3-RMS dispersion of 979.6 m. At 15 km, the mean error is 74.6 m dispersed over 176.2 m. Finally at 3 km with the HR, the error goes down to 4 m in mean and 47.2 m of dispersion (3 RMS), which is already well within the 100-m touchdown requirement of pinpoint landing.

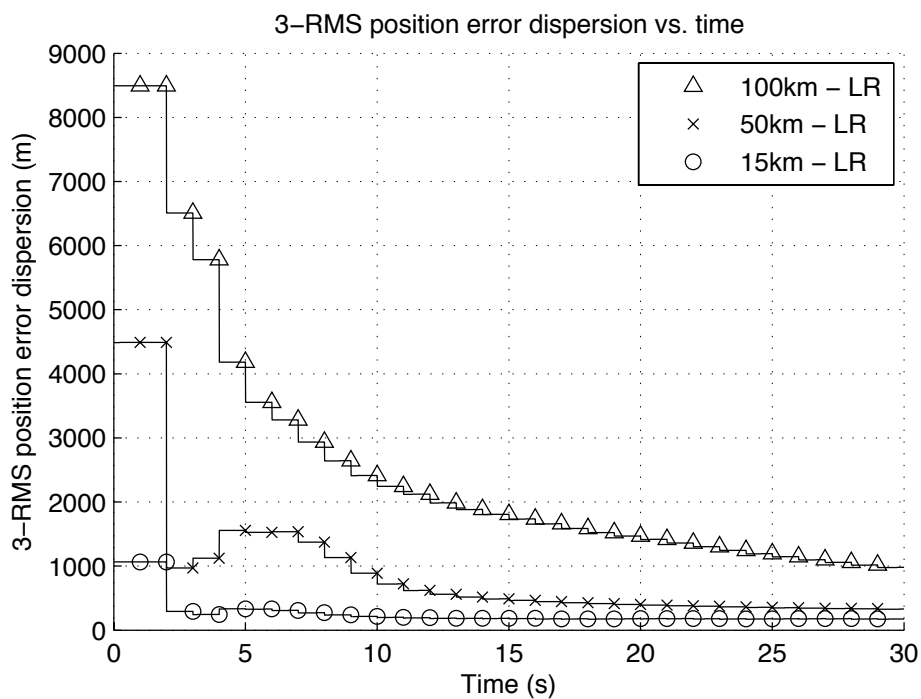
These results obviously include the navigation error associated to *Lion*, but also that due only to *Visilab* only and discussed in Section 5.4. Figure 6.3 scales up the position error environmental contributions of *Visilab*, computed in Section 5.5, and compares them to the final navigation dispersion values measured in the tests. Beyond the limited absolute navigation error observability in *Visilab*, the actual navigation performance of *Lion* may be significantly better than measured here in a real mission as for instance the 3-RMS dispersion was measured at 47.2 m at the 3-km altitude, while in parallel we determined 19.3 m dispersion (3 RMS) due to the test bench only. We demonstrated the accuracy of *Lion* in *Visilab* and its ability to converge from a large initial navigation error in orbit down to error values largely below the 100-m pinpoint landing requirement at low altitudes. Furthermore, this performance includes a *Visilab*-related error contribution due to the descent camera limited ground truth accuracy which will not matter in an actual landing. We can now proceed to a robustness analysis in order to evaluate the variation of performance when some descent parameters vary.

6.2 ROBUSTNESS TESTS

We evaluated the robustness of *Lion* to various parameters: camera sensor change between descent and orbital conditions, off-nadir descent camera inclination, illumination change between descent and orbital conditions, and non-planar terrain topography. These tests were static, running *Lion* filter updates iteratively 10 times on each

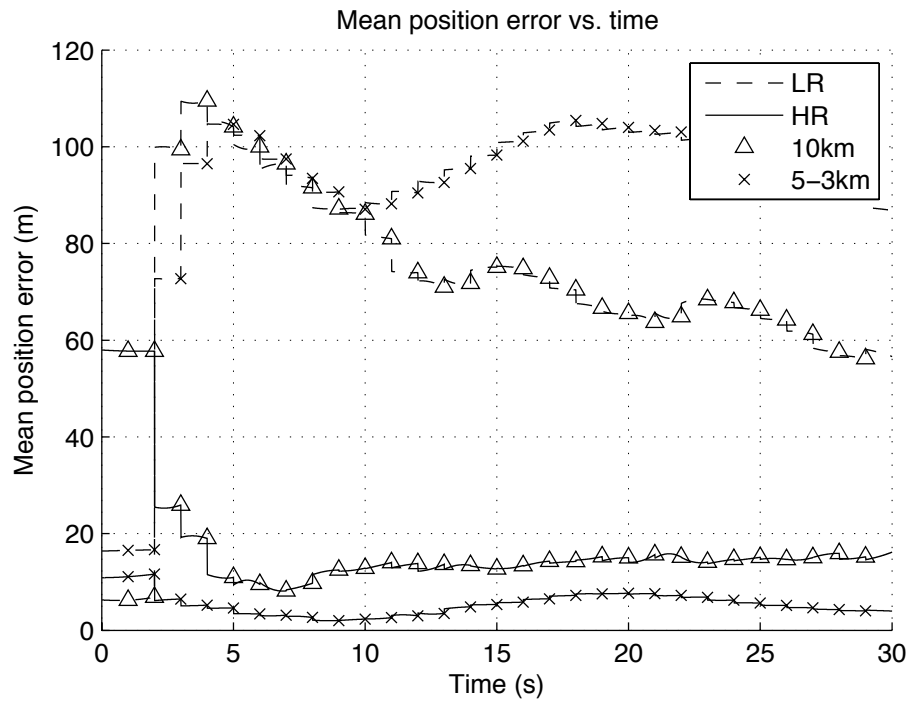


(a) Mean error norm

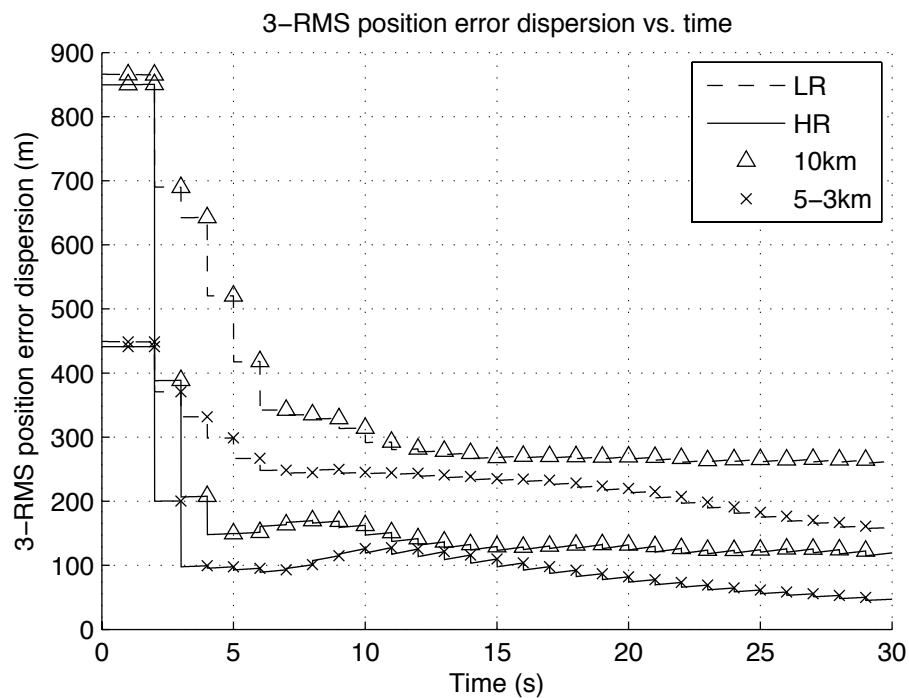


(b) 3-RMS dispersion

Figure 6.1 – Position estimation errors at high altitude with the low resolution map (LR)



(a) Mean error norm



(b) 3-RMS dispersion

Figure 6.2 – Comparison of position estimation errors at low altitude with both the low and high resolution maps (LR and HR, respectively).

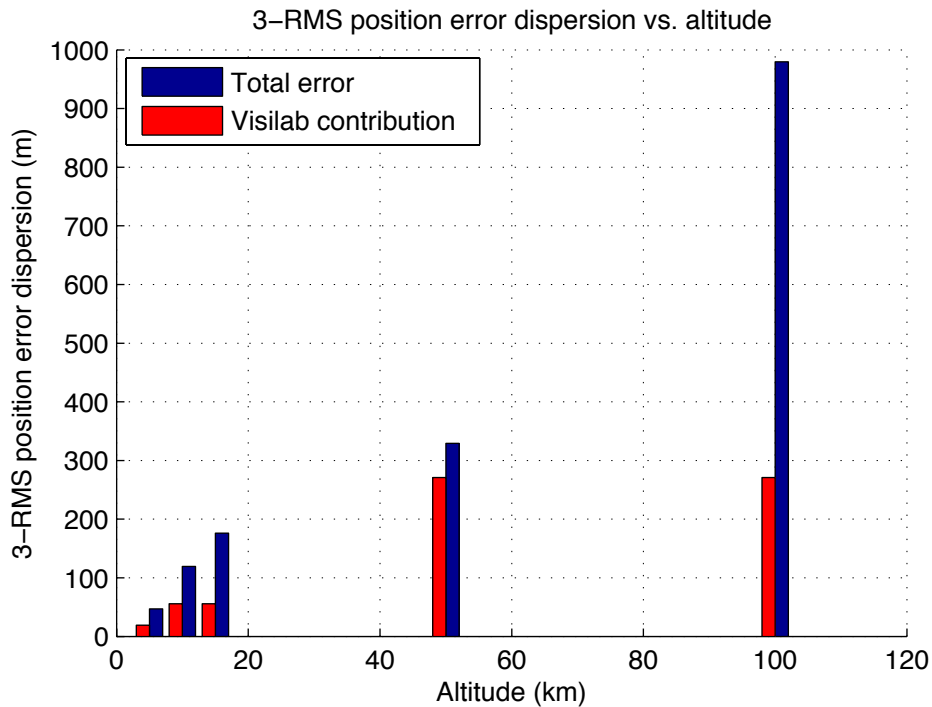


Figure 6.3 – Contribution of Visilab environmental error to final position dispersion

image to assess its ability to converge in each situation. Results from this section cannot be compared directly to those of Section 6.1 as different scalings were used.

Using the 3.5-mm lens, Monte Carlo position and orientation errors were initialized randomly using centered Gaussian distributions respectively at 5 % of the altitude and 0.15 deg (3σ) per axis. All images were acquired at altitudes ranging from 20 to 90 km at scale.

6.2.1 Camera sensor change

The cameras employed to acquire the descent and the orbiter images will most likely be different ones in a real mission. For instance, a mission flying five years from now might be still be using NASA LRO data which have already been acquired. It is thus relevant to evaluate the performance of *Lion* facing a camera sensor change between the orbital and the descent images. Unfortunately, only one camera was available in *Visilab* but we have been able to generate virtual images using *PANGU* in order to simulate data from another camera sensor. An example of a *Visilab* image and a *PANGU* one for the same area is shown in Figure 6.4. The shadow projections are very similar, however as mentioned in Subsection 5.2.5 the mean light flux varies from left to right in *Visilab* while it is constant in *PANGU*. Nevertheless, that should not matter so much for *Lion* as it is based on feature points using local intensity gradient information whereas the non-uniformity of the flux is only noticeable at global image scale. We can also notice a sharpness decrease at the edges of the *Visilab* image due to the correction of optical distortions.

Figure 6.5 compares the *Lion* performance in *Visilab* when using a map based on *Visilab* images or one from *PANGU* images. 200 Monte Carlo runs were performed in each case.

Results show that *Lion* was able to process data from different sensors, one real

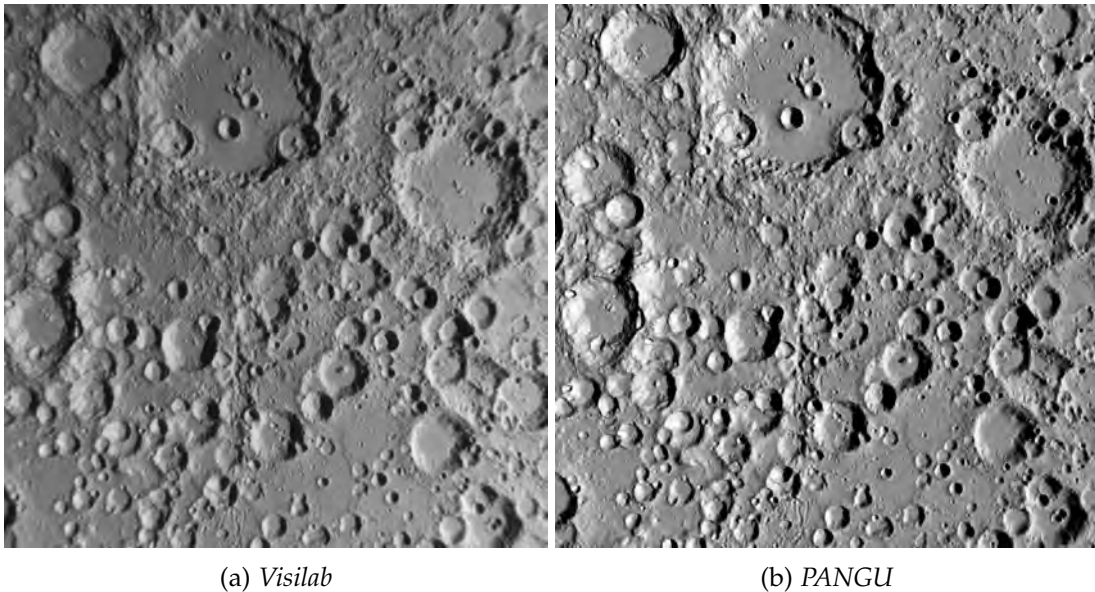


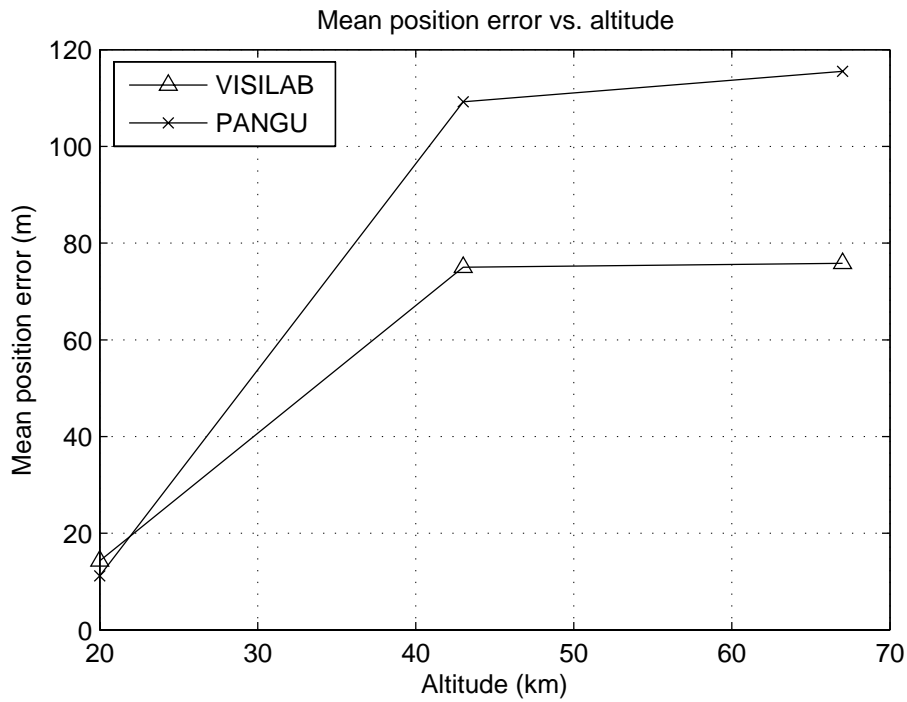
Figure 6.4 – Real image from the *Visilab* camera sensor and virtual image of the same area by *PANGU*.

and one virtual, with similar navigation performances. While the mean error increases from 80 to about 110 m at high altitudes with the *PANGU* map, it actually reduces the dispersion at all altitudes, e.g. from 143 m to 82 m (3 RMS) at a 20 km of altitude. Although we could argue that the *PANGU* images have perfect ground truth accuracy and thus are advantaged, our observable absolute navigation error model from Subsection 5.4.2 could not explain such differences. One possible explanation could be that the absence of distortions on *PANGU* images made the maps more accurate.

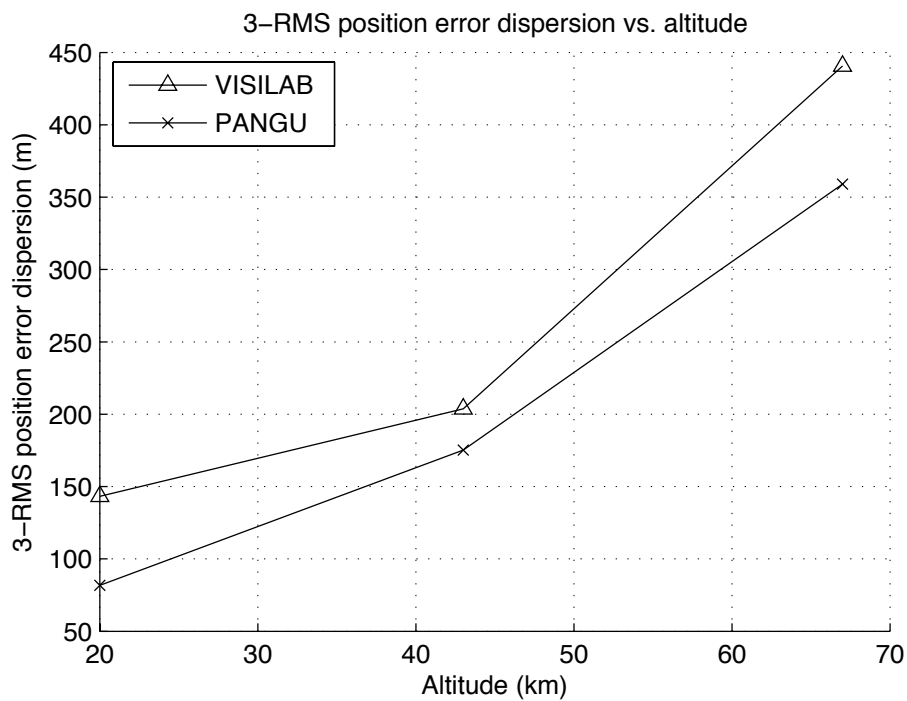
6.2.2 Off-nadir descent camera inclination

In the accuracy tests of Section 6.1, the camera was pointing nadir. However depending on mission scenario and on the camera configuration on the vehicle, the optical axis might be tilted with respect to the local vertical. We tested the robustness to tilt angles varying from 20 to 60 degrees for the descent camera. Results are shown in Figure 6.6. 120 Monte Carlos were run for each inclination angle.

Lion was found to be affected by tilt angles of 40 degrees or more. Indeed, position initialization had to be reduced from 5 % of the altitude to 2 % for the 40 and 60-deg inclination, otherwise no matches could be found. This prevents the use of such inclinations when navigation uncertainties are high. But luckily such angles might only appear during the approach phase late in the descent when the lander is pitching up for the terminal vertical maneuver. This is long after absolute navigation has been turned on and it should already show good enough convergence. Apart from the reduced initialization envelope, mean error converges below 130 m in all cases with no visible influence of the tilt. Dispersion globally increases with tilt. This could be expected as due to the inclined view angle, the footprint a descent camera pixel on the mapped terrain is larger.

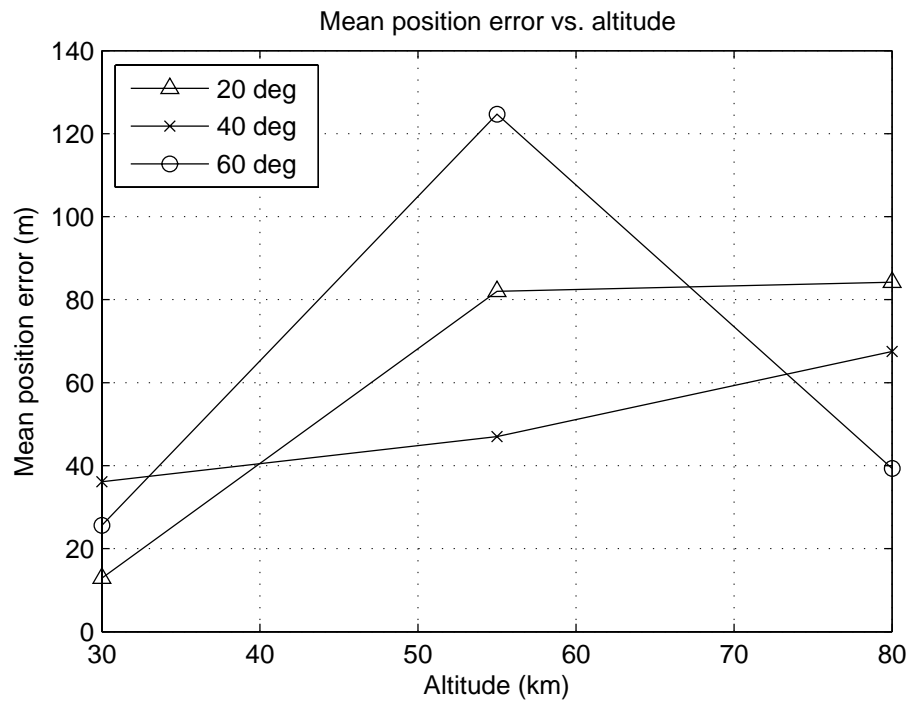


(a) Mean error norm

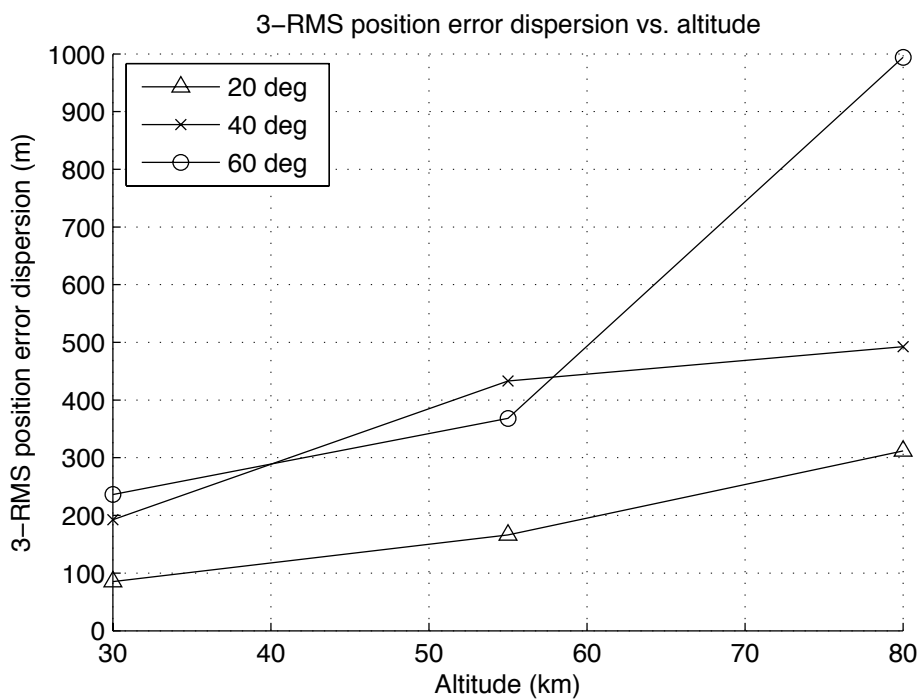


(b) 3-RMS dispersion

Figure 6.5 – Comparison of estimated position error with a Visilab or a PANGU map



(a) Mean error norm



(b) 3-RMS dispersion

Figure 6.6 – Comparison of estimated position error for different camera tilt angles

6.2.3 Illumination change

We evaluated *Lion* performance variations when the illumination at the time of landing varies with respect to that of the map. The azimuth light angle was varied up to 180 deg and the elevation angle within a 20-deg range from a 30-deg elevation reference. 63 Monte Carlos were run for each illumination case. Maps were built from *PANGU* images as the illumination angle range was not sufficient in *Visilab*, descent images are still real ones though. Figure 6.7 shows the results with azimuth being changed while Figure 6.8 shows the influence of an elevation change.

The first significant results is that in all illumination cases the filter converged and improved the initial estimation error. Even with a 90-deg azimuth light change at 20 km of altitude, the system brought the initial 3-RMS 1700-m dispersion down to 680 m, which is more than twice less. Globally, the error mean and dispersion increase with the illumination difference. Performances are more severely affected above 10 deg of elevation change, and 20 deg of azimuth. These angle ranges may be considered as the illumination domain for which pinpoint landing can be considered with *Lion*. In addition to *Visilab* hardware results, we also remind the reader that the pinpoint-class software-based performances of Subsection 4.4.3 were obtained with a 20-deg illumination azimuth change.

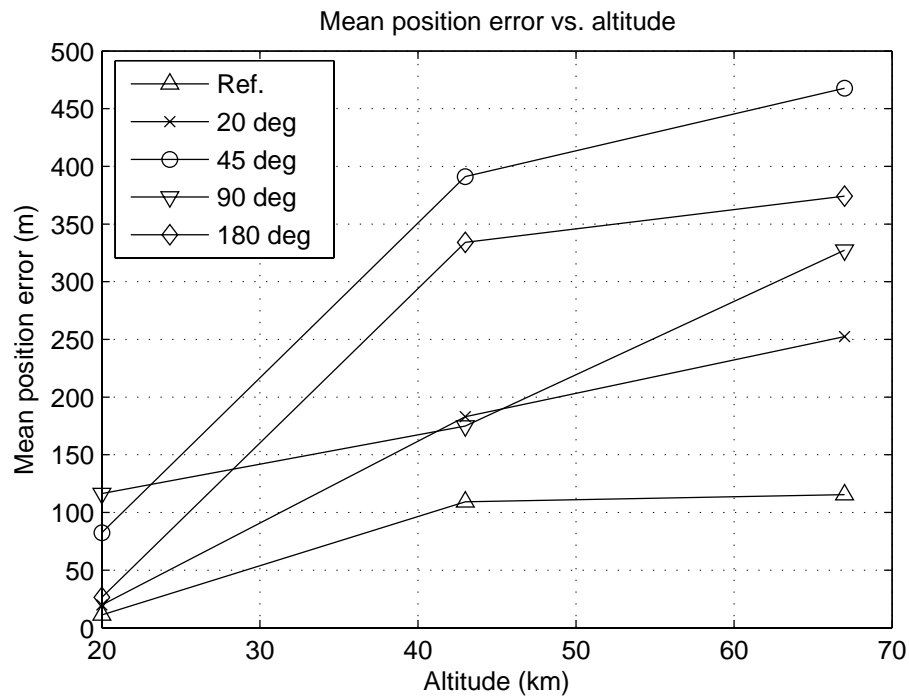
The evolution of the performance *Lion* with illumination changes is evaluated for robustness purposes but we anticipate that such a mission is planned years in advance and that orbital imagery with illumination conditions similar to that of the landing can be used to create the map. In addition for the Moon, another significant result is that we demonstrated in Subsection 6.2.1 that information from the DEM can be used to generate a virtual orbital image with illumination conditions similar to the descent to build the map.

6.2.4 Non-flat terrain

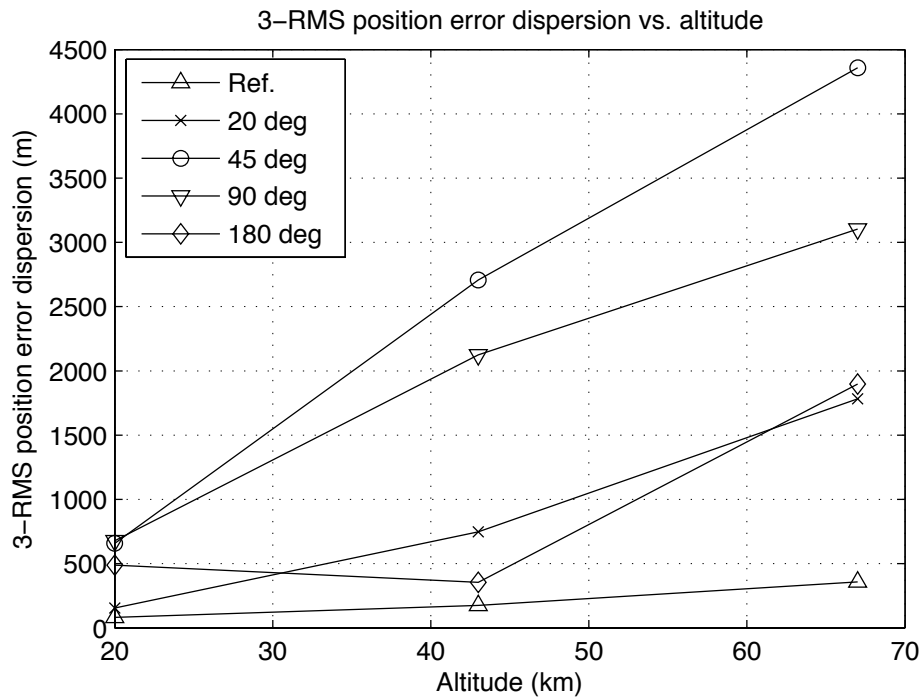
Lion was designed to be fully compatible with any terrain topography, whether it is flat or not. 3D terrains might appear at low altitudes on the Moon depending on the local relief at the landing site selected. We showed robustness to these cases in software in Chapter 4, where the terrain had a 500-m irregular height variation and the trajectory was from 2 km high to touchdown. However, there is an easier-to-characterize 3D phenomenon which is visible for sure on orbit around the Moon: the sphericity of its surface. We compared the performance of the *Lion* system with respect to a modified version making the flat-world assumption discussed in Section 2.1 and often met in literature.

Because the *Visilab* mock-up is globally flat, we could not test the robustness to spherical terrains in hardware directly. Though, we did apply a spheric correction on the *Visilab* DEM to simulate the lunar radius. The new DEM is illustrated in Figure 6.9. It was used in *PANGU* to render virtual descent images based on the actual *Visilab* mock-up topography modified to take into account lunar sphericity at altitudes varying from 50 to 200 km. Such altitudes correspond to orbital or final planet approach phases. 100 runs Monte Carlo were performed at each altitude and for each version of the algorithm. Results are displayed in Figure 6.10.

Lion significantly outperforms its counterpart assuming a flat world which appears clearly biased. At the 100-km parking orbit planned for the ESA lunar lander, the mean error is 23 m for *Lion* versus 1379 m for the flat version, and 3-RMS dispersion is 347 m versus 674 m. Table E.8 actually shows the mean error is nearly-exclusively distributed over the local vertical axis, perpendicular to the plane assumed by the flat-

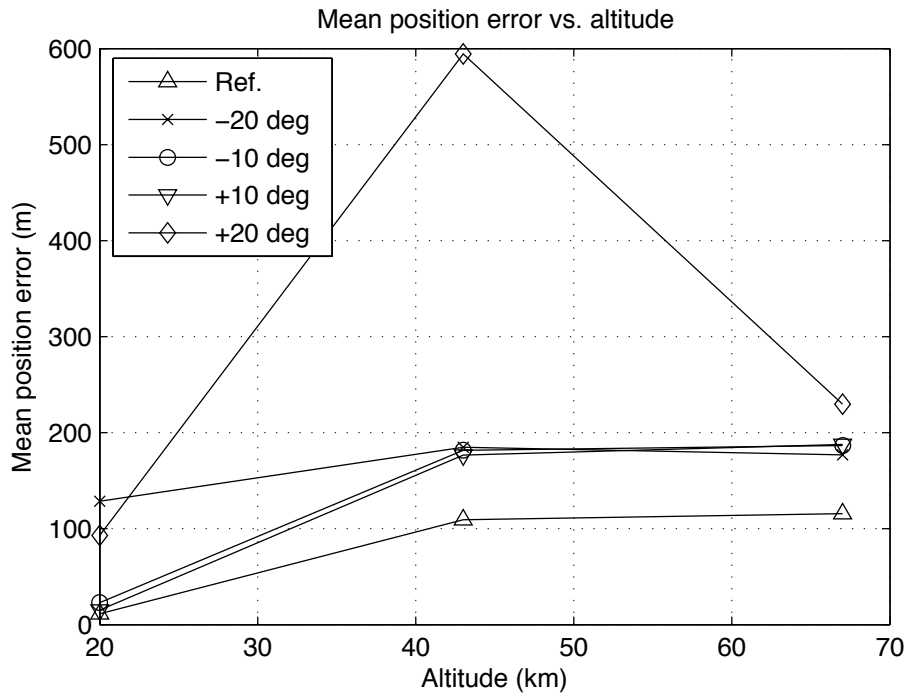


(a) Mean error norm

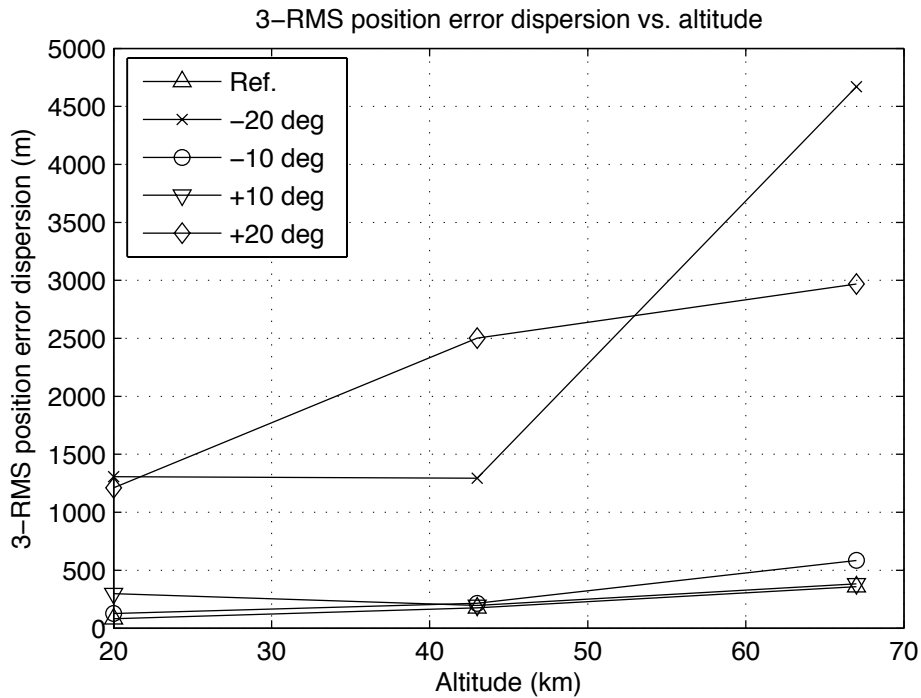


(b) 3-RMS dispersion

Figure 6.7 – Comparison of estimated position error for azimuth light change with respect to the map.



(a) Mean error norm



(b) 3-RMS dispersion

Figure 6.8 – Comparison of estimated position error for elevation light change with respect to the map.

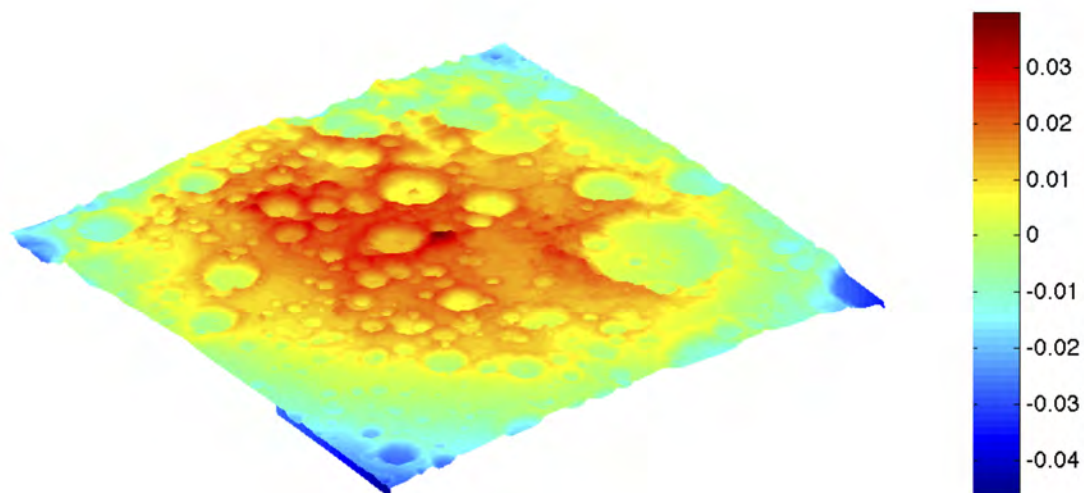
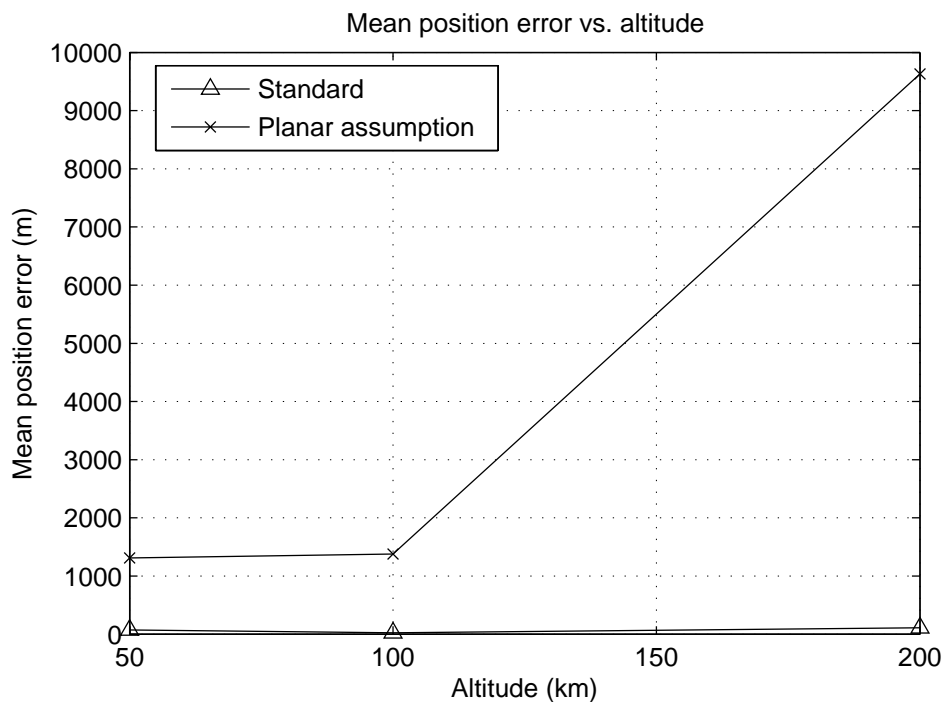


Figure 6.9 – *Visilab* DEM modified with a spherical distortion in order to evaluate the navigation consequences of the flat-terrain assumption in orbit. The height scale is in meter in *Visilab*.

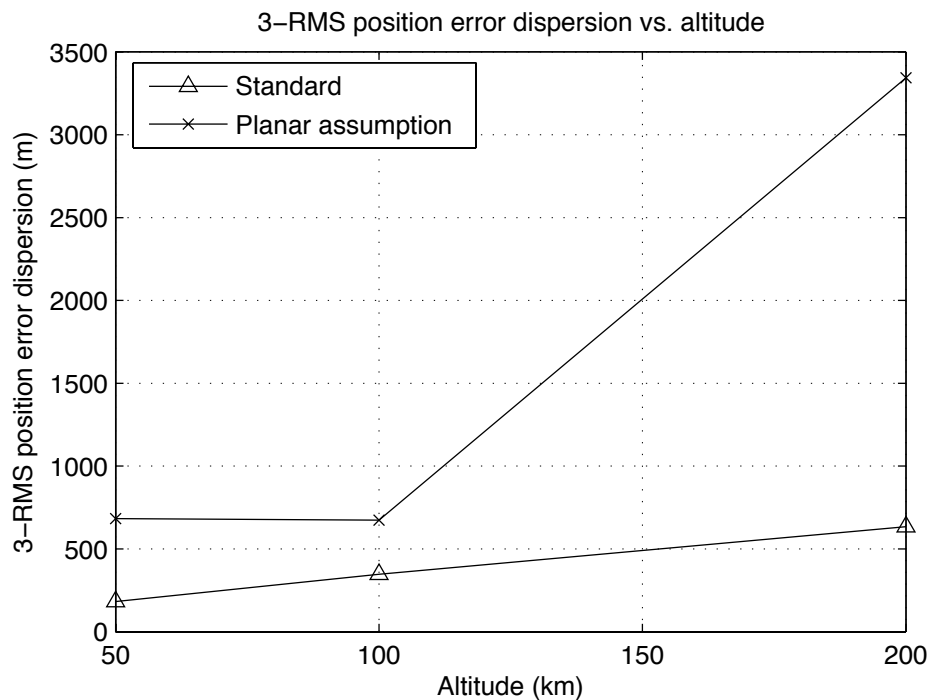
world version. This version paradoxically shows similar performance in terms of error mean and dispersion between 50 and 100 km of altitude, whereas the sphericity effect should be less important at 50 km. We noticed that a strong peak lies at the center of the DEM shown in Figure 6.9, right down under the camera location. At scale for the 50-km image, the camera actually stands virtually 13 cm over the peak for which the height difference with respect to its immediate neighborhood is 4 cm. This rough 30% ratio between surface height variation and the altitude suggests that the 3D relief effects associated to the peak keeps degrading the performance while the effect of the curvature decreases.

CHAPTER CONCLUSION

We ran a scaled lunar descent test in the *Visilab* test bench and demonstrated the accuracy of the estimation of *Lion* at various key altitudes of a lunar landing with a nadir-looking camera. The 4-m mean and 47-m 3-RMS dispersion at 3 km of altitude are compatible with the pinpoint landing requirements. *Lion* was not affected by a change of camera sensor between the orbital and descent images and it still managed to converge with off-nominal variations of illumination or tilt, although performance decreases and was strongly affected over respectively 20 deg and 10 deg of azimuth and elevation light difference, or tilt angles over 40 deg. Eventually, we confirmed the significant performance gain of *Lion* at high altitudes with respect to navigation systems making a flat-terrain assumption when the Moon actually appears spheric.



(a) Mean error norm



(b) 3-RMS dispersion

Figure 6.10 – Comparison of estimated position error at orbital altitudes with and without planar terrain assumption.

7

CONCLUSION

CONTENTS

7.1	MAIN CONTRIBUTIONS	117
7.2	PERSPECTIVES	118
7.2.1	Validation	118
7.2.2	Extending Lion	120

LION is a full absolute visual-inertial navigation system which is able to match details of an image of the surface taken by an on-board camera during a planetary descent with landmarks from a map created before the mission, so as to infer the position and orientation of the vehicle. This chapter summarizes the work done and discusses it to offer perspectives.

7.1 MAIN CONTRIBUTIONS

Lion is based on a tight absolute visual-inertial EKF fusion architecture which keeps the pose estimate at the acquisition time of the last image as an additional static state to take into account time delays associated with absolute image matching. We assessed the navigation performance of this EKF in a point-based simulation assuming ideal absolute measurements, and testing the final part of the trajectory over surface elevation ranges from 0 to 1000 m. The dispersion was below 22 m (3-RMS) at touchdown on all topographies and there was no apparent direct link between the surface elevation range and the quality of the estimation, which appears more affected by the number of landmark measurements. Indeed, the dispersion actually fell down to 2.2 m (3-RMS) at the end of a so-called visual phase, before fewer if any mapped landmarks were left visible and a final inertial drift started to increase the navigation error. The most important result is that with ideal measurements the *Lion* EKF estimation architecture is capable to reach pinpoint landing accuracy whatever the terrain structure.

From these results in ideal point-based simulation conditions, the issue of processing actual images in order to recognize mapped landmarks in them was tackled and it intends to keep a similar level of performance. Three geometric image-to-map matching techniques for generic landmarks have been proposed, implemented along with the tight navigation filter, and evaluated in image-based software simulations representative of the approach phase of a lunar landing. All of them are compatible with any terrain topography, whether it is flat or not, through an efficient use of the camera projection function and a priori pose mean and covariance information available from the filter instead of a planar rectification. Landmark repeatability issues met by most geometric methods, and troublesome on landing trajectories due to significant altitude change, led to completely rethink the way image scale is used in the algorithm. While other algorithms in the literature only proceed to global image scale corrections if ever, we show scale can be efficiently used as an individual descriptor for each landmark in a new method which was selected as the baseline for *Lion*. Based on Harris-Laplace orbital landmarks, it raises the repeatability rates significantly and improves the accuracy down to 20.2 m at the end of the visual phase and 53.8 m at touchdown (3 RMS), with a 20-deg illumination change. These performances are largely below the 100-m accuracy requirement for pinpoint landing and were obtained with significant 3D terrain relief, namely surface height variations of 500 m on a lunar landing approach phase trajectory starting at 2 km.

A lunar-analogue indoor optical test bench called *Visilab* was designed in order to prove that *Lion* can estimate the pose of real camera in a physical environment representative of the Moon. The representativity is ensured by a set of requirements. Although *Visilab* aims at assessing the performance of *Lion* in realistic hardware conditions, it may also be used to evaluate any type of optical navigation: absolute or relative, active or passive. We designed and built the hardware along with operation procedures. A key performance criterion of a navigation test bench is the accuracy of the ground truth. The manual operations of *Visilab* led to a ground truth accuracy

of 4.5 mm in position (3 RMS), evaluated using a rigid body transformation framework. A new budget error model was derived to evaluate the contribution of the test bench itself to the measured absolute navigation error due to the limited ground truth accuracy. With manual ground truth, it showed that *Visilab* engenders a 72.6-m position estimation dispersion (3 RMS), even with perfect navigation. To provide better observability of the navigation performance with respect to the 100-m pinpoint accuracy criterion, a new vision-based ground truth determination system was developed and enhanced the ground truth accuracy down to 0.65 mm (3 RMS), with a *Visilab* contribution of now only 19.3 m (3 RMS) to the absolute navigation error.

We eventually ran a scaled lunar descent test in the *Visilab* test bench and demonstrated the accuracy of the estimation of *Lion* at various key altitudes of a lunar landing with a nadir-looking camera. The 4-m mean and 47-m 3-RMS dispersion at 3 km of altitude are compatible with the pinpoint landing requirements. *Lion* was not affected by a change of camera sensor between the orbital and descent images and it still managed to converge with off-nominal variations of illumination or tilt, although performance decreases and was strongly affected over respectively 20 deg and 10 deg of azimuth and elevation light difference, or tilt angles over 40 deg. Eventually, we confirmed the significant performance gain of *Lion* at high altitudes with respect to navigation systems making a flat-terrain assumption when the Moon actually appears spheric.

7.2 PERSPECTIVES

Two types of future works would be relevant to pursue the assessment of *Lion* as a navigation system: further validation activities, and implementing new extensions.

7.2.1 Validation

The use of Harris-Laplace features in *Lion* is motivated by the fact that they are generic enough to be detected on any type of textured terrains, whatever the celestial body explored. Since *Lion* was only tested in lunar conditions in this thesis, it would be interesting to get test data representative of another environment. In addition, testing on real flight data would allow to test the robustness of the image processing to vibrations and validate the inertial propagation model since a real IMU could not be used at scale in *Visilab*. All these aspects can be tested through an helicopter UAV terrestrial flight experiment which was run in February, 2013 on the Caylus site near Toulouse. The orthoimage and the DEM are at a 20-cm resolution (Sanfourche et al., 2012). The trajectory shown in Figure 7.1 involved several horizontal passes at different altitudes to test robustness to scale changes, another low altitude manual flight over houses was done to test robustness to 3D terrains. Testing *Lion* on these data is planned as a follow-on work at ESA-ESTEC. It can be related to previous absolute visual navigation UAV flight experiments by Trawny et al. (2007) and Conte and Doherty (2009).

Because of the limited processing capability of space computers, *Lion* was developed with the objective to make the computational cost as low as possible. The tight association between the online Harris feature extraction process and the a priori information from the filter through the scale management allows to solve the repeatability issue met by geometrical methods while being more efficient than standard scale-invariant operators which have to browse through the scale space (Mikolajczyk and Schmid, 2004; Lowe, 2004). A first step is to embed *Lion* in a space computer representative of a planetary exploration mission to evaluate the absolute image-to-map

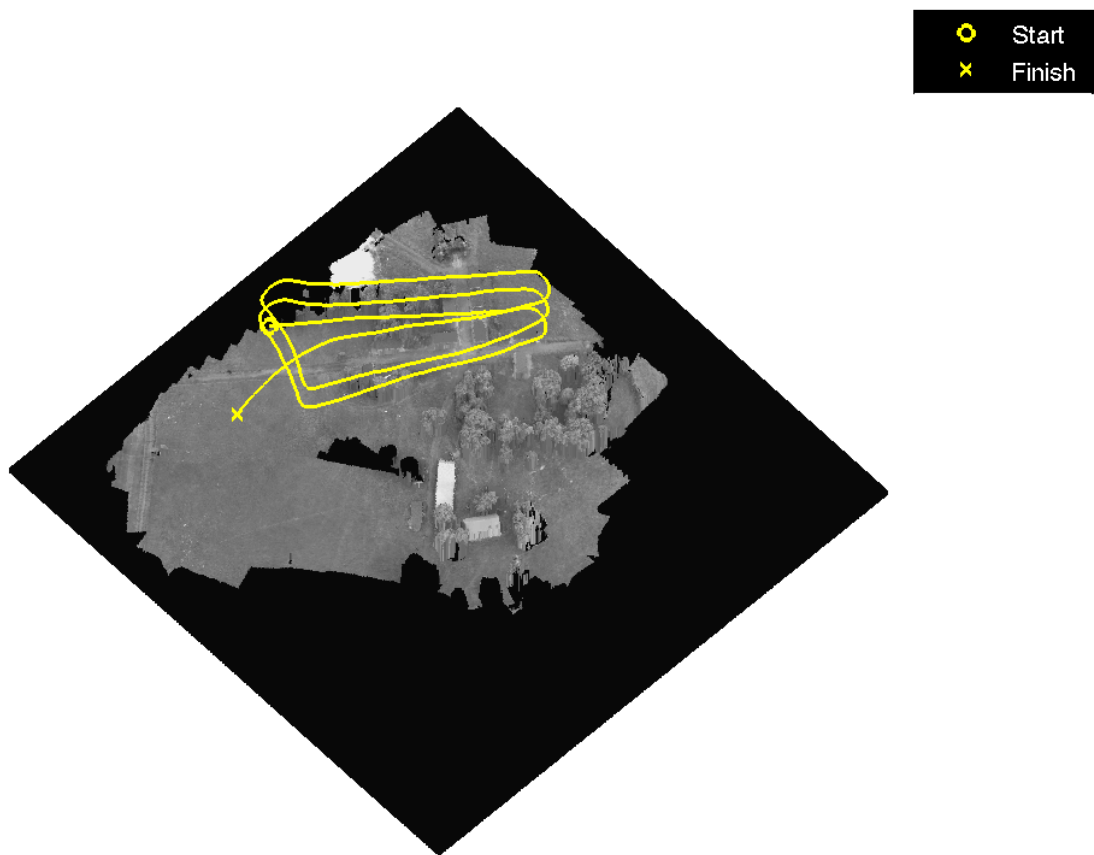


Figure 7.1 – Helicopter UAV test flight trajectory over the Caylus site near Toulouse

matching processing time. Next, the navigation accuracy tests of Chapter 6 should be reevaluated with the new absolute navigation update rate. This real-time version will also allow to trade off the navigation benefits of some extensions which could be brought to *Lion* with respect to additional computational time required.

7.2.2 Extending Lion

Now there are a few extensions which could be added to the current version of *Lion*, or simply compared to it.

The non-linearities of the inertial propagation and image measurement models described in Chapter 3 are processed through a linearization assumption by the EKF of *Lion*. We showed that, though simple, this assumption enables accuracy performances down to 2.2 m (3-RMS) at the end of the visual performance on a lunar landing trajectory with ideal image measurements, which is far enough for the 100-m accuracy requirement. The accuracy difference from 2.2 m to the 20.2 m figure (3-RMS) measured with actual image-to-map matching in Chapter 4 led us to focus on improving this latter step in order to bridge the gap from ideal to real landmark-matching. Though, comparing the ability of an Unscented Kalman Filter (UKF), or of a Particular Filter (PF), to converge with real measurements would be interesting and original in the future (Candy, 2009). Nevertheless, we suggest this comparison takes place once the real-time version of *Lion* is available in order to take of the extra computational cost of UKF and PF.

The Harris-Affine image features defined by Mikolajczyk and Schmid (2004) could be compared to Harris-Laplace for orbital landmarks. Although the authors' test seem to indicate a lesser robustness to scale change, the improved affine robustness could increase the performance under tilted descent camera angles happening at low altitudes during the approach phase. That would require to enhance the map building and online landmark reprojection processes with new information though.

We think that improving robustness to illumination changes between the reference map and the flight data is another interesting path of research too, especially for terrestrial applications of the algorithm. Unfortunately geometric descriptors are very sensitive to this, but one could think of building maps using images from several illumination angles, and only keeping stable and non-ambiguous features. As simpler alternative for illumination robustness for lunar landing, craters can be extracted like in Cheng and Ansar (2005) or Singh and Lim (2008) and processed in exactly the same way as Harris-Laplace landmarks by *Lion*, replacing the landmark image scale by the apparent crater radius in the orbital image.

From a more general standpoint about the filter, integrating relative visual navigation like Mourikis et al. (2009) would enable to limit the inertial drift at the end of the visual phase to increase touchdown accuracy. Nevertheless, our 53.8 m accuracy at touchdown demonstrated in Chapter 4 showed that it is not imperative with respect to the 100-m pinpoint landing requirement.

Finally, Kalman-based estimators are iterative processes which required an initial estimate with some accuracy. Although acquiring this initial estimation accuracy is not problematic on close planetary exploration targets like the Moon or Mars using Earth-based orbit estimation resources, it could be one for further destinations, or for terrestrial applications like urban UAV flight, where GPS is not accurate enough, or model-based augmented reality. The ground truth pose refinement algorithm detailed in Section 5.5 could actually be used for initialization. Affine-robust variations of SIFT can be used instead of SIFT for extreme viewpoint change (Morel and Yu, 2009), and

we had successful early tests of mutual information-base techniques for landmark matching under important illumination changes (Viola and Well, 1997).

A

ELEMENTS OF QUATERNIONS FOR ATTITUDE REPRESENTATION

CONTENTS

A.1	DEFINITION	125
A.2	MULTIPLICATION	126
A.3	RELATIONSHIP WITH ROTATION MATRICES	127
A.4	TIME DERIVATIVE	129

QUATERNIONS are commonly used to represent the orientation, or attitude, of one frame with respect to another. They avoid the gimbal lock singularity problem met by Euler angles while being more compact and more adapted to numerical simulations than rotation matrices. This appendix is an introduction to quaternions which derives the properties employed in the thesis. Trawny and Roumeliotis (2005) and Llibre (2009) provide a more complete description. Shuster (1993) surveys the various attitude representations.

Section A.1 and Section A.2 respectively define the quaternions and their multiplication. The relationship of quaternions with rotation matrices is derived in Section A.3, and their time derivative in Section A.4.

A.1 DEFINITION

A quaternion q is defined as

$$q = q_0 + q_1i + q_2j + q_3k \quad (\text{A.1})$$

where $q_0, q_1, q_2, q_3 \in \mathbb{R}$ and i, j, k are hypercomplex numbers such that

$$i^2 = j^2 = k^2 = ijk = -1. \quad (\text{A.2})$$

This is the standard Hamilton convention, from which we can derive all the products

$$ij = -ji = k \quad (\text{A.3})$$

$$jk = -kj = i \quad (\text{A.4})$$

$$ki = -ik = j. \quad (\text{A.5})$$

The set of quaternions is 4-dimensional euclidean vector space over the real numbers. Quaternions are usually represented as a vector $q \in \mathbb{R}^4$

$$q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T = \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix}, \quad (\text{A.6})$$

where q_0 is the real or scalar part, and \mathbf{q}_v is the imaginary or vector part. They inherit the addition and the scalar multiplication laws from \mathbb{R}^4 .

If a quaternion verifies

$$q = \begin{bmatrix} \cos(\theta/2) \\ u_x \sin(\theta/2) \\ u_y \sin(\theta/2) \\ u_z \sin(\theta/2) \end{bmatrix}, \quad (\text{A.7})$$

then q is called a quaternion of rotation and can model the rotation of angle θ along the unit vector axis $\mathbf{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T$. For a small rotation $\delta\theta$, the small angle approximation yields $\cos(\delta\theta/2) \simeq 1$ and $\sin(\delta\theta/2) \simeq \frac{\delta\theta}{2}$, thus posing $\delta\mathbf{q} = \delta\theta \mathbf{u}$ we have

$$\delta\mathbf{q} \simeq \begin{bmatrix} 1 & \frac{1}{2}\delta\theta^T \end{bmatrix}^T. \quad (\text{A.8})$$

A.2 MULTIPLICATION

The quaternion multiplication \otimes , also called the Hamilton product, is defined as

$$\mathbf{q} \otimes \mathbf{p} = (q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k})(p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}) \quad (\text{A.9})$$

$$\begin{aligned} \mathbf{q} \otimes \mathbf{p} = & q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3 + (q_1p_0 + q_0p_1 - q_3p_2 + q_2p_3)\mathbf{i} \\ & + (q_2p_0 + q_3p_1 + q_0p_2 - q_1p_3)\mathbf{j} + (q_3p_0 - q_2p_1 + q_1p_2 + q_0p_3)\mathbf{k} \end{aligned} \quad (\text{A.10})$$

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3 \\ q_1p_0 + q_0p_1 - q_3p_2 + q_2p_3 \\ q_2p_0 + q_3p_1 + q_0p_2 - q_1p_3 \\ q_3p_0 - q_2p_1 + q_1p_2 + q_0p_3 \end{bmatrix} \quad (\text{A.11})$$

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}. \quad (\text{A.12})$$

If we define the matrix notation for the cross product in \mathbb{R}^3 using the skew-symmetric matrix operator $[\mathbf{q}_v^\wedge]$ as

$$[\mathbf{q}_v^\wedge] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}, \quad (\text{A.13})$$

then we can write

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_0 & -\mathbf{q}_v^T \\ \mathbf{q}_v & q_0\mathbf{I}_3 + [\mathbf{q}_v^\wedge] \end{bmatrix} \begin{bmatrix} p_0 \\ \mathbf{p}_v \end{bmatrix} \quad (\text{A.14})$$

$$= \begin{bmatrix} q_0p_0 - \mathbf{q}_v^T \mathbf{p}_v \\ p_0\mathbf{q}_v + q_0\mathbf{p}_v + [\mathbf{q}_v^\wedge] \mathbf{p}_v \end{bmatrix} \quad (\text{A.15})$$

$$= \begin{bmatrix} p_0q_0 - \mathbf{p}_v^T \mathbf{q}_v \\ q_0\mathbf{p}_v + p_0\mathbf{q}_v - [\mathbf{p}_v^\wedge] \mathbf{q}_v \end{bmatrix} \quad (\text{A.16})$$

$$= \begin{bmatrix} p_0 & -\mathbf{p}_v^T \\ \mathbf{p}_v & p_0\mathbf{I}_3 - [\mathbf{p}_v^\wedge] \end{bmatrix} \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix}. \quad (\text{A.17})$$

The neutral quaternion \mathbf{q}_I with respect to multiplication is

$$\mathbf{q}_I = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \quad (\text{A.18})$$

and for any \mathbf{q} it verifies

$$\mathbf{q} \otimes \mathbf{q}_I = \mathbf{q}_I \otimes \mathbf{q} = \mathbf{q}. \quad (\text{A.19})$$

The inverse of a quaternion \mathbf{q} is the quaternion \mathbf{q}^{-1} such that

$$\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = \mathbf{q}_I. \quad (\text{A.20})$$

For quaternions of rotations, the inverse quaternion¹ is equal to the quaternion conjugate q^*

$$q^{-1} = q^* = \begin{bmatrix} q_0 \\ -\mathbf{q}_v \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ -\mathbf{u} \sin(\theta/2) \end{bmatrix} = \begin{bmatrix} \cos(-\theta/2) \\ \mathbf{u} \sin(-\theta/2) \end{bmatrix}. \quad (\text{A.21})$$

A.3 RELATIONSHIP WITH ROTATION MATRICES

Let \mathbf{p}_v be a vector of \mathbb{R}^3 and let \mathbf{p} be the associated pure quaternion

$$\mathbf{p} = \begin{bmatrix} 0 \\ \mathbf{p}_v \end{bmatrix}. \quad (\text{A.22})$$

If $\mathbf{p}_v^a, \mathbf{p}_v^b \in \mathbb{R}^3$ respectively represent the components of \mathbf{p}_v in two different frames $\{a\}$ and $\{b\}$, and if \mathbf{q}_a^b describes the rotation from $\{a\}$ to $\{b\}$, then

$$\mathbf{p}_v^b = \mathbf{C}(\mathbf{q}_a^b) \mathbf{p}_v^a \quad (\text{A.23})$$

where $\mathbf{C}(\mathbf{q}_a^b)$ is the coordinate change matrix from $\{a\}$ to $\{b\}$. It is orthogonal, thus $\mathbf{C}(\mathbf{q}_a^b)^{-1} = \mathbf{C}(\mathbf{q}_a^b)^T$. In terms of quaternions, as demonstrated in Llibre (2009), this is equivalent to

$$\begin{bmatrix} 0 \\ \mathbf{p}_v^b \end{bmatrix} = (\mathbf{q}_a^b)^{-1} \otimes \begin{bmatrix} 0 \\ \mathbf{p}_v^a \end{bmatrix} \otimes \mathbf{q}_a^b. \quad (\text{A.24})$$

Hence, using a third frame $\{c\}$, we can derive the relationship between the composition of quaternions and the compositions of rotations

$$\mathbf{C}(\mathbf{q}_a^c) = \mathbf{C}(\mathbf{q}_a^b \otimes \mathbf{q}_b^c) = \mathbf{C}(\mathbf{q}_b^c) \mathbf{C}(\mathbf{q}_a^b). \quad (\text{A.25})$$

Let us now develop Equation (A.24) using Equation (A.14). To make reading easier,

¹Which models the inverse rotation.

we note $\mathbf{q} = \mathbf{q}_a^b = \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix}$.

$$\begin{bmatrix} 0 \\ \mathbf{p}_v^b \end{bmatrix} = \begin{bmatrix} q_0 \\ -\mathbf{q}_v \end{bmatrix} \otimes \begin{bmatrix} 0 \\ \mathbf{p}_v^a \end{bmatrix} \otimes \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix} \quad (\text{A.26})$$

$$= \begin{bmatrix} q_0 & \mathbf{q}_v^T \\ -\mathbf{q}_v & q_0 \mathbf{I}_3 - [\mathbf{q}_v \wedge] \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{p}_v^a \end{bmatrix} \otimes \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix} \quad (\text{A.27})$$

$$= \begin{bmatrix} \mathbf{q}_v^T \mathbf{p}_v^a \\ q_0 \mathbf{p}_v^a - [\mathbf{q}_v \wedge] \mathbf{p}_v^a \end{bmatrix} \otimes \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix} \quad (\text{A.28})$$

$$= \begin{bmatrix} q_0 \mathbf{q}_v^T \mathbf{p}_v^a - q_0 \mathbf{q}_v^T \mathbf{p}_v^a - \mathbf{q}_v^T [\mathbf{q}_v \wedge] \mathbf{p}_v^a \\ q_0^2 \mathbf{p}_v^a - q_0 [\mathbf{q}_v \wedge] \mathbf{p}_v^a + \mathbf{q}_v \mathbf{q}_v^T \mathbf{p}_v^a + [q_0 \mathbf{p}_v^a - [\mathbf{q}_v \wedge] \mathbf{p}_v^a \wedge] \mathbf{q}_v \end{bmatrix} \quad (\text{A.29})$$

$$= \begin{bmatrix} -\mathbf{q}_v^T [\mathbf{q}_v \wedge] \mathbf{p}_v^a \\ q_0^2 \mathbf{p}_v^a - 2q_0 [\mathbf{q}_v \wedge] \mathbf{p}_v^a + \mathbf{q}_v \mathbf{q}_v^T \mathbf{p}_v^a - [[\mathbf{q}_v \wedge] \mathbf{p}_v^a \wedge] \mathbf{q}_v \end{bmatrix} \quad (\text{A.30})$$

$$= \begin{bmatrix} 0 \\ q_0^2 \mathbf{p}_v^a - 2q_0 [\mathbf{q}_v \wedge] \mathbf{p}_v^a + \mathbf{q}_v \mathbf{q}_v^T \mathbf{p}_v^a + [\mathbf{q}_v \wedge] [\mathbf{q}_v \wedge] \mathbf{p}_v^a \end{bmatrix} \quad (\text{A.31})$$

$$= \begin{bmatrix} 0 \\ q_0^2 \mathbf{p}_v^a - 2q_0 [\mathbf{q}_v \wedge] \mathbf{p}_v^a + \mathbf{q}_v \mathbf{q}_v^T \mathbf{p}_v^a + \mathbf{q}_v \mathbf{q}_v^T \mathbf{p}_v^a - \mathbf{p}_v^a \mathbf{q}_v^T \mathbf{q}_v \end{bmatrix} \quad (\text{A.32})$$

From Equation (A.7),

$$\|\mathbf{q}\|^2 = q_0^2 + \mathbf{q}_v^T \mathbf{q}_v \quad (\text{A.33})$$

$$= (\cos(\theta/2))^2 + (\sin(\theta/2))^2 (u_x^2 + u_y^2 + u_z^2) \quad (\text{A.34})$$

$$= (\cos(\theta/2))^2 + (\sin(\theta/2))^2 \quad (\text{A.35})$$

$$= 1. \quad (\text{A.36})$$

Thus we can write

$$\begin{bmatrix} 0 \\ \mathbf{p}_v^b \end{bmatrix} = \begin{bmatrix} 0 \\ q_0^2 \mathbf{p}_v^a - 2q_0 [\mathbf{q}_v \wedge] \mathbf{p}_v^a + 2\mathbf{q}_v \mathbf{q}_v^T \mathbf{p}_v^a - (1 - q_0^2) \mathbf{p}_v^a \end{bmatrix} \quad (\text{A.37})$$

$$= \begin{bmatrix} \mathbf{0}_{1 \times 3} \\ (2q_0^2 - 1) \mathbf{I}_3 - 2q_0 [\mathbf{q}_v \wedge] + 2\mathbf{q}_v \mathbf{q}_v^T \end{bmatrix} \mathbf{p}_v^a, \quad (\text{A.38})$$

and by identification

$$\mathbf{C}(\mathbf{q}) = (2q_0^2 - 1) \mathbf{I}_3 - 2q_0 [\mathbf{q}_v \wedge] + 2\mathbf{q}_v \mathbf{q}_v^T. \quad (\text{A.39})$$

In small angle approximation, $\delta \mathbf{q} \simeq \begin{bmatrix} 1 & \frac{1}{2} \delta \boldsymbol{\theta}^T \end{bmatrix}^T$ and at first order

$$\mathbf{C}(\delta \mathbf{q}) = \mathbf{I}_3 - [\delta \boldsymbol{\theta} \wedge]. \quad (\text{A.40})$$

As an extension to Equation (A.39), we can derive for any quaternion \mathbf{p}

$$\mathbf{q}^{-1} \otimes \mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} 1 & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{C}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} p_0 \\ \mathbf{p}_v \end{bmatrix}. \quad (\text{A.41})$$

A.4 TIME DERIVATIVE

Let $\mathbf{q}_a^{b(t)}$ be the quaternion which describes the rotation from a reference frame $\{a\}$ to a dynamic frame $\{b(t)\}$ changing through time. Its time derivative is defined as

$$\dot{\mathbf{q}}_a^{b(t)} = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left(\mathbf{q}_a^{b(t+\Delta t)} - \mathbf{q}_a^{b(t)} \right) \quad (\text{A.42})$$

$$= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left(\mathbf{q}_a^{b(t)} \otimes \mathbf{q}_{b(t)}^{b(t+\Delta t)} - \mathbf{q}_a^{b(t)} \otimes \mathbf{q}_1 \right), \quad (\text{A.43})$$

where the small angle approximation can be applied to $\mathbf{q}_{b(t)}^{b(t+\Delta t)}$

$$\mathbf{q}_{b(t)}^{b(t+\Delta t)} \simeq \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} \quad (\text{A.44})$$

$$\dot{\mathbf{q}}_a^{b(t)} \simeq \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left(\mathbf{q}_a^{b(t)} \otimes \left(\begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} - \begin{bmatrix} 1 \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \right) \right). \quad (\text{A.45})$$

Since by definition, $\lim_{\Delta t \rightarrow 0} \frac{\delta\boldsymbol{\theta}}{\Delta t} = \boldsymbol{\omega}_{ab(t)}^{b(t)}$ where $\boldsymbol{\omega}_{ab(t)}^{b(t)}$ is the angular velocity vector of frame $\{b(t)\}$ with respect to $\{a\}$ and projected in $\{b(t)\}$, we have

$$\dot{\mathbf{q}}_a^{b(t)} = \frac{1}{2} \mathbf{q}_a^{b(t)} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{ab(t)}^{b(t)} \end{bmatrix}. \quad (\text{A.46})$$

From Equation (A.17), this leads to

$$\dot{\mathbf{q}}_a^{b(t)} = \frac{1}{2} \begin{bmatrix} 0 & -\boldsymbol{\omega}_{ab(t)}^{b(t)T} \\ \boldsymbol{\omega}_{ab(t)}^{b(t)} & -[\boldsymbol{\omega}_{ab(t)}^{b(t)} \wedge] \end{bmatrix} \mathbf{q}_a^{b(t)} \quad (\text{A.47})$$

$$\dot{\mathbf{q}}_a^{b(t)} = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}_{ab(t)}^{b(t)}) \mathbf{q}_a^{b(t)}, \quad (\text{A.48})$$

where the operator $\boldsymbol{\Omega}$ is defined for all $\boldsymbol{\omega} \in \mathbb{R}^3$ by

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -[\boldsymbol{\omega} \wedge] \end{bmatrix}. \quad (\text{A.49})$$

B

NAVIGATION SYSTEMS

CONTENTS

B.1	NAVIGATION FRAMES	133
B.1.1	Inertial frame	133
B.1.2	Global frame	134
B.1.3	Body frame	134
B.1.4	Local frame	135
B.2	SENSOR INPUT	135
B.2.1	Inertial measurement unit	135
B.2.2	Star tracker	136
B.2.3	Terrain sensors	136
B.2.4	Summary	137
B.3	TERRAIN MAP DATA SOURCE	138
B.3.1	Orbital image	138
B.3.2	Digital elevation model	138
B.4	ERROR STATISTICS	138
B.4.1	Mean	139
B.4.2	Dispersion	139

THE standard GNC architecture is detailed in Figure B.1. The navigation system estimates the position, velocity and attitude of the spacecraft. These quantities form a state vector which is fed to the guidance and control systems to achieve a reference trajectory. Navigation uses measurement signals coming from sensors along with cartographic data stored in the on-board memory as inputs.

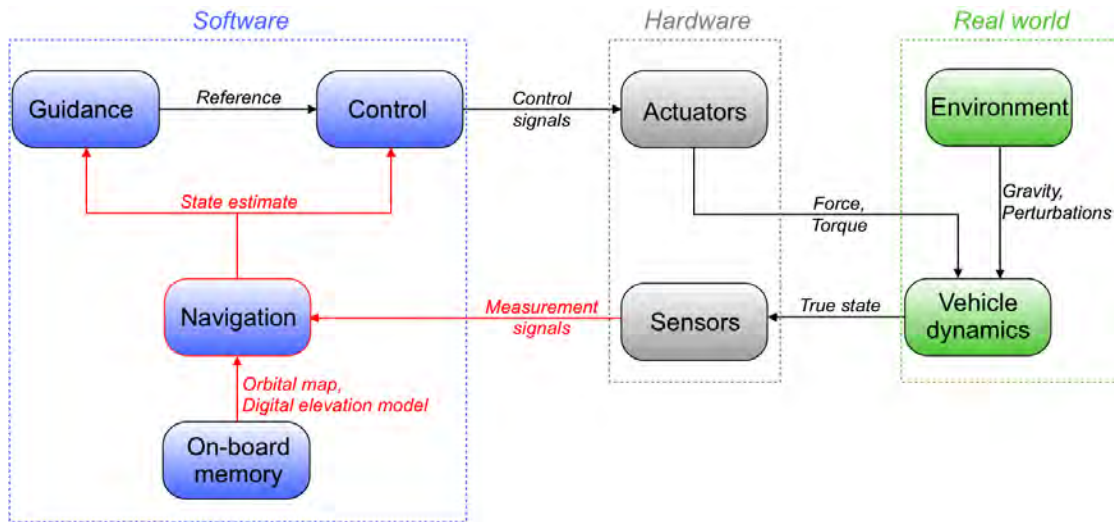


Figure B.1 – Detailed GNC architecture. The inputs and outputs of the navigation system are highlighted in red. Navigation computes an estimate of the state vector using measurement signals coming from the sensors and the cartographic data stored in the on-board memory.

Section B.1 defines the reference frames relevant for planetary landing. Sensor and cartographic data inputs are respectively presented in Section B.2 and Section B.3. Only the frames, states, sensors and cartographic data relevant to planetary landing are discussed here. Eventually, Section B.4 defines the statistical indicators used to characterize the navigation performance in Monte Carlo tests.

B.1 NAVIGATION FRAMES

Coordinate frames involved in navigation classically consist in an origin point and a set of three axes. The axis unit vectors form an orthogonal right-handed basis set. Navigation is a multiple coordinate frame problem in which the position and orientation of an object frame is determined with respect to a reference frame.

B.1.1 Inertial frame

An *inertial* frame is defined as one that does not accelerate or rotate with respect to the rest of the Universe. There is a multitude of them then. The inertial frame $\{i\}$ considered in this study is illustrated in Figure B.2 and is a planet-relative variant of the J_{2000} frame used for Earth-centered navigation problems. The origin at the centre of mass of the planet, and the axes are fixed with respect to distant stars. Technically speaking, this is not a true inertial frame since any planet is experiencing translational and rotational accelerations as it orbits around the Sun. However, the approximation is accurate over the timescale of a descent and landing trajectory.

The z_i axis points along the spin axis of the planet from the origin to the north pole. The x_i axis is defined as the direction from the origin to the Sun at spring equinox in

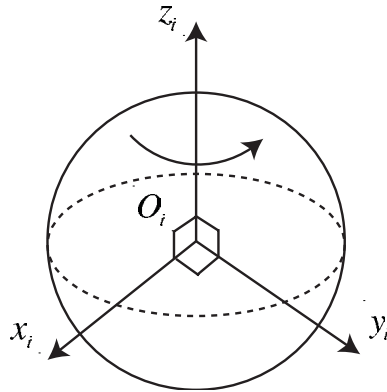


Figure B.2 – Inertial frame

year 2000, namely when the Sun goes from the southern to the northern hemisphere. The y_i axis completes the orthogonal right-handed set. The x_i and y_i axes lie in the equatorial plane and do not rotate with the planet.

B.1.2 Global frame

The *global* frame $\{g\}$ has the same origin and z axis as $\{i\}$. However, the x_g and y_g axes follow the planetary rotation. The x_g axis points from the origin to intersect the surface at the 0-degree longitude. The y_g axis completes the orthogonal right-handed set as shown in Figure B.3. This is not an inertial frame, but this is the one tied to the on-board maps. The landing site is initially defined by its coordinates in $\{g\}$, thus the navigation solution is also expressed in it.

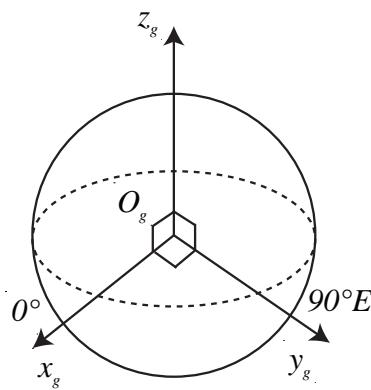


Figure B.3 – Global frame

B.1.3 Body frame

The *body* frame $\{b\}$ is tied to the lander vehicle and comprises its position and orientation information. The origin is the one point of the vehicle the navigation solution is sought for. The configuration used in our simulations is shown in Figure, the z_b axis points upwards along the thrust axis, namely opposite the engine thrust vector. The x_b axis is defined orthogonal to the z_b axis, such that the x_b - z_b plane contains the camera optical axis. The y_b axis completes the orthogonal right-handed set.

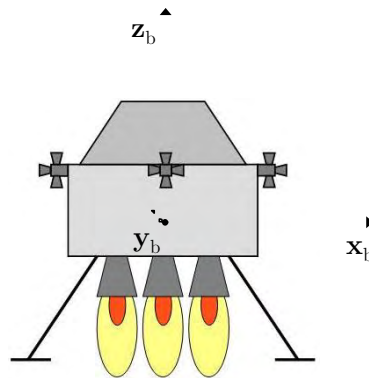


Figure B.4 – Body frame

B.1.4 Local frame

A local frame $\{l\}$ is any one fixed with respect to the terrain, with its origin above or on the surface. This type of frame is used in relative terrain navigation. Most often, it corresponds to the body frame at a previous time step with respect to which the pose of current body frame is determined, as illustrated in Figure B.5.

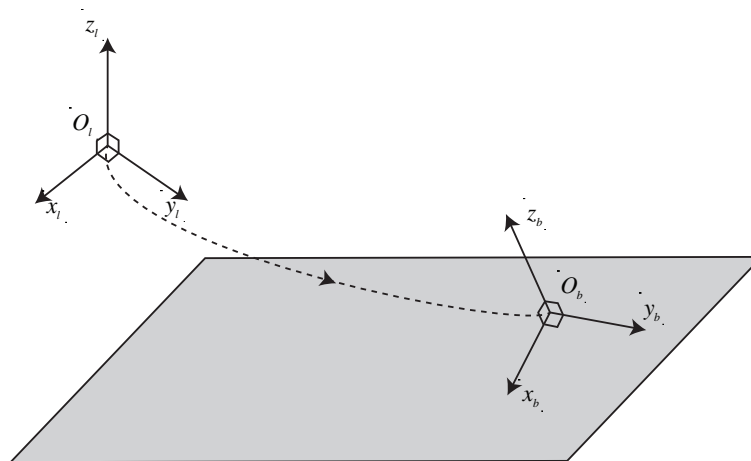


Figure B.5 – Body and local frames

B.2 SENSOR INPUT

Measurement signals coming from the on-board sensors are the only source of online information for an autonomous navigation system. This section reviews the various sensing devices which are mentioned in the planetary navigation literature.

B.2.1 Inertial measurement unit

Inertial sensors include any type of accelerometers or gyroscopes (gyros). An IMU is made of accelerometers and gyros. Three of each are deployed, one per axis, to produce 3D measurements of specific force $(f_{ib}^b)_m$ and angular rate $(\omega_{ib}^b)_m$ of the body frame with respect to the inertial frame $\{i\}$. The specific force measured by accelerometers is the non-gravitational acceleration of the body frame. Attitude, velocity, and

position estimates can be obtained by integrating the IMU signals at rates ranging from 100 to 1000 Hz. The reference IMU is an aerospace-class model by Northrop Grumman (2010). Its precision is characterized by an uncertainty of $300 \mu g$ (σ) over each accelerometer measurement and of 0.5 deg.hr^{-1} (σ) over the gyro measurements, σ is the standard deviation.

B.2.2 Star tracker

Stellar sensors measure the line-of-sight vectors of known stars to get an attitude fix $(q_i^b)_m$ with respect to the inertial axes. A strapdown star tracker, also known as a star imager, is a digital camera device using integrated software to match observed stars with those of an embedded catalogue. It can provide accurate attitude estimation in orbit or initialization for terrain-relative navigation using another attitude estimation system. Star trackers can deliver data at a rate from 1 to 20 Hz, with a pointing accuracy of the order of 0.01 deg (σ) (EADS Sodern, 2010).

B.2.3 Terrain sensors

Altimeter

Both radar and laser altimeters can be used for planetary landing applications. An electromagnetic signal is emitted downward and the height $(h)_m$ is computed by measuring the time elapsed for the reflected signal to be received. Radar altimeters work with radio signals, while laser devices make use of ultraviolet, visible or near-infrared frequencies. True altitude, with respect to a reference surface level, can be computed if an height database is stored on-board and if longitude and latitude estimates are available. Measurements are usually available from heights below 50 km at rates from 1 to 70 Hz and with an accuracy of 10 cm (σ) (Smith et al., 2010). In addition to height measurement, it should be noted that some radars can use the Doppler effect to provide ground-relative velocity measurements with an accuracy below 10 cm.s^{-1} (σ) (Pollard and Sadowy, 2005).

Lidar

The term LIDAR is acronym for LIght Detection And Ranging. Due to the smaller beamwidth of the laser range finders employed compared to radar devices, they can be used to build up a 3D profile below the vehicle. Such a profile is called a range image and it is the output a lidar device. There exist scanning and flash LIDARs.

A scanning lidar sends one tightly-collimated laser pulse at a time for each point in a grid spanning across the field of view. The range image is constructed by post-processing the data to take into account vehicle motion between each ranging measurement. A flash lidar uses only one laser pulse to build the whole range image. The pulse is optically diffused across the field of view and the time delay is measured in each direction by one of the pixel of the detector array. Flash devices are very interesting for EDL as nearly-instantaneous data collection makes post-processing much easier than in scanning. They can offer a 256×256 -pixel range image measurements at a rate of 30 Hz and with a range accuracy of 5 cm (σ) within 5.5 km of the target terrain (Weinberg et al., 2009).

Camera

Like commercial digital cameras, those used for planetary landing consist in optical lenses coupled with flight-qualified image sensors. They output intensity images which can be processed to detect surface features. Only cameras operating in the visible part of the spectrum are considered in the literature. As a passive device, cameras can operate at any distance and provide bearing-only information, with a ground resolution varying with altitude as shown in Figure B.6. They can acquire images at a rate from 1 to 100 Hz but suffer from radial distortions and a zero-mean Gaussian intensity noise with a standard deviation of 1 intensity level on an 8-bit image, thus leading a Signal-to-Noise Ratio (SNR) of about 250 (σ). The resolution of current European sensors available for planetary landing, notably the baseline for the ESA lunar lander mission, is 1024×1024 pixels, covering a field of view of 70 deg (Fischer et al., 2012; Flandin et al., 2009). For optimal navigation performances, one would want these two parameters to be as high as possible. Indeed, a higher sensor resolution allows for more precise direction determination of the imaged terrain features while a larger field of view makes the estimation of orientation and position along the optical axis more accurate. However, due to the limited bandwidth of current Flexible Programmable Gate Array (FPGA) boards used in space hardware and which have to transmit an image to the navigation system at a given frequency, the maximum sensor resolution is constrained. Likewise, the field of view is limited by the maximum optical distortion level tolerated in the output image.

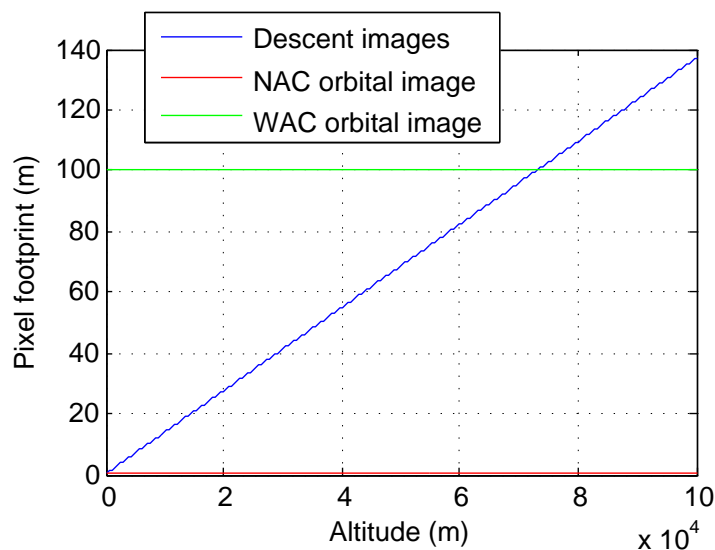


Figure B.6 – Pixel footprint of the descent camera images as a function of altitude during a lunar descent. In comparison, the green and the red horizontal lines show the footprint respectively for the Wide-Angle Camera (WAC) and the Narrow-Angle Camera (NAC) onboard the NASA-LRO spacecraft.

B.2.4 Summary

The type of measurement signals outputted by the different sensors reviewed is summarized in Table B.1.

Table B.1 – *Sensor signals and update rates*

Sensor	Measurement signal	Update rate (Hz)	Accuracy (σ)
IMU	Specific force (f_{ib}^b) _m	100-1000	300 μg
	Angular rate (ω_{ib}^b) _m	100-1000	0.5 deg.hr ⁻¹
Star tracker	Attitude quaternion (q_i^b) _m	1-20	0.01 deg
Altimeter	Height (h) _m	1-70	10 cm
Lidar	Range image	1-30	5 cm
Camera	Intensity image	10-100	SNR \simeq 250 (σ)

B.3 TERRAIN MAP DATA SOURCE

The only way to obtain pinpoint accuracy for the global state estimate \hat{x}_g in planetary landing is to identify terrain landmark from terrain sensors by matching them with reference maps stored in the on-board memory. These maps are usually created from two types of orbital data acquired prior to the landing mission: images and digital elevation models.

B.3.1 Orbital image

On planets considered as landing targets, visible image databases are available with resolution which can vary from a hundred meters to below one meter. Each image is referenced in terms of position and orientation of the camera with respect the global coordinate frame $\{g\}$. They can be used to detect descent landmarks for the landing camera.

B.3.2 Digital elevation model

The surface topography of a planet can be represented by a DEM. It is built from altimetry profile measurements or by stereo imaging. The information of a DEM can be put as a 3-row table: longitude, latitude, and elevation with respect to a reference level. Map quality is affected by horizontal and vertical resolution and by the terrain topography itself.

B.4 ERROR STATISTICS

The performance of a navigation system is characterized by the mean and the dispersion of the state estimation error probability distribution. Such systems are usually tested through Monte Carlo simulations, in which they are run an important number of times N while randomly sampling the uncertain parameters.

Let $x = \begin{bmatrix} x & y & z \end{bmatrix}^T$ be the real vector value of a 3-dimensional state which could be the position at some time instant of the trajectory for instance. The values of this state estimated by the navigation system throughout the N Monte Carlo simulations are included in Table B.2.

Table B.2 – Monte Carlo test estimates

x	y	z
\hat{x}_1	\hat{y}_1	\hat{z}_1
\vdots	\vdots	\vdots
\hat{x}_N	\hat{y}_N	\hat{z}_N

B.4.1 Mean

The mean navigation error may be expressed as a scalar value over each axis with

$$\begin{cases} \mu_x = \frac{1}{N} \sum_{i=1}^N (\hat{x}_i - x) \\ \mu_y = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y) \\ \mu_z = \frac{1}{N} \sum_{i=1}^N (\hat{z}_i - z) \end{cases}, \quad (\text{B.1})$$

or as a vector quantity with

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix}. \quad (\text{B.2})$$

The norm $\|\boldsymbol{\mu}\| = \sqrt{\mu_x^2 + \mu_y^2 + \mu_z^2}$ evaluates the magnitude of the mean error.

B.4.2 Dispersion

The statistical dispersion is a measure of how stretched the error distribution is with respect to the mean error. It is measured over one axis using the standard deviation

$$\begin{aligned} \sigma_x &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{x}_i - \mu_x)^2} \\ \sigma_y &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - \mu_y)^2} \\ \sigma_z &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{z}_i - \mu_z)^2} \end{aligned}. \quad (\text{B.3})$$

If a scalar random variable x follows a Gaussian distribution, then it verifies $P(\mu_x - 3\sigma_x < x < \mu_x + 3\sigma_x) \simeq 0.9973$. This 3σ envelope is often used to illustrate the uncertainty area of a variable, even if it is not Gaussian, but in this latter case it has no strict statistical meaning. The equivalent of the standard deviation for vector quantities is the covariance matrix

$$\begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{zy} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 \end{bmatrix}, \quad (\text{B.4})$$

where non-diagonal terms are correlation coefficients.

To get a quicker grasp of dispersion for vector quantities, the Root Mean Square (RMS) measure of the norm of the vector difference with respect to the mean provides a scalar indicator defined by

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{x}}_i - \boldsymbol{\mu}\|^2}, \quad (\text{B.5})$$

which can actually be related to the various standard deviations as

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N \left((\hat{x}_i - \mu_x)^2 + (\hat{y}_i - \mu_y)^2 + (\hat{z}_i - \mu_z)^2 \right)} \quad (\text{B.6})$$

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{x}_i - \mu_x)^2 + \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - \mu_y)^2 + \frac{1}{N} \sum_{i=1}^N (\hat{z}_i - \mu_z)^2} \quad (\text{B.7})$$

$$RMS = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}. \quad (\text{B.8})$$

Similarly to the 3σ envelope, the 3-RMS measure is used as a scalar indicator of the uncertainty area of the vector variable.

C

KALMAN OPTIMAL FILTERING AND DERIVATION OF *Lion* EKF EQUATIONS

CONTENTS

C.1	KALMAN AND EXTENDED KALMAN FILTERING	143
C.1.1	Continuous-time Kalman filter	143
C.1.2	Discrete-time Kalman filter	144
C.1.3	Extended Kalman filter	145
C.2	DERIVATION OF <i>Lion</i> EKF EQUATIONS	146
C.2.1	Linearized propagation model	146
C.2.2	Linearized measurement model	149
C.2.3	State augmentation	152

KALMAN optimal filtering aims at estimating recursively the state model of an uncertain linear model from a measurement perturbed by a noise. The estimate is optimal with respect to the sum of the variances of the state vector components, which it minimizes (Alazard, 2011). It is widely used in GNC systems for mobile vehicles, especially aircraft and spacecraft.

Section C.1 introduces the process and the equations of the Kalman and extended Kalman filters. The actual equations of the *Lion* filter are derived in Section C.2.

C.1 KALMAN AND EXTENDED KALMAN FILTERING

This section summarizes the general equations of the Kalman filter for linear systems, and its extension to nonlinear systems with the extended Kalman filter. The derivation of these equations can be found in Alazard (2011). Candy (2009) or Idier et al. (2004) provide alternative demonstrations for discrete-time systems. Another introduction to both Kalman and extended Kalman filtering can be found in Welch and Bishop (2006).

Nowadays, all calculators are digital so one may either choose to integrate the continuous-time equations numerically or to use directly the discrete-time equations. *Lion* actually does both as it uses a continuous-time propagation model with discrete-time updates.

C.1.1 Continuous-time Kalman filter

Let us consider a continuous-time linear state-space model

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{w}(t) \\ \mathbf{z}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{v}(t) \end{cases} \quad (\text{C.1})$$

where :

- $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector of the system,
- $\mathbf{w}(t) \in \mathbb{R}^q$ is the process noise which models the uncertainty of the system through the matrix $\mathbf{G} \in \mathbb{R}^{n \times q}$,
- $\mathbf{z}(t) \in \mathbb{R}^m$ is measurement vector,
- $\mathbf{v}(t) \in \mathbb{R}^m$ is the measurement noise.

\mathbf{F} , \mathbf{G} and \mathbf{H} matrices can be constant¹ or change through time. $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are centered Gaussian white noises with respective power spectral densities \mathbf{W} and \mathbf{V} .

Integrating the state equation leads to the general solution :

$$\mathbf{x}(t) = e^{\mathbf{F}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^t e^{\mathbf{F}(t-\tau)}\mathbf{G}\mathbf{w}(\tau) d\tau . \quad (\text{C.2})$$

¹Stationary case

Propagation

An initial Gaussian estimate of the state of mean $\hat{\mathbf{x}}(t_0)$ of covariance matrix $\mathbf{P}(t_0)$ can be propagated through time by applying the expectation operator to the state equation in the System (C.1) or to Equation (C.2) :

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}\hat{\mathbf{x}}(t) \quad (\text{C.3})$$

$$\hat{\mathbf{x}}(t) = e^{\mathbf{F}(t-t_0)}\hat{\mathbf{x}}(t_0) . \quad (\text{C.4})$$

The covariance matrix $\mathbf{P}(t_0)$ can be propagated using the linear differential equation

$$\dot{\mathbf{P}}(t) = \mathbf{F}\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T + \mathbf{G}\mathbf{W}\mathbf{G}^T . \quad (\text{C.5})$$

Update

The updated Kalman filter state estimate $\hat{\mathbf{x}}^+(t)$ can be defined as a new dynamic system with the measurement vector $\mathbf{z}(t)$ at input through a gain matrix $\mathbf{K}(t)$

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}'\hat{\mathbf{x}}(t) + \mathbf{K}(t)\mathbf{z}(t) . \quad (\text{C.6})$$

The estimate $\hat{\mathbf{x}}^+(t)$ can be shown to be unbiased if and only if $\mathbf{F}' = \mathbf{F} - \mathbf{K}(t)\mathbf{H}$, then

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}\hat{\mathbf{x}}(t) + \mathbf{K}(t)(\mathbf{z}(t) - \mathbf{H}\hat{\mathbf{x}}(t)) . \quad (\text{C.7})$$

The first term corresponds a propagation like in Equation (C.3) while the second term is equivalent to a correction : the so-called *update*. Among all the possible gain matrices, the Kalman one is derived as that minimizing the sum of the variances of the state, thus the trace of the covariance matrix $\mathbf{P}^+(t)$:

$$\mathbf{K}(t) = \mathbf{P}^+(t)\mathbf{H}^T\mathbf{V}^{-1} . \quad (\text{C.8})$$

The continuous-time Kalman covariance update follows the Riccati equation

$$\dot{\mathbf{P}}^+(t) = \mathbf{F}\mathbf{P}^+(t) + \mathbf{P}^+(t)\mathbf{F}^T - \mathbf{P}^+(t)\mathbf{H}^T\mathbf{V}^{-1}\mathbf{H}\mathbf{P}^+(t) + \mathbf{G}\mathbf{W}\mathbf{G}^T . \quad (\text{C.9})$$

C.1.2 Discrete-time Kalman filter

Likewise, a discrete-time linear state-space model can be defined by the stochastic difference and measurement equations

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{w}_k \\ \mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \end{cases} \quad (\text{C.10})$$

where \mathbf{w}_k and \mathbf{v}_k are centered Gaussian white noises with respective covariance matrices \mathbf{W} and \mathbf{V} .

Propagation

The propagation step of an initial Gaussian estimate of the state $\hat{\mathbf{x}}_0$ and of its covariance \mathbf{P}_0 now gives the equations :

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{F}\hat{\mathbf{x}}_k \quad (\text{C.11})$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}\mathbf{P}_k\mathbf{F}^T + \mathbf{G}\mathbf{W}\mathbf{G}^T \quad (\text{C.12})$$

Update

The updated Kalman filter state estimate $\hat{\mathbf{x}}_{k+1}^+$ can be obtained with

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1}(\mathbf{z}_{k+1} - \mathbf{H}\hat{\mathbf{x}}_{k+1}^-) \quad (\text{C.13})$$

using the discrete Kalman gain

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T + \mathbf{V})^{-1} . \quad (\text{C.14})$$

The discrete-time Kalman covariance is updated by

$$\mathbf{P}_{k+1}^+ = (\mathbf{I}_n - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}_{k+1}^- . \quad (\text{C.15})$$

C.1.3 Extended Kalman filter

Unfortunately, most real systems are not linear. Models like that of System (C.1) in continuous time, or that of System (C.10) in discrete time, are then not valid any more and the standard Kalman filtering framework discussed cannot be applied.

Let us consider the continuous-time nonlinear state-space model

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{w}(t)) \\ \mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{v}(t) \end{cases} . \quad (\text{C.16})$$

where \mathbf{f} and \mathbf{h} are two nonlinear functions of the state $\mathbf{x}(t) \in \mathbb{R}^n$. The process noise $\mathbf{w}(t) \in \mathbb{R}^q$ and the measurement noise $\mathbf{v}(t) \in \mathbb{R}^m$ are centered Gaussian white noises with respective power spectral densities \mathbf{W} and \mathbf{V} .

\mathbf{f} allows to propagate a state estimate, and \mathbf{h} to predict the measurement from it. Though, the error covariance matrix cannot be propagated like in Kalman, neither an optimal update.

The extended Kalman filter linearizes the System (C.16) about the current estimate $\hat{\mathbf{x}}(t)$ using multivariable Taylor series expansion

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{0}) + \mathbf{F}_J(\mathbf{x}(t) - \hat{\mathbf{x}}(t)) + \mathbf{G}_J\mathbf{w}(t) \\ \mathbf{z}(t) = \mathbf{h}(\hat{\mathbf{x}}(t)) + \mathbf{H}_J(\mathbf{x}(t) - \hat{\mathbf{x}}(t)) + \mathbf{v}(t) \end{cases} \quad (\text{C.17})$$

with the Jacobian matrices $\mathbf{F}_J = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t), \mathbf{w}=\mathbf{0}}$, $\mathbf{G}_J = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t), \mathbf{w}=\mathbf{0}}$ and $\mathbf{H}_J = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t)}$.

Let $\delta \mathbf{x} = \mathbf{x}(t) - \hat{\mathbf{x}}(t)$ be the estimation error and $\delta \mathbf{z} = \mathbf{z}(t) - \hat{\mathbf{z}}(t) = \mathbf{z}(t) - \mathbf{h}(\hat{\mathbf{x}}(t))$ the measurement prediction error. The System (C.17) then becomes

$$\begin{cases} \dot{\delta \mathbf{x}} = \mathbf{F}_J \delta \mathbf{x} + \mathbf{G}_J \mathbf{w}(t) \\ \delta \mathbf{z} = \mathbf{H}_J \delta \mathbf{x} + \mathbf{v}(t) \end{cases} . \quad (\text{C.18})$$

The estimation error $\delta \mathbf{x}$ thus follows an observable linear state-space model with centered Gaussian white noises, on which the standard Kalman framework of Subsection C.1.1 can be applied. The state estimate is not mathematically optimal due to the linearization, but the EKF empirically works well and provides near-optimal results when the system is not too nonlinear over the error range. This subsection only discussed the continuous-time case but the EKF can also be derived in discrete time.

C.2 DERIVATION OF *Lion* EKF EQUATIONS

This section demonstrates the equations used in the *Lion* EKF and discussed in Section 3.2. The notations are identical.

C.2.1 Linearized propagation model

Result

The state estimation error $\delta x_V = x_V - \hat{x}_V$ allows to linearize the EKF inertial propagation model with respect to the estimated state vector \hat{x}_V

$$\delta x_V = F_V \delta x_V + G_V n_{IMU}, \quad (3.9)$$

$$\text{where } F_V = \begin{bmatrix} -[\hat{\omega}^\wedge] & -I_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -C(\hat{q}_g^b)^T [\hat{a}^\wedge] & \mathbf{0}_3 & -2[\omega_{ig}^g \wedge] & -C(\hat{q}_g^b)^T & -[\omega_{ig}^g \wedge]^2 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & I_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \quad (3.10)$$

$$\text{and } G_V = \begin{bmatrix} -I_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & I_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -C(\hat{q}_g^b)^T & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & I_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}. \quad (3.11)$$

Proof

Each component of $\delta x_V = \left[\delta \theta_g^b{}^T \quad \delta \mathbf{b}_{gyr}{}^T \quad \delta \mathbf{v}_{gb}^g{}^T \quad \delta \mathbf{b}_{acc}{}^T \quad \delta \mathbf{p}_{gb}^g{}^T \right]^T$ can be expressed linearly as a function δx_V and n_{IMU} .

The biases of the gyroscopes are non-static and modeled by a random walk process, then

$$\delta \dot{\mathbf{b}}_{gyr} = \dot{\mathbf{b}}_{gyr} - \hat{\dot{\mathbf{b}}}_{gyr} \quad (C.19)$$

$$= \mathbf{n}_{bgyr} \quad (C.20)$$

$$= \mathbf{0}_{3 \times 15} \delta x_V + \begin{bmatrix} \mathbf{0}_3 & I_3 & \mathbf{0}_{3 \times 6} \end{bmatrix} n_{IMU}. \quad (C.21)$$

Likewise for the biases of the accelerometers:

$$\delta \dot{\mathbf{b}}_{acc} = \dot{\mathbf{b}}_{acc} - \hat{\dot{\mathbf{b}}}_{acc} \quad (C.22)$$

$$= \mathbf{n}_{bacc} \quad (C.23)$$

$$= \mathbf{0}_{3 \times 15} \delta x_V + \begin{bmatrix} \mathbf{0}_{3 \times 9} & I_3 \end{bmatrix} n_{IMU}. \quad (C.24)$$

By definition, the attitude quaternion can be written as

$$\mathbf{q}_g^b = \hat{\mathbf{q}}_g^b \otimes \delta \mathbf{q}_g^b \quad (C.25)$$

$$\dot{\mathbf{q}}_g^b = \dot{\hat{\mathbf{q}}}_g^b \otimes \delta \mathbf{q}_g^b + \hat{\mathbf{q}}_g^b \otimes \delta \dot{\mathbf{q}}_g^b. \quad (C.26)$$

From Equation (A.46), $\hat{q}_g^b = \frac{1}{2}q_g^b \otimes \begin{bmatrix} 0 \\ \omega_{gb}^b \end{bmatrix}$ thus

$$\frac{1}{2}q_g^b \otimes \begin{bmatrix} 0 \\ \omega_{gb}^b \end{bmatrix} = \frac{1}{2}\hat{q}_g^b \otimes \begin{bmatrix} 0 \\ \hat{\omega}_{gb}^b \end{bmatrix} \otimes \delta q_g^b + \hat{q}_g^b \otimes \delta \dot{q}_g^b \quad (\text{C.27})$$

$$\hat{q}_g^b \otimes \delta \dot{q}_g^b = \frac{1}{2} \left(q_g^b \otimes \begin{bmatrix} 0 \\ \omega_{gb}^b \end{bmatrix} - \hat{q}_g^b \otimes \begin{bmatrix} 0 \\ \hat{\omega}_{gb}^b \end{bmatrix} \otimes \delta q_g^b \right). \quad (\text{C.28})$$

A left multiplication by $(\hat{q}_g^b)^{-1}$ on each side yields

$$\delta \dot{q}_g^b = \frac{1}{2} \left(\delta q_g^b \otimes \begin{bmatrix} 0 \\ \omega_{gb}^b \end{bmatrix} - \begin{bmatrix} 0 \\ \hat{\omega}_{gb}^b \end{bmatrix} \otimes \delta q_g^b \right). \quad (\text{C.29})$$

The gyroscope measurement model of Equation (3.4) can be written

$$\omega_{gb}^b = \omega_{IMU} - C(q_g^b)\omega_{ig}^s - \mathbf{b}_{gyr} - \mathbf{n}_{gyr} \quad (\text{C.30})$$

$$\hat{\omega}_{gb}^b = \omega_{IMU} - C(\hat{q}_g^b)\omega_{ig}^s - \hat{\mathbf{b}}_{gyr}. \quad (\text{C.31})$$

Since $q_g^b = \hat{q}_g^b \otimes \delta q_g^b$, from Equation (A.25) we have $C(q_g^b) = C(\delta q_g^b)C(\hat{q}_g^b)$ and

$$\omega_{gb}^b = \omega_{IMU} - C(\delta q_g^b)C(\hat{q}_g^b)\omega_{ig}^s - \mathbf{b}_{gyr} - \mathbf{n}_{gyr}. \quad (\text{C.32})$$

Using small angle approximation $C(\delta q_g^b) = \mathbf{I}_3 - [\delta \theta_g^b \wedge]$ from Equation (A.40) gives

$$\omega_{gb}^b = \omega_{IMU} - C(\hat{q}_g^b)\omega_{ig}^s + [\delta \theta_g^b \wedge] C(\hat{q}_g^b)\omega_{ig}^s - \mathbf{b}_{gyr} - \mathbf{n}_{gyr} \quad (\text{C.33})$$

$$= \omega_{IMU} - C(\hat{q}_g^b)\omega_{ig}^s - [C(\hat{q}_g^b)\omega_{ig}^s \wedge] \delta \theta_g^b - \mathbf{b}_{gyr} - \mathbf{n}_{gyr} \quad (\text{C.34})$$

$$= \hat{\omega}_{gb}^b - \delta \mathbf{b}_{gyr} - \mathbf{n}_{gyr} - [C(\hat{q}_g^b)\omega_{ig}^s \wedge] \delta \theta_g^b. \quad (\text{C.35})$$

By substituting this result in Equation (C.29), we get

$$\begin{aligned} \delta \dot{q}_g^b &= \frac{1}{2} \left(\delta q_g^b \otimes \begin{bmatrix} 0 \\ \omega_{gb}^b \end{bmatrix} - \begin{bmatrix} 0 \\ \hat{\omega}_{gb}^b \end{bmatrix} \otimes \delta q_g^b \right) \\ &\quad - \frac{1}{2} \delta q_g^b \otimes \begin{bmatrix} 0 \\ \delta \mathbf{b}_{gyr} + \mathbf{n}_{gyr} + [C(\hat{q}_g^b)\omega_{ig}^s \wedge] \delta \theta_g^b \end{bmatrix} \end{aligned} \quad (\text{C.36})$$

$$\begin{aligned} \delta \dot{q}_g^b &= \frac{1}{2} \left(\delta q_g^b \otimes \begin{bmatrix} 0 \\ \omega_{gb}^b \end{bmatrix} - \begin{bmatrix} 0 \\ \hat{\omega}_{gb}^b \end{bmatrix} \otimes \delta q_g^b \right) \\ &\quad - \frac{1}{2} \delta q_g^b \otimes \begin{bmatrix} 0 \\ \delta \mathbf{b}_{gyr} + \mathbf{n}_{gyr} \end{bmatrix} + o(\delta \|x\|). \end{aligned} \quad (\text{C.37})$$

Writing the quaternion product as in Equation (A.14) and Equation (A.17) leads to

$$\begin{aligned} \dot{\delta q}_g^b &= \frac{1}{2} \left(\begin{bmatrix} 0 & -\hat{\omega}_{gb}^b{}^T \\ \hat{\omega}_{gb}^b & -[\hat{\omega}_{gb}^b \wedge] \end{bmatrix} \delta q_g^b - \begin{bmatrix} 0 & -\hat{\omega}_{gb}^b{}^T \\ \hat{\omega}_{gb}^b & [\hat{\omega}_{gb}^b \wedge] \end{bmatrix} \delta q_g^b \right) \\ &\quad - \frac{1}{2} \delta q_g^b \otimes \begin{bmatrix} 0 \\ \delta \mathbf{b}_{gyr} + \mathbf{n}_{gyr} \end{bmatrix} + o(\delta \| \mathbf{x} \|) \end{aligned} \quad (C.38)$$

$$\dot{\delta q}_g^b = \frac{1}{2} \begin{bmatrix} 0 & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & -2[\hat{\omega}_{gb}^b \wedge] \end{bmatrix} \delta q_g^b - \frac{1}{2} \delta q_g^b \otimes \begin{bmatrix} 0 \\ \delta \mathbf{b}_{gyr} + \mathbf{n}_{gyr} \end{bmatrix} + o(\delta \| \mathbf{x} \|) \quad (C.39)$$

$$\begin{aligned} \dot{\delta q}_g^b &= \frac{1}{2} \begin{bmatrix} 0 & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & -2[\hat{\omega}_{gb}^b \wedge] \end{bmatrix} \delta q_g^b - \frac{1}{2} \begin{bmatrix} 0 & -(\delta \mathbf{b}_{gyr} + \mathbf{n}_{gyr})^T \\ \delta \mathbf{b}_{gyr} + \mathbf{n}_{gyr} & -[\delta \mathbf{b}_{gyr} + \mathbf{n}_{gyr} \wedge] \end{bmatrix} \begin{bmatrix} 1 \\ \frac{1}{2} \delta \boldsymbol{\theta}_g^b \end{bmatrix} \\ &\quad + o(\delta \| \mathbf{x} \|). \end{aligned} \quad (C.40)$$

The gyro noise \mathbf{n}_{gyr} is equivalent to a first-order term, hence

$$\begin{aligned} \dot{\delta q}_g^b &= \frac{1}{2} \begin{bmatrix} 0 & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & -2[\hat{\omega}_{gb}^b \wedge] \end{bmatrix} \delta q_g^b - \frac{1}{2} \begin{bmatrix} 0 \\ \delta \mathbf{b}_{gyr} + \mathbf{n}_{gyr} \end{bmatrix} + o(\delta \| \mathbf{x} \|) \quad (C.41) \\ \begin{bmatrix} 0 \\ \frac{1}{2} \delta \boldsymbol{\theta}_g^b \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} 0 & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & -2[\hat{\omega}_{gb}^b \wedge] \end{bmatrix} \begin{bmatrix} 1 \\ \frac{1}{2} \delta \boldsymbol{\theta}_g^b \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 0 \\ \delta \mathbf{b}_{gyr} + \mathbf{n}_{gyr} \end{bmatrix} + o(\delta \| \mathbf{x} \|), \end{aligned} \quad (C.42)$$

which leads to the result by linearization:

$$\begin{aligned} \delta \boldsymbol{\theta}_g^b &= -[\hat{\omega}_{gb}^b \wedge] \delta \boldsymbol{\theta}_g^b - \delta \mathbf{b}_{gyr} - \mathbf{n}_{gyr} \\ &= \begin{bmatrix} -[\hat{\omega}_{gb}^b \wedge] & -\mathbf{I}_3 & \mathbf{0}_{3 \times 9} \end{bmatrix} \delta \mathbf{x}_V + \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3 \times 9} \end{bmatrix} \mathbf{n}_{IMU}. \end{aligned} \quad (C.43)$$

As of the error velocity vector, by definition

$$\delta \mathbf{v}_{gb}^g = \hat{\mathbf{v}}_{gb}^g - \hat{\mathbf{v}}_{gb}^g \quad (C.44)$$

$$\delta \mathbf{v}_{gb}^g = \hat{\mathbf{a}}_{gb}^g - \hat{\mathbf{a}}_{gb}^g \quad (C.45)$$

$$\begin{aligned} \delta \mathbf{v}_{gb}^g &= \left(\mathbf{C}(\hat{q}_g^b)^T (\mathbf{a}_{IMU} - \mathbf{b}_{acc} - \mathbf{n}_{acc}) - 2[\boldsymbol{\omega}_{ig}^g \wedge] \mathbf{v}_{gb}^g - [\boldsymbol{\omega}_{ig}^g \wedge]^2 \mathbf{p}_{gb}^g + \mathbf{g}_b^g \right) \\ &\quad - \left(\mathbf{C}(\hat{q}_g^b)^T (\mathbf{a}_{IMU} - \hat{\mathbf{b}}_{acc}) - 2[\boldsymbol{\omega}_{ig}^g \wedge] \hat{\mathbf{v}}_{gb}^g - [\boldsymbol{\omega}_{ig}^g \wedge]^2 \hat{\mathbf{p}}_{gb}^g + \mathbf{g}_b^g \right). \end{aligned} \quad (C.46)$$

Since $q_g^b = \hat{q}_g^b \otimes \delta q_g^b$, from Equation (A.25) we have $\mathbf{C}(q_g^b) = \mathbf{C}(\delta q_g^b) \mathbf{C}(\hat{q}_g^b)$ and

$$\begin{aligned} \delta \mathbf{v}_{gb}^g &= (\mathbf{C}(\hat{q}_g^b)^T \mathbf{C}(\delta q_g^b)^T (\mathbf{a}_{IMU} - \mathbf{b}_{acc} - \mathbf{n}_{acc}) - 2[\boldsymbol{\omega}_{ig}^g \wedge] \mathbf{v}_{gb}^g - [\boldsymbol{\omega}_{ig}^g \wedge]^2 \mathbf{p}_{gb}^g \\ &\quad - \mathbf{C}(\hat{q}_g^b)^T (\mathbf{a}_{IMU} - \hat{\mathbf{b}}_{acc}) + 2[\boldsymbol{\omega}_{ig}^g \wedge] \hat{\mathbf{v}}_{gb}^g + [\boldsymbol{\omega}_{ig}^g \wedge]^2 \hat{\mathbf{p}}_{gb}^g). \end{aligned} \quad (C.47)$$

Using small angle approximation $\mathbf{C}(\delta q_g^b) = \mathbf{I}_3 - [\delta \boldsymbol{\theta}_g^b \wedge]$ from Equation (A.40), and

$[\delta\boldsymbol{\theta}_g^b \wedge]^T = -[\delta\boldsymbol{\theta}_g^b \wedge]$, gives

$$\begin{aligned} \dot{\boldsymbol{v}}_{gb}^g &= \mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T (\mathbf{a}_{IMU} - \mathbf{b}_{acc} - \mathbf{n}_{acc}) + \mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T [\delta\boldsymbol{\theta}_g^b \wedge] (\mathbf{a}_{IMU} - \mathbf{b}_{acc} - \mathbf{n}_{acc}) \\ &\quad - 2 [\boldsymbol{\omega}_{ig}^g \wedge] \mathbf{v}_{gb}^g - [\boldsymbol{\omega}_{ig}^g \wedge]^2 \mathbf{p}_{gb}^g - \mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T (\mathbf{a}_{IMU} - \hat{\mathbf{b}}_{acc}) + 2 [\boldsymbol{\omega}_{ig}^g \wedge] \hat{\boldsymbol{v}}_{gb}^g \\ &\quad + [\boldsymbol{\omega}_{ig}^g \wedge]^2 \hat{\mathbf{p}}_{gb}^g \end{aligned} \quad (\text{C.48})$$

$$\begin{aligned} \delta\boldsymbol{v}_{gb}^g &= \mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T [\delta\boldsymbol{\theta}_g^b \wedge] (\mathbf{a}_{IMU} - \hat{\mathbf{b}}_{acc} - \delta\mathbf{b}_{acc} - \mathbf{n}_{acc}) \\ &\quad - 2 [\boldsymbol{\omega}_{ig}^g \wedge] \delta\boldsymbol{v}_{gb}^g - \mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T \delta\mathbf{b}_{acc} - [\boldsymbol{\omega}_{ig}^g \wedge]^2 \delta\mathbf{p}_{gb}^g - \mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T \mathbf{n}_{acc}. \end{aligned} \quad (\text{C.49})$$

The accelerometer noise \mathbf{n}_{acc} is equivalent to a first-order term, hence

$$\begin{aligned} \delta\boldsymbol{v}_{gb}^g &= -\mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T \left[(\mathbf{a}_{IMU} - \hat{\mathbf{b}}_{acc}) \wedge \right] \delta\boldsymbol{\theta}_g^b - 2 [\boldsymbol{\omega}_{ig}^g \wedge] \delta\boldsymbol{v}_{gb}^g - \mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T \delta\mathbf{b}_{acc} \\ &\quad - [\boldsymbol{\omega}_{ig}^g \wedge]^2 \delta\mathbf{p}_{gb}^g - \mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T \mathbf{n}_{acc} + o(\delta\|\mathbf{x}\|), \end{aligned} \quad (\text{C.50})$$

which leads to the result by linearization:

$$\begin{aligned} \delta\dot{\boldsymbol{v}}_{gb}^g &= \begin{bmatrix} -\mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T [\hat{\mathbf{a}} \wedge] & \mathbf{0}_3 & -2 [\boldsymbol{\omega}_{ig}^g \wedge] & -\mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T & -[\boldsymbol{\omega}_{ig}^g \wedge]^2 \end{bmatrix} \delta\mathbf{x}_V \\ &\quad + \begin{bmatrix} \mathbf{0}_{3 \times 6} & -\mathbf{C}(\hat{\boldsymbol{q}}_g^b)^T & \mathbf{0}_3 \end{bmatrix} \mathbf{n}_{IMU}. \end{aligned} \quad (\text{C.51})$$

And finally for the error position vector we get

$$\delta\dot{\mathbf{p}}_{gb}^g = \dot{\mathbf{p}}_{gb}^g - \hat{\dot{\mathbf{p}}}_{gb}^g \quad (\text{C.52})$$

$$= \mathbf{v}_{gb}^g - \hat{\boldsymbol{v}}_{gb}^g \quad (\text{C.53})$$

$$= \delta\boldsymbol{v}_{gb}^g \quad (\text{C.54})$$

$$= \begin{bmatrix} \mathbf{0}_{3 \times 6} & \mathbf{I}_3 & \mathbf{0}_{3 \times 6} \end{bmatrix} \delta\mathbf{x}_V + \mathbf{0}_{3 \times 12} \mathbf{n}_{IMU}. \quad (\text{C.55})$$

C.2.2 Linearized measurement model

Result

The innovation $\delta\mathbf{z}_j = \mathbf{z}_j - \hat{\mathbf{z}}_j$ can be linearized as

$$\delta\mathbf{z}_j \simeq \mathbf{H}_j \delta\mathbf{x} + \mathbf{n}_j = \mathbf{H}_{j,\theta} \delta\boldsymbol{\theta}_g^c + \mathbf{H}_{j,p} \delta\mathbf{p}_{gc}^g + \mathbf{n}_j, \quad (\text{3.18})$$

$$\text{with } \mathbf{H}_{j,\theta} = \frac{1}{z_{clj}^c} \begin{bmatrix} \mathbf{I}_2 & -\hat{\mathbf{z}}_j \end{bmatrix} \left[\mathbf{C}(\hat{\boldsymbol{q}}_g^c) (\mathbf{p}_{glj}^g - \hat{\mathbf{p}}_{gc}^g) \wedge \right], \quad (\text{3.19})$$

$$\mathbf{H}_{j,p} = \frac{1}{z_{clj}^c} \begin{bmatrix} \mathbf{I}_2 & -\hat{\mathbf{z}}_j \end{bmatrix} \mathbf{C}(\hat{\boldsymbol{q}}_g^c), \quad (\text{3.20})$$

$$\text{and } \mathbf{H}_j = \begin{bmatrix} \mathbf{0}_{2 \times 15} & \mathbf{H}_{j,\theta} & \mathbf{H}_{j,p} \end{bmatrix}. \quad (\text{3.21})$$

Proof

$$\delta z_j = z_j - \hat{z}_j = h_j(\mathbf{x}) - h_j(\hat{\mathbf{x}}) + \mathbf{n}_j \quad (\text{C.56})$$

From Equation (3.6) and Equation (3.17), we have

$$\delta z_j = \frac{1}{z_{cl_j}^c} \begin{bmatrix} x_{cl_j}^c \\ y_{cl_j}^c \\ z_{cl_j}^c \end{bmatrix} - \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} \hat{x}_{cl_j}^c \\ \hat{y}_{cl_j}^c \\ \hat{z}_{cl_j}^c \end{bmatrix} + \mathbf{n}_j. \quad (\text{C.57})$$

Since $\mathbf{p}_{cl_j}^c = \begin{bmatrix} x_{cl_j}^c & y_{cl_j}^c & z_{cl_j}^c \end{bmatrix}^T = \mathbf{C}(\mathbf{q}_g^c)(\mathbf{p}_{gl_j}^g - \mathbf{p}_{gc}^g)$, we can write

$$\begin{aligned} \delta z_j &= \frac{1}{z_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\mathbf{q}_g^c)(\mathbf{p}_{gl_j}^g - \mathbf{p}_{gc}^g) \\ &\quad - \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) + \mathbf{n}_j. \end{aligned} \quad (\text{C.58})$$

By definition $\mathbf{p}_{gc}^g = \hat{\mathbf{p}}_{gc}^g + \delta \mathbf{p}_{gc}^g$ and $\mathbf{q}_g^c = \hat{\mathbf{q}}_g^c \otimes \delta \mathbf{q}_g^c$, thus from Equation (A.25) $\mathbf{C}(\mathbf{q}_g^c) = \mathbf{C}(\delta \mathbf{q}_g^c) \mathbf{C}(\hat{\mathbf{q}}_g^c)$ and

$$\begin{aligned} \delta z_j &= \frac{1}{z_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\delta \mathbf{q}_g^c) \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g - \delta \mathbf{p}_{gc}^g) \\ &\quad - \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) + \mathbf{n}_j. \end{aligned} \quad (\text{C.59})$$

Using small angle approximation $\mathbf{C}(\delta \mathbf{q}_g^c) = \mathbf{I}_3 - [\delta \boldsymbol{\theta}_g^c \wedge]$ from Equation (A.40) yields

$$\begin{aligned} \delta z_j &= \frac{\hat{z}_{cl_j}^c}{z_{cl_j}^c \hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g - \delta \mathbf{p}_{gc}^g) \\ &\quad - \frac{\hat{z}_{cl_j}^c}{z_{cl_j}^c \hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} [\delta \boldsymbol{\theta}_g^c \wedge] \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g - \delta \mathbf{p}_{gc}^g) \\ &\quad - \frac{z_{cl_j}^c}{z_{cl_j}^c \hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) + \mathbf{n}_j, \end{aligned} \quad (\text{C.60})$$

and at the denominator, the depth of the landmark along the optical axis of the camera $z_{cl_j}^c$ turns into

$$z_{cl_j}^c = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{C}(\delta \mathbf{q}_g^c) \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g - \delta \mathbf{p}_{gc}^g) \quad (\text{C.61})$$

$$\begin{aligned} z_{cl_j}^c &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c) \delta \mathbf{p}_{gc}^g \\ &\quad - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} [\delta \boldsymbol{\theta}_g^c \wedge] \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g - \delta \mathbf{p}_{gc}^g) \end{aligned} \quad (\text{C.62})$$

$$z_{cl_j}^c = \hat{z}_{cl_j}^c - \mathbf{A} \delta \mathbf{p}_{gc}^g + \mathbf{B} \delta \boldsymbol{\theta}_g^c + o(\delta \|\mathbf{x}\|), \quad (\text{C.63})$$

with $\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c)$ and $\mathbf{B} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \left[\mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) \wedge \right]$. The innovation can thus be developed as

$$\begin{aligned} \delta \mathbf{z}_j &= \frac{z_{cl_j}^c + \mathbf{A} \delta \mathbf{p}_{gc}^g - \mathbf{B} \delta \boldsymbol{\theta}_g^c + o(\delta \|\mathbf{x}\|)}{z_{cl_j}^c \hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g - \delta \mathbf{p}_{gc}^g) \\ &+ \frac{z_{cl_j}^c + \mathbf{A} \delta \mathbf{p}_{gc}^g - \mathbf{B} \delta \boldsymbol{\theta}_g^c + o(\delta \|\mathbf{x}\|)}{z_{cl_j}^c \hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left[\mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g - \delta \mathbf{p}_{gc}^g) \wedge \right] \delta \boldsymbol{\theta}_g^c \\ &- \frac{z_{cl_j}^c}{z_{cl_j}^c \hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) + \mathbf{n}_j. \end{aligned} \quad (\text{C.64})$$

This can be simplified into

$$\begin{aligned} \delta \mathbf{z}_j &= \frac{\mathbf{A} \delta \mathbf{p}_{gc}^g - \mathbf{B} \delta \boldsymbol{\theta}_g^c}{z_{cl_j}^c \hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g - \delta \mathbf{p}_{gc}^g) \\ &- \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c) \delta \mathbf{p}_{gc}^g + \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left[\mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) \wedge \right] \delta \boldsymbol{\theta}_g^c \\ &+ \mathbf{n}_j + o(\delta \|\mathbf{x}\|) \end{aligned} \quad (\text{C.65})$$

$$\begin{aligned} \delta \mathbf{z}_j &= \frac{\mathbf{A} \delta \mathbf{p}_{gc}^g - \mathbf{B} \delta \boldsymbol{\theta}_g^c}{\hat{z}_{cl_j}^c - \mathbf{A} \delta \mathbf{p}_{gc}^g + \mathbf{B} \delta \boldsymbol{\theta}_g^c + o(\delta \|\mathbf{x}\|)} \hat{\mathbf{z}}_j - \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c) \delta \mathbf{p}_{gc}^g \\ &+ \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left[\mathbf{C}(\hat{\mathbf{q}}_g^c)(\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) \wedge \right] \delta \boldsymbol{\theta}_g^c + \mathbf{n}_j + o(\delta \|\mathbf{x}\|). \end{aligned} \quad (\text{C.66})$$

By applying the Taylor theorem to $f : x \rightarrow \frac{1}{1+x}$ in 0, we get

$$\frac{1}{\hat{z}_{cl_j}^c - \mathbf{A} \delta \mathbf{p}_{gc}^g + \mathbf{B} \delta \boldsymbol{\theta}_g^c + o(\delta \|\mathbf{x}\|)} = \frac{1}{\hat{z}_{cl_j}^c} \frac{1}{1 + \frac{1}{\hat{z}_{cl_j}^c} (-\mathbf{A} \delta \mathbf{p}_{gc}^g + \mathbf{B} \delta \boldsymbol{\theta}_g^c + o(\delta \|\mathbf{x}\|))} \quad (\text{C.67})$$

$$= \frac{1}{\hat{z}_{cl_j}^c} \left(1 - \frac{1}{\hat{z}_{cl_j}^c} (-\mathbf{A} \delta \mathbf{p}_{gc}^g + \mathbf{B} \delta \boldsymbol{\theta}_g^c) + o(\delta \|\mathbf{x}\|) \right). \quad (\text{C.68})$$

This can be reinjected into Equation (C.66) and gives

$$\begin{aligned} \delta z_j &= \frac{1}{\hat{z}_{cl_j}^c} \left(\hat{z}_j \mathbf{A} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c) \right) \delta \mathbf{p}_{gc}^g \\ &\quad - \frac{1}{\hat{z}_{cl_j}^c} \left(\hat{z}_j \mathbf{B} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left[\mathbf{C}(\hat{\mathbf{q}}_g^c) (\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) \wedge \right] \right) \delta \boldsymbol{\theta}_g^c + \mathbf{n}_j + o(\delta \|\mathbf{x}\|) \end{aligned} \quad (\text{C.69})$$

$$\begin{aligned} \delta z_j &= \frac{1}{\hat{z}_{cl_j}^c} \left(\hat{z}_j \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \right) \mathbf{C}(\hat{\mathbf{q}}_g^c) \delta \mathbf{p}_{gc}^g \\ &\quad - \frac{1}{\hat{z}_{cl_j}^c} \left(\hat{z}_j \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \right) \left[\mathbf{C}(\hat{\mathbf{q}}_g^c) (\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) \wedge \right] \delta \boldsymbol{\theta}_g^c \\ &\quad + \mathbf{n}_j + o(\delta \|\mathbf{x}\|) \end{aligned} \quad (\text{C.70})$$

$$\begin{aligned} \delta z_j &= -\frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} \mathbf{I}_2 & -\hat{z}_j \end{bmatrix} \mathbf{C}(\hat{\mathbf{q}}_g^c) \delta \mathbf{p}_{gc}^g \\ &\quad + \frac{1}{\hat{z}_{cl_j}^c} \begin{bmatrix} \mathbf{I}_2 & -\hat{z}_j \end{bmatrix} \left[\mathbf{C}(\hat{\mathbf{q}}_g^c) (\mathbf{p}_{gl_j}^g - \hat{\mathbf{p}}_{gc}^g) \wedge \right] \delta \boldsymbol{\theta}_g^c \\ &\quad + \mathbf{n}_j + o(\delta \|\mathbf{x}\|) \\ \delta z_j &= \mathbf{H}_{j,\theta} \delta \boldsymbol{\theta}_g^c + \mathbf{H}_{j,p} \delta \mathbf{p}_{gc}^g + \mathbf{n}_j + o(\delta \|\mathbf{x}\|) . \end{aligned} \quad (\text{C.71})$$

After linearization

$$\delta z_j \simeq \mathbf{H}_{j,\theta} \delta \boldsymbol{\theta}_g^c + \mathbf{H}_{j,p} \delta \mathbf{p}_{gc}^g + \mathbf{n}_j . \quad (3.18)$$

C.2.3 State augmentation

Result

The Jacobian matrix J allows to augment the error covariance matrix from $\mathbb{R}^{15 \times 15}$ to $\mathbb{R}^{21 \times 21}$

$$\mathbf{P}(t) \leftarrow \begin{bmatrix} \mathbf{I}_{15} \\ J \end{bmatrix} \mathbf{P}(t) \begin{bmatrix} \mathbf{I}_{15} \\ J \end{bmatrix}^T \quad (3.24)$$

$$J = \begin{bmatrix} \mathbf{C}(\mathbf{q}_b^c) & \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 3} \\ -\mathbf{C}(\mathbf{q}_g^b)^T [\mathbf{p}_{bc}^b \wedge] & \mathbf{0}_{3 \times 9} & \mathbf{I}_3 \end{bmatrix} . \quad (3.25)$$

Proof

Let us write $\delta \mathbf{x} \in \mathbb{R}^{15}$ the error state vector just before augmentation

$$\delta \mathbf{x} = \left[\delta \boldsymbol{\theta}_g^b{}^T \quad \delta \mathbf{b}_{gyr}{}^T \quad \delta \mathbf{v}_{gb}^g{}^T \quad \delta \mathbf{b}_{acc}{}^T \quad \delta \mathbf{p}_{gb}^g{}^T \right]^T , \quad (\text{C.72})$$

and $\delta \mathbf{x}' \in \mathbb{R}^{21}$ the augmented error state vector

$$\delta \mathbf{x}' = \left[\delta \boldsymbol{\theta}_g^b{}^T \quad \delta \mathbf{b}_{gyr}{}^T \quad \delta \mathbf{v}_{gb}^g{}^T \quad \delta \mathbf{b}_{acc}{}^T \quad \delta \mathbf{p}_{gb}^g{}^T \quad \delta \boldsymbol{\theta}_g^c{}^T \quad \delta \mathbf{p}_{gc}^g{}^T \right]^T . \quad (\text{C.73})$$

We seek the function g such that $\delta x' = g(\delta x)$. To do so, let us first determine the subfonctions g_θ and g_p such that respectively $\delta \theta_g^c = g_\theta(\delta x)$ and $\delta p_{gc}^g = g_p(\delta x)$.

$$q_g^c = q_g^b \otimes q_b^c \quad (C.74)$$

$$\hat{q}_g^c \otimes \delta q_g^c = \hat{q}_g^b \otimes \delta q_g^b \otimes q_b^c \quad (C.75)$$

$$\delta q_g^c = (\hat{q}_g^c)^{-1} \otimes \hat{q}_g^b \otimes \delta q_g^b \otimes q_b^c \quad (C.76)$$

$$= (q_b^c)^{-1} \otimes (\hat{q}_g^b)^{-1} \otimes \hat{q}_g^b \otimes \delta q_g^b \otimes q_b^c \quad (C.77)$$

$$= (q_b^c)^{-1} \otimes \delta q_g^b \otimes q_b^c \quad (C.78)$$

Using Equation (A.41), we can write

$$\delta q_g^c = \begin{bmatrix} 1 \\ \frac{1}{2} \delta \theta_g^c \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{C}(q_b^c) \end{bmatrix} \begin{bmatrix} 1 \\ \frac{1}{2} \delta \theta_g^b \end{bmatrix}. \quad (C.79)$$

Then, by identification:

$$\delta \theta_g^c = \mathbf{C}(q_b^c) \delta \theta_g^b \quad (C.80)$$

$$= \begin{bmatrix} \mathbf{C}(q_b^c) & \mathbf{0}_{3 \times 12} \end{bmatrix} \delta x \quad (C.81)$$

$$= g_\theta(\delta x). \quad (C.82)$$

Likewise for g_p :

$$\delta p_{gc}^g = p_{gc}^g - \hat{p}_{gc}^g \quad (C.83)$$

$$= p_{gb}^g + \mathbf{C}(q_g^b)^T p_{bc}^b - \hat{p}_{gb}^g - \mathbf{C}(\hat{q}_g^b)^T p_{bc}^b \quad (C.84)$$

$$= \delta p_{gb}^g + \left(\left[\mathbf{C}(\delta q_g^b) \mathbf{C}(\hat{q}_g^b) \right]^T - \mathbf{C}(\hat{q}_g^b)^T \right) p_{bc}^b \quad (C.85)$$

$$= \delta p_{gb}^g + \mathbf{C}(\hat{q}_g^b)^T \left(\mathbf{C}(\delta q_g^b)^T - \mathbf{I}_3 \right) p_{bc}^b. \quad (C.86)$$

Using small angle approximation $\mathbf{C}(\delta q_g^b) = \mathbf{I}_3 - [\delta \theta_g^b \wedge]$ from Equation (A.40) yields

$$\delta p_{gc}^g = \delta p_{gb}^g - \mathbf{C}(\hat{q}_g^b)^T [\delta \theta_g^b \wedge]^T p_{bc}^b \quad (C.87)$$

$$= \delta p_{gb}^g + \mathbf{C}(\hat{q}_g^b)^T [\delta \theta_g^b \wedge] p_{bc}^b \quad (C.88)$$

$$= \delta p_{gb}^g - \mathbf{C}(\hat{q}_g^b)^T [p_{bc}^b \wedge] \delta \theta_g^b \quad (C.89)$$

$$= \begin{bmatrix} -\mathbf{C}(\hat{q}_g^b)^T [p_{bc}^b \wedge] & \mathbf{0}_{3 \times 9} & \mathbf{I}_3 \end{bmatrix} \delta x \quad (C.90)$$

$$= g_p(\delta x). \quad (C.91)$$

Hence, the function g is linear and we have

$$\delta x' = \begin{bmatrix} \mathbf{I}_{15} \\ J \end{bmatrix} \delta x, \quad (C.92)$$

$$\text{with } J = \begin{bmatrix} \mathbf{C}(q_b^c) & \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 3} \\ -\mathbf{C}(q_g^b)^T [p_{bc}^b \wedge] & \mathbf{0}_{3 \times 9} & \mathbf{I}_3 \end{bmatrix}. \quad (C.93)$$

Let us note $\mathbf{P}(t) \in \mathbb{R}^{15 \times 15}$ the covariance matrix just before augmentation and $\mathbf{P}'(t) \in \mathbb{R}^{21 \times 21}$ the augmented one.

$$\mathbf{P}'(t) = E \left[\delta \mathbf{x}'(t) \delta \mathbf{x}'(t)^T \right] \quad (\text{C.94})$$

$$= E \left[\begin{bmatrix} \mathbf{I}_{15} \\ \mathbf{J} \end{bmatrix} \delta \mathbf{x}(t) \delta \mathbf{x}(t)^T \begin{bmatrix} \mathbf{I}_{15} \\ \mathbf{J} \end{bmatrix}^T \right] \quad (\text{C.95})$$

$$= \begin{bmatrix} \mathbf{I}_{15} \\ \mathbf{J} \end{bmatrix} E \left[\delta \mathbf{x}(t) \delta \mathbf{x}(t)^T \right] \begin{bmatrix} \mathbf{I}_{15} \\ \mathbf{J} \end{bmatrix}^T \quad (\text{C.96})$$

$$= \begin{bmatrix} \mathbf{I}_{15} \\ \mathbf{J} \end{bmatrix} \mathbf{P}(t) \begin{bmatrix} \mathbf{I}_{15} \\ \mathbf{J} \end{bmatrix}^T \quad (\text{C.97})$$

D

VISILAB FRAMES

CONTENTS

D.1	MOCK-UP	157
D.2	CALIBRATION PATTERN	157
D.3	CAMERA CALIBRATION	157
D.4	WORKING CAMERA	158

CAMERA true pose determination is required in order to acquire workable images for navigation testing in *Visilab*. This process was discussed in Subsection 5.3.2 and involves a series of four frames for which the origin and the orientation of the axes are defined in this appendix. The horizontal plane and the vertical axis are respectively defined by the mean plane of the optical table on which the camera support platform is set up, and by its normal axis.

D.1 MOCK-UP

The mock-up frame $\{r\}$ is the reference navigation frame in *Visilab* in which the DEM is defined. It is equivalent to the lunar surface-fixed frame $\{g\}$ employed in planetary landing.

- *Origin* O_r : on the bottom right corner of the mock-up when facing it, on the back panel.
- x_r axis: vertical upwards.
- y_r axis: horizontal leftwards.
- z_r axis: horizontal towards the front of the mock-up.

D.2 CALIBRATION PATTERN

A calibration pattern is stuck on an aluminium plate located at the left of the mock-up when facing it and screwed on the same longitudinal bars the mock-up is attached to. The associated frame $\{p\}$ is used as an intermediary one to compute the pose of the camera with respect to the mock-up.

- *Origin* O_p : on the front surface of the aluminum plate, the exact position depends on the square corner chosen for optical pose computation.
- x_p axis: vertical upwards.
- y_p axis: horizontal leftwards.
- z_p axis: horizontal towards the front of the mock-up.

D.3 CAMERA CALIBRATION

The calibration pose and its frame $\{c\}$ were defined to compute the orientation of the camera with respect to the mock-up, and to enable the computation of the final working position. The camera shall be oriented with the desired attitude as close as possible to the calibration pattern so that it appears as big as possible in the image. The wider the pattern in the image, the more accurate the camera calibration pose estimation.

- *Origin* O_c : optical center of the camera in calibration pose.
- x_c axis: line axis of the array on the sensor, towards the right when facing the observed scene.
- y_c axis: completes the direct trihedron.
- z_c axis: along the optical axis, towards the observed scene.

D.4 WORKING CAMERA

The working pose of the camera at frame $\{w\}$ is the one in which the final image used by the navigation system is acquired, and for which ground truth is required. The position is computed from the known displacement between from the calibration pose while the orientation must remain the same.

- *Origin O_w* : optical center of the camera in working pose.
- *x_w axis*: line axis of the array on the sensor, towards the right when facing the observed scene.
- *y_w axis*: completed the direct trihedron.
- *z_w axis*: along the optical axis, towards the observed scene.

E

HARDWARE PERFORMANCES TABLES

CONTENTS

E.1	ACCURACY EVALUATION	161
E.1.1	Nominal descent accuracy	161
E.1.2	Contribution of Visilab environmental error	162
E.2	ROBUSTNESS EVALUATION	163
E.2.1	Camera sensor change	163
E.2.2	Off-nadir descent camera inclination	163
E.2.3	Illumination change	164
E.2.4	Non-flat terrain	166

DETAILED result tables for the hardware tests of *Lion* performed in *Visilab* and discussed in Chapter 6 are included in this appendix. The reference navigation frame for all these tests is the mock-up frame $\{r\}$ in *Visilab*, defined in Appendix D. As explained in Subsection 6.1.2, the ground truth trajectory of the body frame $\{b\}$ was interpolated between the refined ground truth camera working pose estimates $\{w\}$.

Section E.1 contains the results of the navigation accuracy tests over the dynamic scaled lunar landing sequence while Section E.2 deals with the robustness tests over the static sequences.

E.1 ACCURACY EVALUATION

E.1.1 Nominal descent accuracy

Table E.1 – *Lion* navigation performance during lunar coasting phase with the low resolution map. The errors are represented by a couple $(\mu, 3\sigma)$. μ is the mean error. σ denotes the dispersion either as the standard deviation over one axis or as the RMS value for the total error.

Lunar altitude		100 km	50 km	15 km
Initial errors	$\delta\theta_{r,i}^b$ (per axis, deg)	(0, 0.5)	(0, 0.5)	(0, 0.5)
	Total $\delta\theta_r^b$ (deg)	(0, 0.9)	(0, 0.9)	(0, 0.9)
	$\delta v_{r,b,i}^r$ (per axis, m.s ⁻¹)	(0, 5)	(0, 5)	(0, 5)
	Total δv_{rb}^r (m.s ⁻¹)	(0, 8.7)	(0, 8.7)	(0, 8.7)
	$\delta p_{r,b,i}^r$ (per axis, m)	(0, 5000)	(0, 2500)	(0, 750)
	Total δp_{rb}^r (m)	(0, 8660)	(0, 4330)	(0, 1299)
Final errors	$\delta\theta_{r,x}^b$ (deg)	(0.04, 0.4)	(-0.1, 0.6)	(0.2, 0.4)
	$\delta\theta_{r,y}^b$ (deg)	(0.03, 0.3)	(0.1, 0.3)	(0.01, 0.4)
	$\delta\theta_{r,z}^b$ (deg)	(-0.03, 0.04)	(-0.02, 0.04)	(0.1, 0.1)
	Total $\delta\theta_r^b$ (deg)	(0.1, 0.5)	(0.1, 0.7)	(0.1, 0.6)
	$\delta v_{r,b,x}^r$ (crossrange, m.s ⁻¹)	(3.7, 5.7)	(3.3, 5.5)	(0.6, 5.1)
	$\delta v_{r,b,y}^r$ (downrange, m.s ⁻¹)	(3.9, 5.2)	(9.5, 5.6)	(0.3, 4.6)
	$\delta v_{r,b,z}^r$ (height, m.s ⁻¹)	(-0.002, 7.0)	(0.9, 5.9)	(0.6, 4.2)
	Total δv_{rb}^r (m.s ⁻¹)	(5.4, 10.4)	(10.1, 9.8)	(0.9, 8.0)
	$\delta p_{r,b,x}^r$ (crossrange, m)	(128.9, 634.3)	(57.8, 228.6)	(74.2, 131.8)
	$\delta p_{r,b,y}^r$ (downrange, m)	(-26.3, 712.2)	(35.3, 193.6)	(7.6, 104.5)
	$\delta p_{r,b,z}^r$ (height, m)	(-45.1, 224.0)	(54.4, 136.4)	(-0.5, 52.7)
	Total δp_{rb}^r (m)	(139.1, 979.6)	(86.8, 329.2)	(74.6, 176.2)

Table E.2 – *Lion navigation performance during braking phase with the Low Resolution (LR) and High Resolution (HR) maps. The errors are represented by a couple $(\mu, 3\sigma)$. μ is the mean error. σ denotes the dispersion either as the standard deviation over one axis or as the RMS value for the total error.*

Lunar altitude		10 km LR	10 km HR	5-3 km LR	5-3 km HR
Initial errors	$\delta\theta_{r,i}^b$ (per axis, deg)	(0, 0.5)	(0, 0.5)	(0, 0.5)	(0, 0.5)
	Total $\delta\theta_r^b$ (deg)	(0, 0.9)	(0, 0.9)	(0, 0.9)	(0, 0.9)
	$\delta v_{r,i}^r$ (per axis, m.s ⁻¹)	(0, 5)	(0, 5)	(0, 5)	(0, 5)
	Total δv_{rb}^r (m.s ⁻¹)	(0, 8.7)	(0, 8.7)	(0, 8.7)	(0, 8.7)
	$\delta p_{r,i}^r$ (per axis, m)	(0, 500)	(0, 500)	(0, 250)	(0, 250)
	Total δp_{rb}^r (m)	(0, 866)	(0, 866)	(0, 433)	(0, 433)
Final errors	$\delta\theta_{r,x}^b$ (deg)	(0.1, 0.7)	(-0.005, 0.4)	(0.4, 1.4)	(-0.03, 0.5)
	$\delta\theta_{r,y}^b$ (deg)	(-0.1, 1.0)	(0.01, 0.3)	(-0.4, 1.0)	(0.1, 0.4)
	$\delta\theta_{r,z}^b$ (deg)	(-0.03, 0.2)	(-0.01, 0.1)	(-0.8, 0.6)	(-0.0006, 0.1)
	Total $\delta\theta_r^b$ (deg)	(0.2, 1.3)	(0.01, 0.5)	(0.9, 1.8)	(0.1, 0.6)
	$\delta v_{rb,x}^r$ (crossrange, m.s ⁻¹)	(-0.7, 4.0)	(-0.4, 2.6)	(-0.6, 2.8)	(-0.2, 1.4)
	$\delta v_{rb,y}^r$ (downrange, m.s ⁻¹)	(-1.6, 3.0)	(-1.6, 3.6)	(-0.3, 3.9)	(-0.4, 1.0)
	$\delta v_{rb,z}^r$ (height, m.s ⁻¹)	(0.04, 4.6)	(0.04, 3.4)	(-0.2, 4.0)	(-0.1, 2.0)
	Total δv_{rb}^r (m.s ⁻¹)	(1.8, 6.8)	(1.6, 5.7)	(0.7, 6.2)	(0.5, 2.6)
	$\delta p_{rb,x}^r$ (crossrange, m)	(45.4, 200.6)	(1.6, 67.5)	(61.4, 82.2)	(3.9, 29.5)
	$\delta p_{rb,y}^r$ (downrange, m)	(31.0, 146.2)	(-16.1, 88.7)	(60.6, 102.6)	(0.2, 32.1)
	$\delta p_{rb,z}^r$ (height, m)	(-13.4, 85.7)	(-0.05, 43.0)	(10.3, 89.3)	(-0.6, 18.0)
	Total δp_{rb}^r (m)	(56.6, 262.6)	(16.2, 119.4)	(86.9, 158.9)	(4.0, 47.2)

E.1.2 Contribution of Visilab environmental error

Table E.3 – *Contribution of Visilab environmental error to 3 σ final position dispersion. σ denotes the dispersion either as the standard deviation over one axis or as the RMS value for the total error.*

Lunar altitude		100 km - LR	50 km - LR	15 km - LR	10 km - HR	5-3 km - HR
Total errors	$\delta p_{rb,x}^r$ (crossrange, m)	634.3	228.6	131.8	67.5	29.5
	$\delta p_{rb,y}^r$ (downrange, m)	712.2	193.6	104.5	88.7	32.1
	$\delta p_{rb,z}^r$ (height, m)	224.0	136.4	52.7	43.0	18.0
	Total δp_{rb}^r (m)	979.6	329.2	176.2	119.4	47.2
Environmental errors	$\delta p_{rb,x}^r$ (crossrange, m)	163.9	163.9	33.6	33.6	11.7
	$\delta p_{rb,y}^r$ (downrange, m)	140.9	140.9	28.9	28.9	10.0
	$\delta p_{rb,z}^r$ (height, m)	163.6	163.6	33.5	33.5	11.6
	Total δp_{rb}^r (m)	271.1	271.1	55.6	55.6	19.3

E.2 ROBUSTNESS EVALUATION

E.2.1 Camera sensor change

Table E.4 – *Lion navigation robustness to camera sensor change (Visilab/Moon scale $\approx 1.1 \times 10^{-5}$). The errors are represented by a couple $(\mu, 3\sigma)$. μ is the mean error. σ denotes the dispersion either as the standard deviation over one axis or as the RMS value for the total error.*

Lunar altitude		20 km	43 km	67 km	
Initial errors	$\delta p_{rb,i}^r$ (per axis, m)	(0, 1000)	(0, 2150)	(0, 3350)	
	Total δp_{rb}^r (m)	(0, 1732)	(0, 3724)	(0, 5802)	
Final errors	Regular map	$\delta p_{rb,x}^r$ (crossrange, m)	(-2.6, 48.3)	(11.9, 86.2)	(8.1, 175.9)
		$\delta p_{rb,y}^r$ (downrange, m)	(-5.7, 91.1)	(0.9, 146.1)	(-10.6, 211.7)
		$\delta p_{rb,z}^r$ (height, m)	(12.9, 99.4)	(-3.3, 112.9)	(-79.6, 344.1)
		Total δp_{rb}^r (m)	(14.3, 143.3)	(75.0, 203.7)	(75.8, 440.6)
	Virtual-image map	$\delta p_{rb,x}^r$ (crossrange, m)	(2.2, 49.9)	(12.2, 103.9)	(34.7, 205.5)
		$\delta p_{rb,y}^r$ (downrange, m)	(-7.0, 36.6)	(19.5, 108.1)	(9.8, 199.0)
		$\delta p_{rb,z}^r$ (height, m)	(8.5, 53.6)	(53.5, 90.5)	(-148.7, 217.0)
		Total δp_{rb}^r (m)	(11.2, 81.8)	(109.2, 175.2)	(115.5, 359.0)

E.2.2 Off-nadir descent camera inclination

Table E.5 – *Lion navigation robustness to off-nadir descent camera inclination (Visilab/Moon scale $\approx 1.1 \times 10^{-5}$). The errors are represented by a couple $(\mu, 3\sigma)$. μ is the mean error. σ denotes the dispersion either as the standard deviation over one axis or as the RMS value for the total error.*

Lunar altitude h		h < 40 km	40 km < h < 70 km	70 km < h	
20-deg off nadir	Initial errors	$\delta p_{rb,i}^r$ (per axis, m)	(0, 1500)	(0, 2750)	(0, 4000)
		Total δp_{rb}^r (m)	(0, 2598)	(0, 4763)	(0, 6928)
	Final errors	$\delta p_{rb,x}^r$ (crossrange, m)	(-1.0, 47.4)	(11.3, 92.4)	(-2.2, 155.4)
		$\delta p_{rb,y}^r$ (downrange, m)	(-4.8, 58.3)	(-6.9, 130.4)	(-31.2, 215.2)
		$\delta p_{rb,z}^r$ (height, m)	(-11.9, 40.6)	(-52.8, 45.3)	(150.9, 163.8)
		Total δp_{rb}^r (m)	(12.9, 85.4)	(82.0, 166.1)	(84.2, 311.9)
40-deg off nadir	Initial errors	$\delta p_{rb,i}^r$ (per axis, m)	(0, 600)	(0, 1100)	(0, 1600)
		Total δp_{rb}^r (m)	(0, 1039)	(0, 1905)	(0, 2771)
	Final errors	$\delta p_{rb,x}^r$ (crossrange, m)	(35.4, 107.2)	(18.0, 144.1)	(17.8, 207.1)
		$\delta p_{rb,y}^r$ (downrange, m)	(-6.6, 112.6)	(45.6, 233.5)	(-3.1, 235.7)
		$\delta p_{rb,z}^r$ (height, m)	(2.4, 113.4)	(45.6, 334.9)	(15.4, 379.7)
		Total δp_{rb}^r (m)	(36.1, 192.5)	(47.0, 432.9)	(67.5, 492.5)
60-deg off nadir	Initial errors	$\delta p_{rb,i}^r$ (per axis, m)	(0, 600)	(0, 1100)	(0, 1600)
		Total δp_{rb}^r (m)	(0, 1039)	(0, 1905)	(0, 2771)
	Final errors	$\delta p_{rb,x}^r$ (crossrange, m)	(22.9, 103.8)	(30.9, 197.1)	(3.1, 211.9)
		$\delta p_{rb,y}^r$ (downrange, m)	(3.6, 173.6)	(34.5, 225.6)	(-38.7, 515.3)
		$\delta p_{rb,z}^r$ (height, m)	(-10.8, 121.9)	(56.2, 213.9)	(-21.6, 823.6)
		Total δp_{rb}^r (m)	(25.6, 236.2)	(124.7, 368.2)	(39.3, 994.3)

E.2.3 Illumination change

Table E.6 – *Lion navigation robustness to light azimuth change (Visilab/Moon scale $\approx 1.1 \times 10^{-5}$). The errors are represented by a couple $(\mu, 3\sigma)$. μ is the mean error. σ denotes the dispersion either as the standard deviation over one axis or as the RMS value for the total error.*

Lunar altitude			20 km	43 km	67 km
Initial errors		$\delta p_{rb,i}^r$ (per axis, m)	(0, 1000)	(0, 2150)	(0, 3350)
		Total $\delta \mathbf{p}_{rb}^r$ (m)	(0, 1732)	(0, 3724)	(0, 5802)
Final errors	Reference illumination	$\delta p_{rb,x}^r$ (crossrange, m)	(2.2, 49.9)	(12.2, 103.9)	(34.7, 205.5)
		$\delta p_{rb,y}^r$ (downrange, m)	(-7.0, 36.6)	(19.5, 108.1)	(9.8, 199.0)
		$\delta p_{rb,z}^r$ (height, m)	(8.5, 53.6)	(53.5, 90.5)	(-148.7, 217.0)
		Total $\delta \mathbf{p}_{rb}^r$ (m)	(11.2, 81.8)	(109.2, 175.2)	(115.5, 359.0)
	20-deg azimuth change	$\delta p_{rb,x}^r$ (crossrange, m)	(-3.1, 90.0)	(-32.8, 660.0)	(-171.7, 1568.4)
		$\delta p_{rb,y}^r$ (downrange, m)	(-14.3, 81.7)	(2.3, 193.2)	(49.8, 675.7)
		$\delta p_{rb,z}^r$ (height, m)	(13.3, 96.0)	(-101.5, 295.1)	(-224.6, 508.6)
		Total $\delta \mathbf{p}_{rb}^r$ (m)	(19.8, 154.9)	(183.1, 748.3)	(252.5, 1781.9)
	45-deg azimuth change	$\delta p_{rb,x}^r$ (crossrange, m)	(11.1, 304.8)	(-29.7, 1512.1)	(184.2, 2977.2)
		$\delta p_{rb,y}^r$ (downrange, m)	(-53.7, 344.4)	(-109.6, 860.0)	(-86.0, 1728.4)
		$\delta p_{rb,z}^r$ (height, m)	(61.6, 469.5)	(106.4, 2074.1)	(-112.3, 2673.1)
		Total $\delta \mathbf{p}_{rb}^r$ (m)	(82.5, 657.2)	(391.1, 2707.0)	(467.8, 4358.5)
80-deg azimuth change	$\delta p_{rb,x}^r$ (crossrange, m)	(-61.0, 246.5)	(15.5, 1405.5)	(-142.1, 1631.7)	
	$\delta p_{rb,y}^r$ (downrange, m)	(14.5, 417.4)	(109.8, 1145.7)	(86.6, 1395.0)	
	$\delta p_{rb,z}^r$ (height, m)	(97.9, 474.9)	(-27.1, 1107.8)	(-234.2, 2241.9)	
	Total $\delta \mathbf{p}_{rb}^r$ (m)	(116.3, 678.6)	(175.0, 2124.9)	(327.3, 3103.9)	
180-deg azimuth change	$\delta p_{rb,x}^r$ (crossrange, m)	(-26.3, 155.9)	(-7.5, 199.4)	(-68.5, 648.2)	
	$\delta p_{rb,y}^r$ (downrange, m)	(1.4, 200.3)	(23.2, 129.8)	(-78.1, 1094.4)	
	$\delta p_{rb,z}^r$ (height, m)	(2.5, 416.9)	(-34.1, 262.2)	(-252.2, 1409.3)	
	Total $\delta \mathbf{p}_{rb}^r$ (m)	(26.4, 488.1)	(334.2, 354.1)	(374.2, 1898.4)	

Table E.7 – *Lion navigation robustness to light elevation change (Visilab/Moon scale $\approx 1.1 \times 10^{-5}$). The errors are represented by a couple $(\mu, 3\sigma)$. μ is the mean error. σ denotes the dispersion either as the standard deviation over one axis or as the RMS value for the total error.*

Lunar altitude		20 km	43 km	67 km	
Initial errors	$\delta p_{rb,i}^r$ (per axis, m)	(0, 1000)	(0, 2150)	(0, 3350)	
	Total δp_{rb}^r (m)	(0, 1732)	(0, 3724)	(0, 5802)	
Final errors	Reference illumination	$\delta p_{rb,x}^r$ (crossrange, m)	(2.2, 49.9)	(12.2, 103.9)	(34.7, 205.5)
		$\delta p_{rb,y}^r$ (downrange, m)	(-7.0, 36.6)	(19.5, 108.1)	(9.8, 199.0)
		$\delta p_{rb,z}^r$ (height, m)	(8.5, 53.6)	(53.5, 90.5)	(-148.7, 217.0)
		Total δp_{rb}^r (m)	(11.2, 81.8)	(109.2, 175.2)	(115.5, 359.0)
	-20-deg elevation change	$\delta p_{rb,x}^r$ (crossrange, m)	(-19.1, 588.1)	(-104.4, 617.1)	(-61.6, 3361.6)
		$\delta p_{rb,y}^r$ (downrange, m)	(-6.6, 360.6)	(62.2, 715.7)	(54.2, 2685.2)
		$\delta p_{rb,z}^r$ (height, m)	(126.8, 1110.5)	(-91.4, 884.7)	(23.0, 1814.5)
		Total δp_{rb}^r (m)	(128.4, 1307.3)	(184.7, 1294.6)	(176.7, 4669.4)
	-10-deg elevation change	$\delta p_{rb,x}^r$ (crossrange, m)	(-7.1, 58.8)	(-1.7, 111.9)	(7.4, 218.2)
		$\delta p_{rb,y}^r$ (downrange, m)	(-21.2, 69.3)	(-3.7, 109.6)	(-59.1, 245.5)
		$\delta p_{rb,z}^r$ (height, m)	(5.3, 88.4)	(-63.2, 148.6)	(-125.7, 483.0)
		Total δp_{rb}^r (m)	(22.9, 126.8)	(181.5, 215.9)	(186.5, 584.1)
+10-deg azimuth change	$\delta p_{rb,x}^r$ (crossrange, m)	(-9.1, 150.2)	(-6.0, 115.8)	(-33.2, 226.3)	
	$\delta p_{rb,y}^r$ (downrange, m)	(-12.0, 61.7)	(37.5, 102.9)	(6.0, 231.6)	
	$\delta p_{rb,z}^r$ (height, m)	(-0.2, 250.0)	(-53.5, 120.8)	(-238.1, 204.2)	
	Total δp_{rb}^r (m)	(15.1, 298.0)	(176.6, 196.5)	(187.6, 382.8)	
+20-deg azimuth change	$\delta p_{rb,x}^r$ (crossrange, m)	(31.4, 503.1)	(101.7, 1135.9)	(9.02, 1179.4)	
	$\delta p_{rb,y}^r$ (downrange, m)	(85.7, 727.2)	(16.4, 148.5)	(51.1, 2013.8)	
	$\delta p_{rb,z}^r$ (height, m)	(17.6, 826.8)	(106.1, 2223.3)	(-115.5, 1835.4)	
	Total δp_{rb}^r (m)	(92.9, 1210.6)	(594.5, 2501.1)	(229.7, 2969.0)	

E.2.4 Non-flat terrain

Table E.8 – *Lion navigation robustness to non-flat terrain (Visilab/Moon scale $\approx 3.3 \times 10^{-6}$). The errors are represented by a couple $(\mu, 3\sigma)$. μ is the mean error. σ denotes the dispersion either as the standard deviation over one axis or as the RMS value for the total error.*

Lunar altitude			50 km	100 km	200 km
Initial errors		$\delta p_{rb,i}^r$ (per axis, m)	(0, 2500)	(0, 5000)	(0, 10000)
		Total δp_{rb}^r (m)	(0, 4330)	(0, 8660)	(0, 17321)
Final errors	Flat-terrain assumption	$\delta p_{rb,x}^r$ (crossrange, m)	(8.6, 166.7)	(8.5, 269.7)	(182.0, 610.5)
		$\delta p_{rb,y}^r$ (downrange, m)	(-46.3, 200.1)	(-81.8, 269.7)	(-690.6, 747.0)
		$\delta p_{rb,z}^r$ (height, m)	(1310.8, 630.9)	(1376.2, 556.0)	(9334.0, 3201.9)
		Total δp_{rb}^r (m)	(1311.7, 682.8)	(1378.7, 674.2)	(9361.3, 3344.1)
	No terrain assumption	$\delta p_{rb,x}^r$ (crossrange, m)	(-17.7, 134.5)	(16.9, 250.7)	(72.0, 354.6)
		$\delta p_{rb,y}^r$ (downrange, m)	(14.6, 114.7)	(11.4, 237.3)	(84.2, 479.7)
		$\delta p_{rb,z}^r$ (height, m)	(-70.6, 44.6)	(10.8, 36.8)	(-11.6, 215.5)
		Total δp_{rb}^r (m)	(74.2, 182.2)	(23.1, 347.1)	(111.4, 634.3)

REFERENCES

- Adams, D., Criss, T. B., and Shankar, U. J. (2008). Passive Optical Terrain Relative Navigation Using APLNav. In *2008 IEEE Aerospace Conference*, Big Sky, MT. (Cited at pages 19, 21, 22, 24, 28, and 29.)
- Alazard, D. (2011). *Introduction to Kalman filtering*. ISAE, Toulouse, France. (Cited at page 143.)
- Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359. (Cited at pages 20, 21, and 22.)
- Belongie, S., Mori, G., and Malik, J. (2000). Matching with Shape Contexts. In *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*, Hilton Head Island, SC. (Cited at pages 21 and 50.)
- Bonin-Font, F., Ortiz, A., and Oliver, G. (2008). Visual Navigation for Mobile Robots: A Survey. *Journal of Intelligent and Robotic Systems*, 53(3):263–296. (Cited at page 17.)
- Braun, R. D. and Manning, R. M. (2006). Mars Exploration Entry, Descent and Landing Challenges. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT. (Cited at page 9.)
- Caballero, F., Merino, L., Ferruz, J., and Ollero, a. (2008). Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs. *Journal of Intelligent and Robotic Systems*, 54(1-3):137–161. (Cited at page 26.)
- Candy, J. V. (2009). *Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods*. (Cited at pages 24, 120, and 143.)
- Cheatham, D. C. and Bennett, F. V. (1968). Apollo Lunar Module Landing Strategy. Technical report, NASA. (Cited at page 9.)
- Cheng, Y. and Ansar, A. (2005). A Landmark Based Position Estimation for Pinpoint Landing on Mars. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain. (Cited at pages 12, 20, 21, 22, 29, and 120.)
- Chin, G., Brylow, S., Foote, M., Garvin, J., Kasper, J., Keller, J., Litvak, M., Mitrofanov, I., Paige, D., Raney, K., Robinson, M., Sanin, A., Smith, D., Spence, H., Spudis, P., Stern, S. A., and Zuber, M. (2007). Lunar Reconnaissance Orbiter Overview: The Instrument Suite and Mission. *Space Science Reviews*, 129(4):391–419. (Cited at page 101.)
- Conte, G. and Doherty, P. (2009). Vision-Based Unmanned Aerial Vehicle Navigation Using Geo-Referenced Information. *EURASIP Journal on Advances in Signal Processing*, pages 1–18. (Cited at pages 12, 19, 21, 24, 29, 58, and 118.)

- Davison, A. J. (2003). Real-Time Simultaneous Localisation and Mapping with a Single. In *9th International Conference on Computer Vision*, Nice, France. (Cited at page 25.)
- de Lafontaine, J., Neveu, D., and Hamel, J.-F. (2008). Autonomous Planetary Landing using a LIDAR Sensor: the Landing Dynamic Test Facility. In *7th International ESA Conference on Guidance, Navigation & Control Systems*, Tralee, Ireland. (Cited at page 29.)
- Delaune, J., De Rosa, D., and Hobbs, S. (2010). Guidance and Control system design for Lunar Descent and Landing. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Toronto, Canada. (Cited at page 38.)
- Delaune, J., Le Besnerais, G., Farges, J.-L., Bourdarias, C., Voirin, T., and Piquereau, A. (2011). Tightly-Integrated Visual and Inertial Navigation for Pinpoint Landing on Rugged Terrains. In *11th Symposium on Advanced Space Technologies in Robotics and Automation*, Noordwijk, The Netherlands. (Cited at page 53.)
- Duclos, D., Quinquis, N., Broda, G., Galmiche, F., Oudyi, F., Coulon, N., Cordier, D., and Sonier, C. (2009). A presentation of ATR processing chain validation procedure of IR terminal guidance version of the AASM modular air-to-ground weapon. In *Proceedings of SPIE: Automatic Target Recognition XIX*, Orlando, FL. (Cited at page 51.)
- EADS Sodern, . (2010). SED26 Star Tracker Data Sheet. Technical report. (Cited at page 136.)
- Eric, W. and Grimson, L. (1991). *Object Recognition by Computer: The Role of Geometric Constraints*. (Cited at page 51.)
- Fisackerly, R., Houdou, B., Philippe, C., De Rosa, D., Carpenter, J., Pradier, A., and Gardini, B. (2012). The European Lunar Lander : A Human Exploration Precursor Mission. In *Proceedings of the Global Space Exploration Conference*, Washington, DC. (Cited at page 10.)
- Fischer, D., Zaunick, E., Polle, B., and Kervendal, E. (2012). NPAL Evolutions applied to ESA's Lunar Lander Mission. In *Proceedings of the 35th Annual Guidance and Control Conference*, Breckenridge, CO. (Cited at page 137.)
- Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus : A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395. (Cited at pages 49, 52, and 95.)
- Flandin, G., Polle, B., Frapard, B., Vidal, P., Philippe, C., and Voirin, T. (2009). Vision-Based Navigation for Planetary Exploration. In *Proceedings of the 32nd Annual AAS Rocky Mountain Guidance and Control Conference*, Breckenridge, Colorado. (Cited at pages 25, 26, and 137.)
- Goyo Optical Inc., . (2011). GM23514MCN 3.5-mm lens data sheet. Technical report. (Cited at page 75.)
- Goyo Optical Inc., . (2013). GM24514MCN 4.5-mm lens data sheet. Technical report. (Cited at page 75.)
- Groves, P. D. (2008). *Principles of GNSS , Inertial , and Multisensor Integrated Navigation Systems*. (Cited at page 22.)

- Gura, I. A. and Gersten, R. H. (1970). On Analysis of n-Dimensional Normal Probabilities. Technical report, The Aerospace Corporation. (Cited at page 65.)
- Hamel, J.-F., Neveu, D., and De Lafontaine, J. (2006). Feature Matching Navigation Techniques for Lidar-Based Planetary Exploration. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado. (Cited at page 12.)
- Harris, C. and Stephens, M. (1988). A Combined Corner and Edge Detector. In *4th Alvey Vision Conference*, Manchester, United Kingdom. (Cited at pages 20 and 61.)
- Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in computer vision*. (Cited at pages 19, 49, 52, 53, 58, 87, 91, and 95.)
- Idier, J., Hélène, P.-L., Le Besnerais, G., and Champagnat, F. (2004). *Traitement numérique du signal. Deuxième partie : Algorithmes*. ENSTA, Paris, France. (Cited at page 143.)
- Imaging Development Systems, I. (2012). UI-1240SE camera data sheet. Technical report. (Cited at page 75.)
- Janschek, K., Tchernykh, V., and Beck, M. (2006). Performance Analysis for Visual Planetary Landing Navigation Using Optical Flow and DEM Matching. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Keystone, Colorado. (Cited at pages 19, 21, 25, 28, and 29.)
- Johnson, A., Willson, R., Cheng, Y., Goguen, J., Leger, C., Sanmartin, M., and Matthies, L. (2007). Design Through Operation of an Image-Based Velocity Estimation System for Mars Landing. *International Journal of Computer Vision*, 74(3):319–341. (Cited at page 13.)
- Johnson, A. E., Huertas, A., Werner, R. a., and Montgomery, J. F. (2008). Analysis of On-Board Hazard Detection and Avoidance for Safe Lunar Landing. In *2008 IEEE Aerospace Conference*, Big Sky, MT. (Cited at page 10.)
- Johnson, A. E. and Montgomery, J. F. (2008). Overview of Terrain Relative Navigation Approaches for Precise Lunar Landing. In *2008 IEEE Aerospace Conference*, Big Sky, MT. (Cited at page 17.)
- KANYA, . (2012). The KANYA extrusion-connecting-system PVS : The modular assembly system with infinite possibilities. Technical report. (Cited at pages 82 and 91.)
- Li, S. (2008). Computer Vision Based Autonomous Navigation for Pin-Point Landing Robotic Spacecraft on Asteroids. In *ICIRA '08 Proceedings of the First International Conference on Intelligent Robotics and Applications: Part II*, Wuhan, China. (Cited at page 29.)
- Lindeberg, T. (1994). Scale-space theory : A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):225–270. (Cited at page 60.)
- Llibre, M. (2009). *Représentation des attitudes - Angles d'Euler, Matrices de rotation, Quaternions*. ISAE, Toulouse, France. (Cited at pages 125 and 127.)
- Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110. (Cited at pages 12, 20, 21, 22, 60, 95, and 118.)

- Mazarico, E., Neumann, G. A., Rowlands, D. D., Smith, D. E., and Zuber, M. T. (2012). Topography of the Lunar Poles and Applications to the Geodesy with the Lunar Reconnaissance Orbiter. In *Proceedings of the 43rd Lunar and Planetary Science Conference*, The Woodlands, TX. (Cited at pages 13 and 102.)
- Mazarico, E., Rowlands, D. D., Neumann, G. a., Smith, D. E., Torrence, M. H., Lemoine, F. G., and Zuber, M. T. (2011). Orbit determination of the Lunar Reconnaissance Orbiter. *Journal of Geodesy*, 86(3):193–207. (Cited at pages 77 and 101.)
- Melles Griot, C. (2012). Manufacturing Grade Optical Tabletop data sheet. Technical report. (Cited at page 74.)
- Mikolajczyk, K. and Schmid, C. (2004). Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86. (Cited at pages 60, 61, 63, 118, and 120.)
- Morel, J.-M. and Yu, G. (2009). ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469. (Cited at page 120.)
- Moreno-Noguer, F., Lepetit, V., and Fua, P. (2007). Accurate Non-Iterative $O(n)$ Solution to the PnP Problem. In *2007 IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil. (Cited at page 95.)
- Mourikis, A. I. and Roumeliotis, S. I. (2007). A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy. (Cited at page 27.)
- Mourikis, A. I., Trawny, N., Roumeliotis, S. I., Johnson, A. E., Ansar, A., and Matthies, L. (2009). Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing. *IEEE Transactions on Robotics*, 25(2):264–280. (Cited at pages 19, 21, 27, 29, 30, 35, 58, and 120.)
- Murangira, A., Musso, C., and Nikiforov, I. (2012). Particle filter divergence monitoring with application to terrain navigation. In *15th International Conference on Information Fusion*, Singapore. (Cited at page 11.)
- Neal, C. R. (2009). The Moon 35 years after Apollo: What’s left to learn? *Chemie der Erde - Geochemistry*, 69(1):3–43. (Cited at page 10.)
- Northrop Grumman, . (2010). LN-200 FOG Family Advanced Airborne IMU / AHRS data sheet. Technical report. (Cited at page 136.)
- Parkes, S. M., Martin, I., Dunstan, M., and Matthews, D. (2004). Planet Surface Simulation with PANGU. In *Eighth International Conference on Space Operations*, Montreal, Canada. (Cited at page 28.)
- Pennec, X. and Thirion, J.-p. (2004). A Framework for Uncertainty and Validation of 3-D Registration Methods based on Points and Frames. *International Journal*, 25(3):203–229. (Cited at page 93.)
- Pham, B. V., Lacroix, S., and Devy, M. (2012). Vision-Based Absolute Navigation for Descent and Landing. *Journal of Field Robotics*, 29(4):627–647. (Cited at pages 12, 21, 22, 24, 28, 29, 49, and 58.)

- Pham, B. V., Lacroix, S., Devy, M., Drieux, M., and Phillippe, C. (2009). Visual Landmark Constellation matching for spacecraft pinpoint landing. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Chicago, IL. (Cited at pages 50 and 51.)
- Pollard, B. D. and Sadowy, G. A. (2005). Next Generation Millimeter-wave Radar for Safe Planetary Landing. In *IEEE Aerospace Conference*, Big Sky, MT. (Cited at page 136.)
- Sanfourche, M., Delaune, J., Le Besnerais, G., de Plinval, H., Israel, J., Cornic, P., Treil, A., Watanabe, Y., and Plyer, A. (2012). Perception for UAV : Vision-Based Navigation and Environment Modeling. *Aerospace Lab*, (4). (Cited at page 118.)
- Shuster, M. D. (1993). A Survey of Attitude Representations. *The Journal of the Astronautical Sciences*, 41(4):439–517. (Cited at page 125.)
- Sibley, G., Matthies, L., and Sukhatme, G. (2010). Sliding Window Filter with Application. *Journal of Field Robotics*, 27(5):587–608. (Cited at pages 27 and 29.)
- Simard Bilodeau, V., De Lafontaine, J., and Neveu, D. (2010). A Vision-Based Navigation Algorithm for Pinpoint Landing. In *Proceedings of the 33rd Annual AAS Rocky Mountain Guidance and Control Conference*, Breckenridge, Colorado. (Cited at page 27.)
- Singh, L. and Lim, S. (2008). On Lunar on-orbit Vision-Based Navigation : Terrain Mapping , Feature Tracking driven EKF. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii. (Cited at pages 20, 21, 22, 25, 29, and 120.)
- Siouris, G. M. (2004). *Missile Guidance and Control Systems*. (Cited at pages 11, 19, and 21.)
- Smith, D., Zuber, M., Jackson, G., Cavanaugh, J., Neumann, G., Riris, H., Sun, X., Zellar, R., Coltharp, C., Connelly, J., and Others (2010). The Lunar orbiter laser altimeter investigation on the lunar Reconnaissance orbiter mission. *Space science reviews*, 150(1):209–241. (Cited at pages 47, 77, and 136.)
- Sostaric, R. R. and Rea, J. R. (2005). Powered Descent Guidance Methods For The Moon and Mars. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, San Francisco, CA. (Cited at page 38.)
- Szeliski, R. (2011). *Computer Vision : Algorithms and Applications*. (Cited at pages 48 and 88.)
- Trawny, N., Mourikis, A. I., Roumeliotis, S. I., Johnson, A. E., and Montgomery, J. F. (2007). Vision-Aided Inertial Navigation for Pinpoint Landing using Observations of Mapped Landmarks. *Journal of Field Robotics*, 24(5):357–378. (Cited at pages 12, 20, 21, 22, 29, and 118.)
- Trawny, N. and Roumeliotis, S. I. (2005). Indirect Kalman Filter for 3D Attitude Estimation. Technical report, University of Minnesota. (Cited at page 125.)
- Vanoutryve, B., De Rosa, D., Fisackerly, R., Houdou, B., Carpenter, J., Philippe, C., Pradier, A., Jojaghaian, A., Espinasse, S., and Gardini, B. (2010). An Analysis of Illumination and Communication Conditions Near Lunar South Pole based on Kaguya Data. In *7th International Planetary Probe Workshop*, Barcelona, Spain. (Cited at pages 10 and 79.)

- Viola, P. and Well, W. M. (1997). Alignment by Maximization of Mutual Information. *International Journal of Computer Vision*, 24(2):137–154. (Cited at page 121.)
- Weinberg, J. D., Dissly, R., Nicks, D., and Miller, K. L. (2009). Applications and Field Testing of a Flash Lidar System for Future Planetary Missions. In *40th Lunar and Planetary Science Conference*, The Woodlands, TX. (Cited at page 136.)
- Welch, G. and Bishop, G. (2006). An Introduction to the Kalman Filter. Technical report. (Cited at pages 24 and 143.)
- Wolf, A. A., Graves, C., Powell, R., and Johnson, W. (2004). Systems for Pinpoint Landing at Mars. In *14th AIAA/AAS Space Flight Mechanics Meeting*, Maui, HI. (Cited at page 8.)

PUBLICATIONS

Delaune, J., Voirin, T., et al. (2012). Pinpoint Landing Optical Navigation : Preliminary Hardware Validation (poster). In *9th International Planetary Probe Workshop*, Toulouse, France.

2nd prize at the student poster competition

Delaune, J., Le Besnerais, G., et al. (2012). Optical Terrain Navigation for Pinpoint Landing: Image Scale and Position-Guided Landmark Matching. In *Proceedings of the 35th Annual Guidance and Control Conference*, Breckenridge, CO.

Sanfourche, M., Delaune, J., et al. (2012). Perception for UAV : Vision-Based Navigation and Environment Modeling. In *AerospaceLab Journal*, ONERA.

Delaune, J., Le Besnerais, G., et al. (2011). Optical and Inertial Navigation for Precision Planetary Landing: A Comparative Study of Landmark Matching Methods. In *Space Access International Conference*, Paris, France.

Delaune, J., Le Besnerais, G., et al. (2011). Camera-Aided Inertial Navigation for Pinpoint Planetary Landing on Rugged Terrains. In *8th International Planetary Probe Workshop*, Porsmouth, VA.

1st prize at the student oral presentation competition

Delaune, J., Le Besnerais, G., et al. (2011). Tightly-Integrated Visual and Inertial Navigation for Pinpoint Landing on Rugged Terrains. In *11th Symposium on Advanced Space Technologies in Robotics and Automation*, Noordwijk, Pays-Bas.

Title Vision-Based Navigation for Pinpoint Planetary Landing on any Relief

Abstract This thesis introduces *Lion*, a vision-aided inertial navigation system for pinpoint planetary landing. *Lion* can fly over any type of terrain, whatever its topography, flat or not. Landing an autonomous spacecraft within 100 meters of a mapped target is a navigation challenge in planetary exploration. Vision-based approaches attempt to pair 2D features detected in camera images with 3D mapped landmarks to reach the required precision. *Lion* tightly uses measurements from a novel image-to-map matcher in order to update the state of an extended Kalman filter propagated with inertial data. The image processing uses the state and covariance predictions from the filter to determine the regions and extraction scales in which to search for non-ambiguous landmarks in the image. The individual image scale management process per landmark greatly improves the repeatability rate between the map and descent images. We also designed a lunar-representative optical test bench called *Visilab* to test *Lion* on. The observability of absolute navigation performances in *Visilab* is evaluated with a novel error budget model. Finally, the system performances are evaluated at the key altitudes of a lunar landing, in terms of accuracy and robustness to sensor or illumination changes, off-nadir camera angle, and non-planar topography. We demonstrate error convergence down to a mean of 4 meters and a 3-RMS dispersion of 47 meters at 3 kilometers of altitude in hardware conditions at scale.

Keywords Navigation - Vision - Inertial - Landing - Moon - Precision

Titre Navigation visuelle pour l'atterrissage planétaire de précision indépendante du relief

Résumé Cette thèse présente *Lion*, un système de navigation utilisant des informations visuelles et inertielles pour l'atterrissage planétaire de précision. *Lion* est conçu pour voler au-dessus de n'importe quel type de terrain, plat ou accidenté, et ne fait pas d'hypothèse sur sa topographie. Faire atterrir un véhicule d'exploration planétaire autonome à moins de 100 mètres d'un objectif cartographié est un défi pour la navigation. Les approches basées vision tentent d'apparier des détails 2D détectés dans une image avec des amers 3D cartographiés pour atteindre la précision requise. *Lion* utilise de façon serrée des mesures venant d'un nouvel algorithme d'appariement image-carte afin de mettre à jour l'état d'un filtre de Kalman étendu intégrant des données inertielles. Le traitement d'image utilise les prédictions d'état et de covariance du filtre dans le but de déterminer les régions et échelles d'extraction dans l'image où trouver des amers non-ambigus. Le traitement local par amer de l'échelle image permet d'améliorer de façon significative la répétabilité de leur détection entre l'image de descente et l'image orbitale de référence. Nous avons également conçu un banc d'essai matériel appelé *Visilab* pour évaluer *Lion* dans des conditions représentatives d'une mission lunaire. L'observabilité des performances de navigation absolue dans *Visilab* est évaluée à l'aide d'un nouveau modèle d'erreur. Les performances du système sont évaluées aux altitudes clés de la descente, en terme de précision de navigation et robustesse au changement de capteurs ou d'illumination, inclinaison de la caméra de descente, et sur différents types de relief. *Lion* converge jusqu'à une erreur de 4 mètres de moyenne et 47 mètres de dispersion 3 RMS à 3 kilomètres d'altitude à l'échelle.

Mots-clés Navigation - Vision - Inertiel - Atterrissage - Lune - Précision