UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Dipartimento di
Centro di Ateneo di Studi e Attività Spaziali “Giuseppe Colombo”

CORSO DI DOTTORATO DI RICERCA IN: Scienze Tecnologie e Misure Spaziali
CURRICOLO: Misure Meccaniche per l’Ingegneria e lo Spazio
CICLO XXIX

VISUAL ODOMETRY AND VISION SYSTEM MEASUREMENTS BASED ALGORITHMS FOR ROVER NAVIGATION

Coordinatore: Ch.mo Prof. Giampiero Naletto
Supervisore: Ch.mo Prof. Stefano Debei
Co-Supervisore: Dott. Ing. Marco Pertile

Dottorando : Sebastiano Chiodini

Declaration

The research described in Chapters 1.2, 2.5.2 and 4 of this Ph.D. thesis was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by JPL Visiting Student Research Program and the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

Acknowledgements

First and foremost I want to thank my advisor Prof. Stefano Debei, for his warm encouragement and thoughtful guidance. I am grateful to Marco Pertile for his help.

I would like to express my sincere gratitude to my JPL mentor Robert Reid for all the support and the technical assistance that he gave to me during my intern-ship at Jet Propulsion Laboratory. I wish to thank Issa Nesnas who provided me the opportunity to join the Robotic Mobility Group (347F) as intern. I also thank the Formation Control Testbed team (Carl Seubert, Kevin Lo and Dylan Conway) that contributed to the results presented herein. Thanks to Thomas Paynter for helping me during the experiment set-up.

I am grateful to Temis Srl for the financial support.

I would like to thank Roberto Trucco, Ivano Musso, Lorenzo Bramante and Enrico Ferrentino, who have given me the possibility to collaborate with ALTEC S.p.A and for the useful discussions we had.

Sommario

I *rover* marziani e, più in generale, i robot per l'esplorazione di asteroidi e piccoli corpi celesti, richiedono un alto livello di autonomia. Il controllo da parte di un operatore deve essere ridotto al minimo, al fine di ridurre i tempi di percorrenza, ottimizzare le risorse allocate per le tele-comunicazioni e massimizzare l'output scientifico della missione.

Conoscendo la posizione obiettivo e considerando la dinamica del veicolo, gli algoritmi di controllo forniscono gli input adeguati agli attuatori. Algoritmi di pianificazione della traiettoria, sfruttando modelli tridimensionali del terreno circostante, evitano gli ostacoli con ampi margini di sicurezza. Inoltre i rover per le missioni di *sample and return*, previste per i prossimi anni, devono dimostrare la capacità di tornare in un luogo già visitato per il campionamento di dati scientifici o per riportare i campioni raccolti ad un veicolo di risalita. In tutte queste *task* la stima del moto risulta essere fondamentale. La stima del moto su altri pianeti ha la sua peculiarità. L'odometria tramite encoder, infatti, presenta elevate incertezze a causa dello slittamento delle ruote su superfici sabbiose o scivolose; i sistemi di navigazione inerziale, nel caso della dinamica lenta dei rover, presentano derive non tollerabili per una stima accurata dell'assetto; infine non sono disponibili sistemi di posizionamento globale analoghi al GPS.

Sistemi della stima del moto basati su telecamere hanno dimostrato, già con le missioni MER della NASA, di essere affidabili e accurati. Uno di questi sistemi è l'odometria visuale stereo. In questo algoritmo il moto è stimato calcolando la roto-traslazione di due nuvole di punti misurate a due istanti successivi. La nuvola di punti è generata tramite triangolazione di punti salienti presenti nelle due immagini. Grazie a tecniche di *Simultaneous Localization and Mapping* (SLAM) si dà la capacità ad un rover di costruire una mappa dell'ambiente circostante e di localizzarsi rispetto ad essa. Le tecniche di SLAM presentano due vantaggi: la costruzione della mappa e una stima della traiettoria più accurata, grazie alla soluzione di problemi di minimizzazione che coinvolgono la stima di più posizioni e *landmark* allo stesso tempo.

Subito dopo l'atterraggio, una delle task principali che devono essere svolte dal centro operativo per il controllo di rover è il calcolo accurato della posizione del *lander/rover*

rispetto al sisma di riferimento inerziale e il sistema di riferimento solidale al pianeta, come il sistema J2000 e il Mars Body-Fixed (MBF) frame. Sia per le operazioni scientifiche che ingegneristiche risulta fondamentale la localizzazione accurata rispetto a immagini satellitari e a modelli tridimensionali della zona di atterraggio.

Nella prima parte della tesi viene trattato il problema della localizzazione di un *rover* rispetto ad un'immagine satellitare geo referenziata e orto rettificata e la localizzazione rispetto ad un modello di elevazione digitale (DEM), realizzato da immagini satellitari. È stata svolta l'analisi di una versione modificata dell'algoritmo *Visual Position Estimator for Rover* (VIPER). L'algoritmo trova la posizione e l'assetto di un rover rispetto ad un DEM, comparando la linea d'orizzonte locale con le linee d'orizzonte calcolate in posizioni a priori del DEM. Queste analisi sono state svolte in collaborazione con ALTEC S.p.A., con lo scopo di definire le operazioni che il *Rover Operation Control Center* (ROCC) dovrà svolgere per la localizzazione del rover ExoMars 2020. Una volta effettuate le operazioni di localizzazione, questi metodi possono essere nuovamente utilizzati come verifica e correzione della stima della traiettoria.

Nella seconda parte della dissertazione è presentato un metodo di odometria visuale stereo per rover ed un'analisi di come la distribuzione dei *landmark* triangolati influisca sulla stima del moto. A questo scopo sono stati svolti dei test in laboratorio, variando la distanza della scena. L'algoritmo di odometria visiva implementato è un metodo 3D-to-3D con rimozione dei falsi positivi tramite procedura di *RANdom SAmple Consensus*. La stima del moto è effettuata minimizzando la distanza euclidea tra le due nuvole di punti.

L'ultima parte di questa dissertazione è stata sviluppata in collaborazione con il Jet Propulsion Laboratory (NASA) e presenta un sistema di localizzazione per rover *hopping/tumbling* per l'esplorazione di comete e asteroidi. Tali sistemi innovativi richiedono nuovi approcci per la localizzazione. Viste le risorse limitate di spazio, peso e energia disponibile e le limitate capacità computazionali, si è scelto di basare il sistema di localizzazione su una monocamera. La localizzazione visuale in prossimità di una cometa, inoltre, presenta alcune peculiarità che la rendono più difficoltosa. Questo a causa dei grandi cambiamenti di scala che si presentano durante il movimento della piattaforma, le frequenti occlusioni del campo di vista, la presenza di ombre nette che cambiano con il periodo di rotazione dell'asteroide e la caratteristica visiva del terreno, che risulta essere omogeno nel campo del visibile.

È stato proposto un sistema di visual SLAM collaborativo tra il rover *tumbling/hopping* e il satellite "madre", che ha portato il rover nell'orbita di rilascio. È stato effettuato lo stato dell'arte dei più recenti algoritmi di *visual* SLAM *open-source* e, dopo un'accurata analisi, si è optato per l'utilizzo di ORB-SLAM2, che è stato modificato per far fronte al tipo di applicazione richiesta. È stata introdotta la possibilità di salvare la mappa realizzata

dall'orbiter, che viene utilizzata dal rover per la sua localizzazione. È possibile, inoltre, fondere la mappa realizzata da orbiter con altre misure d'assetto provenienti da altri sensori a bordo dell'orbiter.

L'accuratezza di tale metodo è stata valutata utilizzando una sequenza di immagini raccolta in ambiente rappresentativo e utilizzando un sistema di riferimento esterno. Sono state effettuate simulazioni della fase di mappatura dell'asteroide e localizzazione della piattaforma *hopping/tumbling* e, infine, è stato valutato come migliorare le performances di questo metodo, in seguito al cambiamento delle condizioni di illuminazione.

Abstract

Planetary exploration rovers should be capable of operating autonomously also for long paths with minimal human input. Control operations must be minimized in order to reduce traverse time, optimize the resources allocated for telecommunications and maximize the scientific output of the mission.

Knowing the goal position and considering the vehicle dynamics, control algorithms have to provide the appropriate inputs to actuators. Path planning algorithms use three-dimensional models of the surrounding terrain in order to safely avoid obstacles. Moreover, rovers, for the sample and return missions planned for the next years, have to demonstrate the capability to return to a previously visited place for sampling scientific data or to return a sample to an ascent vehicle.

Motion measurement is a fundamental task in rover control, and planetary environment presents some specific issues. Wheel odometry has wide uncertainty due to slippage of wheels on a sandy surface, inertial measurement has drift problems and GPS-like positioning systems is not available on extraterrestrial planets. Vision systems have demonstrated to be reliable and accurate motion tracking measurement methods. One of these methods is stereo Visual Odometry. Stereo-processing allows estimation of the three-dimensional location of landmarks observed by a pair of cameras by means of triangulation. Point cloud matching between two subsequent frames allows stereo-camera motion computation. Thanks to Visual SLAM (Simultaneous Localization and Mapping) techniques a rover is able to reconstruct a consistent map of the environment and to localize itself with reference to this map. SLAM technique presents two main advantages: the map of the environment construction and a more accurate motion tracking, thanks to the solutions of a large minimization problem which involves multiple camera poses and measurements of map landmarks.

After rover touchdown, one of the key tasks requested to the operations center is the accurate measurement of the rover position on the inertial and fixed coordinate systems, such as the J2000 frame and the Mars Body-Fixed (MBF) frame. For engineering and science operations, high precision global localization and detailed Digital Elevation Models (DEM) of the landing site are crucial.

The first part of this dissertation treats the problem of localizing a rover with respect to a satellite geo-referenced and ortho-rectified images, and the localization with respect to a digital elevation model (DEM) realized starting from satellite images A sensitivity analysis of the Visual Position Estimator for Rover (VIPER) algorithm outputs is presented. By comparing the local skyline, extracted form a panoramic image, and a skyline rendered from a Digital Elevation Model (DEM), the algorithm retrieve the camera position and orientation relative to the DEM map. This algorithm has been proposed as part of the localization procedure realized by the Rover Operation Control Center (ROCC), located in ALTEC, to localize ExoMars 2020 rover after landing and as initialization and verification of rover guidance and navigation outputs. Images from Mars Exploration Rover mission and HiRISE DEM have been used to test the algorithm performances.

During rover traverse, Visual Odometry methods could be used as an asset to refine the path estimation. The second part of this dissertation treats an experimental analysis of how landmark distributions in a scene, as observed by a stereo-camera, affect Visual Odometry measurement performances. Translational and rotational tests have been performed in many different positions in an indoor environment. The Visual Odometry algorithm, which has been implemented, firstly guesses motion by a linear 3D-to-3D method embedded within a RANdom SAmple Consensus (RANSAC) process to remove outliers. Then, motion estimation is computed from the inliers by minimizing the Euclidean distance between the triangulated landmarks.

The last part of this dissertation has been developed in collaboration with NASA Jet Propulsion Laboratory and presents an innovative visual localization method for hopping and tumbling platforms. These new mobility systems for the exploration of comets, asteroids, and other small Solar System bodies, require new approaches for localization. The choice of a monocular onboard camera for perception is constrained by the rover's limited weight and size. Visual localization near the surface of small bodies is difficult due to large scale changes, frequent occlusions, high-contrast, rapidly changing shadows and relatively featureless terrains.

A synergistic localization and mapping approach between the mother spacecraft and the deployed hopping/tumbling daughter-craft rover has been studied and developed. We have evaluated various open-source visual SLAM algorithms. Between them, ORB-SLAM2 has been chosen and adapted for this application. The possibility to save the map made by orbiter observations and re-load it for rover localization has been introduced. Moreover, now it is possible to fuse the map with other orbiter sensor pose measurement.

Collaborative localization method accuracy has been estimated. A series of realistic images of an asteroid mockup have been captured and a Vicon system has been used in order

to give the trajectory ground truth. In addition, we had evaluated this method robustness to illumination changes.

Acronyms

ALTEC	**A**erospace **L**ogistics **T**echnology **E**ngineering **C**ompany
BRIEF	**B**inary **R**obust **I**ndependent **E**lementary **F**eatures
CCD	**C**harge-**C**oupled **D**evice
DEM	**D**igital **E**levation **M**odel
DOF	**D**egrees **O**f **F**reedom
DLR	**D**eutschen Zentrums für **L**uft- und **R**aumfahrt
EDL	**E**ntry **D**escent and **L**anding
EKF	**E**xtended **K**alman **F**ilter
ESA	**E**uropean **S**pace **A**gency
ESOC	**E**uropean **S**pace **O**perations **C**entre
FAST	**F**eatures from **A**ccelerated **S**egment **T**est
FOV	**F**ield **O**f **V**iew
GESTALT	**G**rid-Based **E**stimation of **S**urface **T**raversability **A**pplied to **L**ocal **T**errain
GPS	**G**lobal **P**ositioning **S**ystem
GUM	**G**uide to the Expression of **U**ncertainty in **M**easurement
IMU	**I**nertial **M**easurement **U**nit
HiRISE	**Hi**gh **R**esolution **I**maging **S**cience **E**xperiment
JAXA	**J**apan **A**erospace **E**xploration **A**gency
JPL	**J**et **P**ropulsion **L**aboratory
MAV	**M**ars **A**scent **V**ehicle
MBF	**M**ars **B**ody-**F**ixed
MER	**M**ars **E**xploration **R**overs
MGS	**M**ars **G**lobal **S**urveyor
MOC	**M**ars **O**rbiter **C**amera
MOLA	**M**ars **O**rbiter **L**aser **A**ltimeter
MRO	**M**ars **R**econnaissance **O**rbiter
MSL	**M**ars **S**cience **L**aboratory
NAC	**N**arrow **A**ngle **C**amera
NASA	**N**ational **A**eronautics and **S**pace **A**dministration
NAVCAM	**NAV**igational **CAM**era

ORB	**O**riented FAST and **R**otated **B**RIEF
PTAM	**P**arallel **T**racking **A**nd **M**apping
RANSAC	**RAN**dom **SA**mple **C**onsensus
ROCC	**R**over **O**peration **C**ontrol **C**enter
RMS	**R**oot **M**ean **S**quare
SfM	**S**tructure **f**rom **M**otion
SIFT	**S**cale-**I**nvariant **F**eature **T**ransform
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping
SURF	**S**peeded **U**p **R**obust **F**eature
UAV	**U**nmanned **A**ircraft **V**ehicle
UHF	**U**ltra **H**igh **F**requency
VIPER	**V**isual **P**osition **E**stimator for **R**over
VO	**V**isual **O**dometry
WAC	**W**ide **A**ngle **C**amera

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Thanks to *in situ* scientific measurements, robotics explorers could greatly increase the scientific return of planetary and small solar system bodies exploration missions. NASA Mars Exploration Rovers with the synergistic coordination of three orbiter, Mars Odyssey, Mars Express and Mars Reconnaissance Orbiter, have shown that the surface of Mars has been modelled by the iteration with water across its history. Mars exploration program is now following a scientific goal know as "Seek Signs of Life". Mars Science Laboratory (MSL), which was designed to search for past and present habitable environments at Gale crater, has the capability to detect complex organic molecules in rocks and soils [1]. MSL mission had also demonstrated long-range mobility, long term surface operations and the sky crane precision landing. MSL, launched in 2011, has the following scientific objectives: assess the biological potential of at least one target; study the geology and geochemistry of the landing region; examine planetary process relevant to past habitability and describe the spectrum of the surface radiation. The Pasteur payload of the ExoMars 2018 rover, with his suite of instruments dedicated to exobiology and geochemistry, is able to analyse sample up to a depth of 2 m [2]. It will investigate signs of the past and present life on Mars, inspect the planet's subsurface and deep interior in order to figure out the evolution and the habitability of Mars.

In the next decade Mars Exploration efforts will be focused in the preparation of a Mars sample-return campaign. The possibility to analyse Mars samples on Earth will be open a deeper understanding of the planet Mars, understand the process and history of climate, the geological evolution of the surface and determine if life ever arose on Mars. NASA scheme for this campaign foresee a sequence of three mission: a rover mission to collect and cache the samples, afterward the cached samples will be sent to a Mars Sample Return Orbiter by means of a Mars Ascent Vehicle (MAV) and returned to Earth for an intensive analysis.

Key Technologies for achieves these scientific output are the development of surface robotic explorer that should be able to select samples and documenting the geological context. These explorers should integrate imaging system and remote spectroscopy to establish the context and identify targets. The ability to select potential samples should not be only related to location where liquid water has occurred but also related to the detection of organic molecules and to the possibility of reconstructing the geochemical history of the target rock formation as indicator of organic matter or coupled redox reactions characteristic of life [3].

Rover and robotic technologies will be also fundamental to prepare and as support for a manned mission to Mars. Robotic explorers will be essential to obtain information about hypothetical resources and hazards, testing flight system and build infrastructures for astronauts' exploration activities [4].

The benefit of in situ measurement for the solar system exploration it is not only limited to the Mars and planetary investigations. In situ exploration at designated or multiple location of small solar system body such as Near-Earth Objects and the moons of Mars, will help the scientific community to characterize the early solar system history, to study planetary habitats and highlight the mechanism of planetary process formation [5]. Asteroids, comets and small solar system bodies are characterized by weak gravitational fields, for this reason new mobility concept are needed. Many different small-body mobility concepts have been proposed: hopping, wheeled, legged, hybrid and other mobility platforms.

Spacecraft orbiter and landers soft-landing have been realized on asteroids and comets. In 2001 NASA's NEAR Shoemaker landed on asteroid Eros after a year of observations, during the soft-landing has been possible to obtain images at a resolution much greater than during orbit [6]. JAXA's Hayabusa mission performed two touchdown at Muses Sea on asteroid Itokawa in 2005 [7]. The Philae lander, part of the ESA Rosetta mission, landed on the surface if comet 67P/Churyumov-Gerasimenko in November 2014. Thanks to the circumstances of the multiple landings, it was possible to characterize the mechanical properties (strength and layering) of multiple regions of the comet surface [8]. The 30th of September 2016 token place the controlled impact of Rosetta with the comet, giving the opportunity to study the comet's gas, dust and plasma environment very close to comet surface [9].

To date no robotics mobility system have ever been successfully deployed to the surface of a small body. It is possible to find a review of the attempted deployment to the surface of a small body, and of the state of the art design of such mobility systems in [5]. Soviet RKA Phobos 2 mission was designed to carry on the surface of Phobos a 41-kg robotic hopper called PROP-F, but communications were lost before the hopper deployments [10]. MINERVA rover was developed by JAXA/ISAS for the Hayabusa mission, but unfortunately

the rover fails upon deployments. Currently the DLR rover MASCOTT is part of the Haybusa 2 mission, launched on December 2014 its arrival to the comet is planned to July 2018 [11].

A synergistic approach between an orbiter and a robotic explorer is fundamental for mission success and increase its scientific output. Global positioning is needed to correlate Rover scientific measurements with remote orbiter measurement, and other mission measurement to the same celestial body, and validate planetary global models. Detailed Digital Elevation Models (DEM), produced from orbiter images and stereo-photogrammetry, provides significant support to the landing site selection and to plan rover operations.

Due to spacecraft low-bandwidth and communication delay to Earth, rovers must utilize a high degree of autonomy. Starting from MER rovers, vision system measurements have demonstrated to be an asset for rovers operations during traverse. Relative localization by means of Visual Odometry (VO) has enabled the rovers to drive safely and more effectively in highly sloped and sandy terrains. Indeed, methods that rely on wheel odometry and Inertial Measurement Units (IMU) perform poorly in harsh terrain. As a results of using VO and vision system techniques mission science return has been increased, because they have reduced the number of days required to drive into interesting areas [12].

Nowadays rovers navigate the surface at a rate of 20-30 m/sol using a very computational demanding pipeline. First of all the terrain traversability is verified thanks to stereo imagery, then the rover plans its motion and finally conducts its manoeuvre. Every half-meter step can take up to several minutes. These limits are caused by the available on-board computation and power of the flight-qualified processors. Of great interest for future applications, will be to fuse onboard sensing with higher-resolution orbital imagery for assessing terrain traversability in more effective and automated ways (starek et al.). Moreover, sample and return missions rovers, planned for the next years, have to demonstrate the capability to return a collected sample to the ascent vehicle. This thesis work presents the attempts to fuse the visual information collected by surface imaging of a rover with orbital imagery data. In collaboration with ALTEC and in the framework of the operational analysis for ExoMars 2020 rover, a global localization algorithm which exploited surface images and a DEM obtained by orbiter images has been investigated. For local traverse we have studied a stereo Visual Odometry algorithm.

A synergistic localization and mapping approach between the mother spacecraft and the deployed hopping/tumbling daughter-craft rover has been studied and developed in collaboration with NASA Jet Propulsion Laboratory. By means of visual Simultaneous Localization and Mapping had been possible to use the same computer efficient algorithm for global localization on the asteroid and relative localization during traverse. Visual localization near the surface of small bodies is difficult due to large scale changes, frequent

occlusions, high-contrast, rapidly changing shadows and relatively featureless terrains. Visual localization near the surface of small bodies is difficult due to large scale changes, frequent occlusions, high-contrast, rapidly changing shadows and relatively featureless terrains

1.1 Mars Exploration

In 1997, twenty years after the Viking missions, NASA revisited the planet Mars with Pathfinder lander. The lander contained the mobile robot, Sojourner, a 12 kg six-wheeled mobile robot, which was designated in order to demonstrate mobile robot mobility on Mars. Sojourner rover has taken pictures and has positioned a science instrument against designated soil and rocks.

In [13] we can find a detailed description of Sujourner operations. This rover, which was based on Rocky III and IV JPL robotics demonstrators [14], had six wheels powered by six separate motors (four of these wheels were steerable) and the rocker-bogie system. Rocker-bogie system with six motorized wheels enable the rover to turn around in-place and climb a rock with a rise greater than the diameter of the wheels. Moreover, this rover is able to accomplish multiple movements like wheel spinning, skidding, plowing and trenching. The nominal speed of this rover was 0.4 m/min and maximal turning rate was 7 deg/s. A scheme of Sojourner is depicted in Figure 1.1.

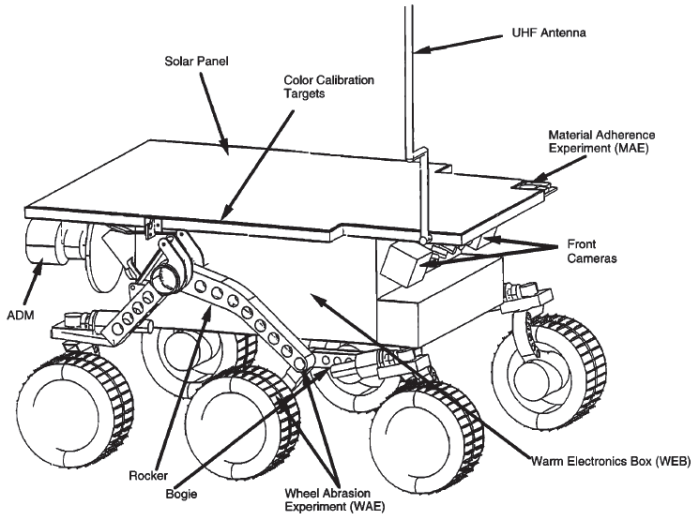


Fig. 1.1 Sketch of Sojourner Rover; solar panel with color calibration targets along the edge, rockebogie suspension system, wheels, front cameras, UHF antenna, warm electronics box, material adherence experiment, wheel abrasion experiment, and APXS deployment mechanism. *Image courtesy of [13].*

To navigate a step by step approach has been adopted. The moment to switch off the motors was triggered on an average of motor encoder (drive) or potentiometer (steering) readings. When the rover was stopped the on-board computer conducted proximity and hazard detection function. The laser striping and camera system has been used to determine the presence of obstacles in its path. In addition, the estimation of the travelled distance and of the heading was computed while the rover was stopped. The number of turns of the wheel motors and the onboard gyroscope was used to provide an estimate of progress to the commanded location and the distance travelled from the lander, both was expressed in the local reference frame.

The Sojourner rover operations range were limited by the lander camera field of view, indeed at the end of each sol of rover traverse, the lander provided a stereo image of the vehicle position in the surrounding terrain. Stereo images, portions of terrain panorama and images from the rover camera was used by an operator at the control station to designated new points as target location for next sol rover traverse. The image size provided by the rover camera is 768×484 pixels.

Thanks to Rocky 7 prototype rover JPL has demonstrate the capability to traverse natural terrain up to a distance of one kilometer. We can find a description of Rover 7 navigation in [15]. The navigation sensors of Rocky 7 are listed below. Three accelerometers measured the tilt of the chassis and potentiometers measured the configuration of the rocker-bogey suspension. The relative heading was provided by the integration over time of the signal of a quartz rate sensor, which measured the rate of rotation of the vehicle about its vertical axis. The major drawback of this method is related to drift as noise and bias error are integrated with the rate signal. In order to provide a reliable measurement of the vehicle heading a wide field of view sun sensor has been employed. The sun sensor was attached to the rover facing upward. By using the accelerometer signal to determine sensor tilt and local time from an on-board clock, was possible to compute the absolute vehicle heading.

Visual Odometry odometry capabilities have been demonstrated for the first time on a planetary environment during Mars Exploration Rover missions: Spirit (MER-A) and Opportunity (MER-B). The twin rovers carries identical science and engineering instrument payloads, a scheme of the rover with its sensors is shown in Figure 1.2. Spirit have explored the landing sites of Gusev Crater, while Opportunity is still exploring Meridiani Planum on the Martian surface. Pancam (panoramic camera) and Navcam (navigation camera) not only have been used for scientific investigation but also as primary instrument for localization and mapping. A description of MER rover sensor, localization strategy and topographic measurements are given in [16–19]. Pancam and Navcam are two stereo cameras with different characteristics. Navcam is used in mapping close objects, has a baseline of 20 cm, a

focal length of 14.67 mm, and an effective depth of fiels of 0.5 meters to infinity, it has a field of view of 45 deg. Pancam is used in mapping far object, has a stereo base of 30 cm and a focal length of 43 mm, the focal depth of field is 3 meters to infinity, its FOV is 16 degrees. The image size for both the cameras is 1024×1024 pixel.

Remote observations performed with Pancam and mini-TES, an infrared spectrometer, are used for identifying promising rocks and soil for detailed studies with the other sensor onboard the rover. After target selection from science team, the rovers were then commanded to position the instrument deployment device (IDD) to the target location. IDD is used to place the rovers analysis tools, like the Mössbauer Spectrometer (MB), Alpha Particle X-Ray Spectrometer (APXS), Microscopic Imager (MI), or Rock Abrasion Tool (RAT) onto rocks or soil for in situ observations. All the rovers operations require accurate relative three-dimensional position of the rover itself and of the target object. Moreover, rover localization has to be converted into the Mars inertial and Mars body-fixed (MBF) reference frame, in order to localize the observations in a regional context captured from orbiter images. An illustration of MER rover cameras and sensors is presented in Figure 1.2

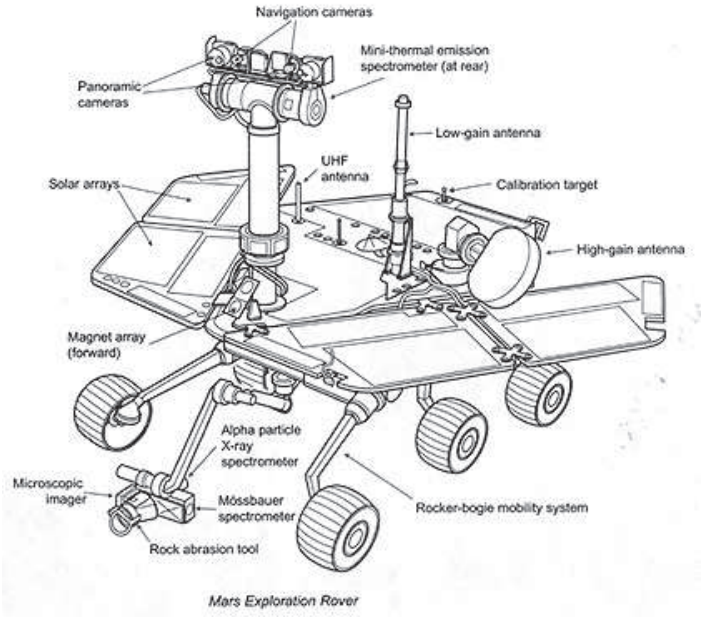


Fig. 1.2 Sketch of Mars Exploration Rovers.

The global localization of the MER rovers were fundamental for planning science and engineering operations. The localization procedure were accomplished during the first eight days after landing (before the rover starts its traverse). First of all rovers position has been estimated in the inertial reference frame through UHF Two-Way Doppler Tracking. Thanks to this technique rover could be localized up to an accuracy of 10 m. The conversion from

the inertial frame to the MBF frame led to an uncertainty in the position of ± 250 m. In order to have an accurate positioning in the regional context, landing site positions were calculated reconstructing Entry, Descent and Landing in DIMES descent images and finding common features with orbiter images in the MBF frame and in the Landing Site Local (LSL) reference frame. When the first panoramas composed by surface images from Navcam and Pancam were available, was possible to triangulate crater and mountain peaks that may appear in the orbital images.

For example the Spirit lander location determined in the inertial space and translated to the MBF system (MOLA IAU 2000) is 14.571892°S, 175.47848°E, and using the cartographic triangulation, exploiting the Mars Global Surveyor (MGS) Mars Orbiter Camera (MOC), is 14.5692°S, 175.4729°E. The Opportunity lander location in the inertial reference frame is. 1.948282°S, 354.47417°E, and using the cartographic triangulation 1.9462°S, 354.4734°E [20].

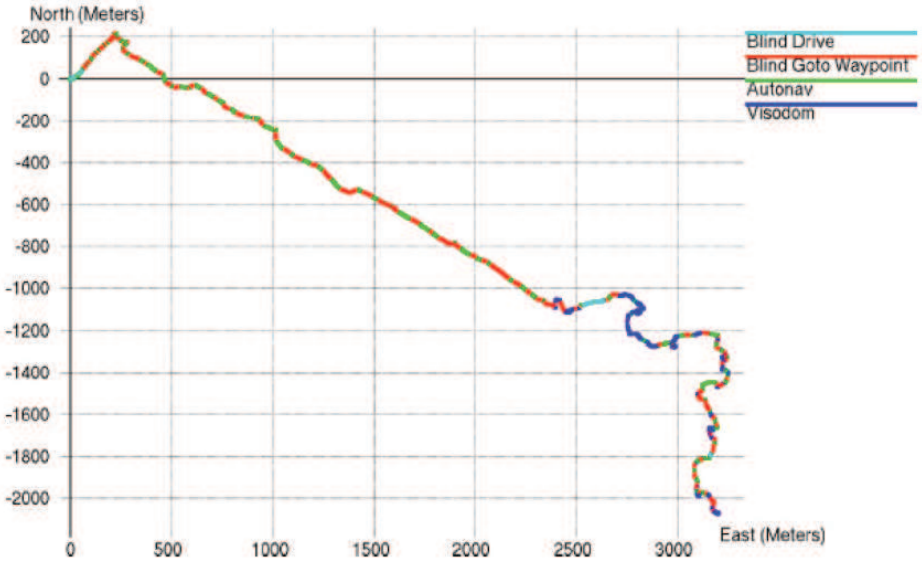


Fig. 1.3 Plot of Spirit's traverse history using Visual Odometry in the Columbia Hills from sols 1–850. Units are in meters from the landing site origin, as measured onboard the rover. Cyan lines indicate directly commanded "blind" drives, red lines indicate blind drives with autonomous heading compensation, green lines indicate autonomous hazard detection, and blue lines indicate Visual Odometry. Spirit only used Visual Odometry within the Columbia Hills, not during its 3 km trek to reach them. *Image courtesy of [12].*

The navigation sensors of MER rovers consist of an a Litton LN-200 IMU, and odometer and the Pancam used as solar imaging camera. The position estimation by means of the integration of IMU gyros led to several degrees of drift. The sun vector, measured by Pancam, with the local vertical and the current local solar time, were used in order eliminate the

angular drift. Generally, MER rovers have been commanded only once per "sol" through a prescheduled sequence of precise metrically specified commands. As explained in [12], the navigation goal for MER rovers was to bound the position estimation error to be no more than 10% during a 100 m drive. At each rover movement the Rover pose was updated at 8Hz. The movement estimation was given by the combination of IMU attitude signal with wheel odometry. This navigation technique was efficient for great part of the terrains, but not on slippery surfaces or slippery slopes. Furthermore [12] paper of highlight the Visual Odometry process and performances used on-board the MER rovers. Visual Odometry has been mainly used to correct the erroneous wheel odometry-based estimation when wheel lose traction on large rocks and steep slopes. VO onboard MER rovers computed an update of 6-DOF rover pose (x, y, z, pitch, roll and yaw) tracking the position of terrain features between two stereo images, in the 2D image space and in 3D space. Rover final motion estimation was realized thanks to a maximum likelihood estimator applied to the computed 3D offset.

Visual Odometry was conceived as an "extra credit" capability and evolved into a critical vehicle safety system, the VO software has been extensively used during high-tilt operations, as shown in Figure 1.3. VO processing on-board MER rover taken long time, effective time speed was reduced up to an order of magnitude. The necessity to use VO has been a compromise between the better position estimation an the desire to cover longer distances. VO has demonstrated good performance with high rates of successful convergence 97% on Spirit, 95% on Opportunity and measured movements as small as 2 mm. VO was used over 14% of the first 10.7 km driven by both rovers during the first two years of operations.

An overview of the MER rover autonomous capabilities is given in [21, 22]. Many software functionality were implemented in the flight software before landing moment, many others were uploaded during surface mission (April 2004, January 2005, September 2006). It is possible to identify four main autonomous driving modes. *Direct* drives commanded the rover to a target position, expressed in the local reference frame, without any compensation for position and attitude drift, and no checking for obstacles. *Visual Odometry* drives compensated the position and attitude drifts but did not check for obstacles. *Terrain Assessment* drives checked for geometric hazards, but did not measure any slip. *Local Path Selector* drives corrected the path based on VO drift estimation and Terrain Assessment.

The geometric information needed by *Terrain Assessment* software for safe navigation was provided by a 3D point cloud. The 3D point cloud has been obtained by applying to the stereo images a windowed 1D search using the Sum-of-Absolute-differences. Images for point cloud generation were down sampled to 256×256 pixels. Geometric hazards in the area around the rover were detected thanks to GESTALT (Grid-based Estimation of surface

Traversability Applied to Local Terrain), the system was able to detect step obstacles, tilt hazards and roughness hazards. Terrain Assessment has been principally used in combination with other driver modes in order to extend the distance driven by the rovers beyond the edge of what can be see by the surface images sent to the human operator.

Local Path Selector mode foreseen a correction of rover's path as it drives toward the target location. In the simplest version of this mode there is not vision mode. The on–board computer uses information by rover current pose obtained by wheel encoders and IMU. It is different form directed drives where the commands are pre-planned motor rotations not taking into account other sensor measurements than wheel encoders. *Terrain Assessment* used in combination with *Local Path Selector* has given to MER rovers a complete obstacle avoidance capability.

The *Local Path Selector* was useful only for the presence of small obstacle. The Field D^* planner has been implemented to avoid bigger obstacles, it maintains a much larger world map (typically 50×50 meters^2 with 0.4 meter cells) and provides the ability to plan arbitrary paths through its map.

As reported in [22], during the first two years of operations, Sprit's longest commanded drive was 124 meters, for 62 meters it used direct driving mode and for the other 62 meters it used Terrain Assessment and Local Path Selection. Opportunity's longest drive travelled distance was 390 meters, divided in 106 meter Local Path Selection without any vision processing and 284 meters using a combination of Local Path Selection with Terrain Assessment. Figure 1.4 Spirit avoided obstacles in previously-unseen terrain

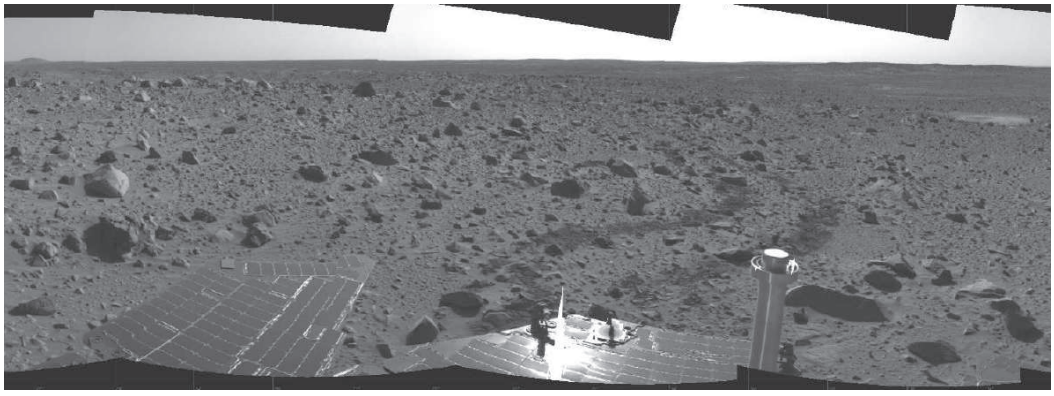


Fig. 1.4 On Sol 107, Spirit avoided obstacles in previously-unseen terrain. *Image courtesy of [22].*

The last NASA's rover sent to Mars has been Curiosity, in the frame of Mars Science Laboratory (MSL) mission. Landed on August 5, 2012, Curiosity is still exploring Gale Crater. [1] describes MSL primary scientific objectives and instruments. Figure 1.5 sows an

image of Curiosity rover [23] report the hazard detection and navigation cameras on-board Curiosity. MSL rover is equipped with 12 engineering cameras, Navigation cameras have a 45 deg FOV and are mounted to a pan/tilt mast. The Hazard Avoidance Cameras have a FOV of 124 deg, and are rigidly mounted to the rover chassis in the front and rear of the vehicle. All of the cameras utilize a 1024 × 1024 pixel detector.

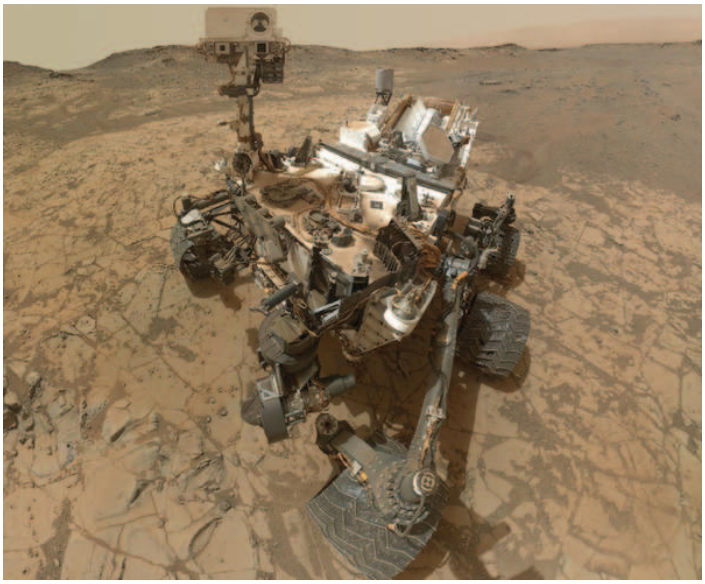


Fig. 1.5 Curiosity self-portrait mosaic acquired by the Mars Hand Lens Imager on the robotic arm on sols 868 to 884. *Image courtesy of [24].*

[25] reviews Curiosity's rover autonomous capabilities. Curiosity inherited many technologies developed for MER rovers mission, like the rocker/bogie suspension, the attitude measurement framework by means of the on-board IMU, the hazard detection system and the navigation cameras. Great part of Curiosity autonomous capabilities are similar to MER rovers ones, but, in order to face the challenges le by a bigger and more complex system onboard software has been updated and the onboard CPU, 200 MHz, is 10 times faster than MER CPU. Multiresolution tracking has been added to the VO software and the total processing time has been reduced to 40 seconds per step. The changes made by [26] to the VO software have enable a more extensive use of VO during MSL mission. Curiosity is able to detect about terrain hazard by looking stereo, it look to step, tilt or roughness hazards using an updated version of GESTALT terrain assessment software.

1.1.1 ExoMars Mission 2020

The ExoMars mission is a join collaboration between ESA and Roscosmos, it consist of the Trace Gas Orbiter plus an Entry, Descent and landing demonstrator Module (EDM),

Schiaparelli, launched on March 2016, and the other, will carry a rover, the launch date is 2020. The ExoMars rover has been designed to search for signs of life. It will collect samples with a drill and analyse them with next-generation instruments [2]. ExoMars will be the first mission to study Mars at depth up to two meters [m].

The ExoMars Trace Gas Orbiter, part of the 2016 ExoMars mission, will support communications. The Rover Operations Control Centre (ROCC) will be located in ALTEC, Turin, Italy. The ROCC will monitor and control the ExoMars rover operations. Commands to the Rover will be transmitted through the Orbiter and the ESA space communications network operated at ESA's European Space Operations Centre (ESOC).

The global localization studies have been developped in collaboration with ALTEC for ExoMars 2020 rover.

Fig. 1.6 The ExoMars rover. Image Credit: ESA.

1.2 Small Solar System Bodies Exploration

To date no robotics mobility system have ever been successfully deployed to the surface of a small body. The two attempted deployment to a small body surface have been: the Soviet RKA PROP-F hopper of the Phobos-2 mission, and the JAXA/ISAS MINERVA rover for the Hayabusa mission. Currently the DLR rover MASCOTT, part of the Haybusa 2 mission, it is heading towards the asteroid 162173 Ryugu.

MINERVA (MIcro/Nano Experimental Robot Vehicle for Asteroid) a small 591 g explorer was part of the JAXA mission Hayabusa, the target of the mission was the exploration of asteroid "Itokawa". The rover was able to move autonomously over the microgravity environment and had few science instruments to characterize the surface of the asteroid. The

rover was deployed on 12 November 2005, unfortunately the rover fails upon deployment. [27, 28] describe the rover actuation system and navigation strategy.

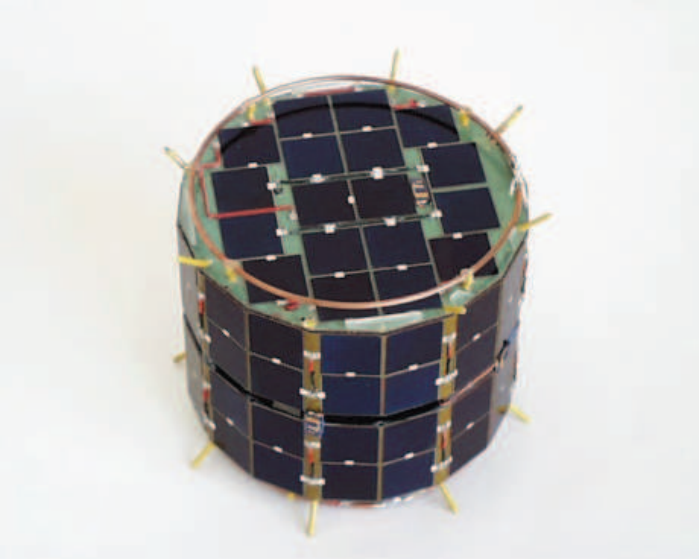


Fig. 1.7 MINERVA small rover. *Image courtesy of [28].*

DC motors gave MINERVA hopping ability. The torque given by one of the two motors was used for making the rover hop, the second one rotates the plate which supported the first motor. The friction between the rover and the surface is not predictable, so it is not possible to control precisely the hopper speed. The on-board DC motor provides a maximal speed of 9 [cm/s], the hop speed was set in order to nor exceed the asteroid escape velocity.

Sensors dedicated to navigation were limited in size and weight, also the computational capabilities were limited: the clock speed of the CPU was 10 MHz. The attitude and position of the rover was supposed to be calculated in two steps. The solar direction is measured after the robot awakes and before the robot falls into sleep and it is used to calculate the rover attitude with reference to the small body. The attitude is then reconstructed by gyroscope signal integration. Then surface images of the asteroid, captured during rover hopping, are used to estimate the hop velocity and gravity, these value were used to estimate absolute and discrete localization. The velocity was estimated using the optical flow, and the distance from the comet by considering rover's own shadow in the acquired image.

Of great interest is the navigation framework developed by [29, 30] for the navigation of the ESA Rosetta spacecraft in proximity of comet 67P/ Churyumov-Gerasimenko. Navigation phases that have seen the use of optical measurements have been the approach, mapping and characterization, landing and comet escort phases. The cometary phase of the navigation has been characterized form 100 km to 50 km distance by hyperbolics arcs in pyramidal shape, and form 30 km to 10 km by circular orbits, moreover Rosetta has performed several

flybys around the comet. The maplet technique has been used to automatically detect comet landmark in order to estimate the spacecraft relative pose.

During the initial characterization phase landmarks were placed by an human operator on NAVCAM comet images, landmarks observations were computed on ground. After few weeks of operations landmark detection techniques passed in automatic mode on order to augment the landmark grid. Mission budget limits the frequency of acquisition of navigation images to one image each hour. This made impossible to use classical feature detectors, like corner or blob detectors, because after one hour the comet appearance might be completely different. To overcome the appearance related problems, maplets technique has been used. Maplets are small scale 3D high resolution maps generated by means of stereophotoclinometry techniques. Stereophotoclinometry consists on the conversion of relief shadows into slopes and then into height maps. Then the rectified images are inverted into slopes and albedo maps using a photometric model. Finally the slopes are integrated into a height map. A maplet is the information union between an height and an albedo. For each maplet it is associated a distance of observation. Maplet landmarks are use to reduce the orbit attitude and prediction errors, as first approximation the error correspond to a shift in x and y position of the expected position of the landmark. Using orbit and attitude prediction model the expected image is build using maplet database, the true image and the simulated one are cross correlated and the pose estimation error computed and corrected. Thanks to this computational expensive technique it is possible to achieve sub-pixel accuracy in the landmark pixel coordinates. These coordinates are converted into the spacecraft attitude.

Table 1.1 Rosetta camera characteristics, as reported by[31].

Parameter	**NAVCAM**	**OSIRIS/NAC**	**OSIRIS/WAC**
Optics type	7 lenses, 3 filters	3 mirrors off-axis, dual filter wheel	2 mirror off-axis, dual filter wheel
Aperture diameter	7 cm	9 cm	2.5 cm
Field of view	$5^\circ \times 5^\circ$	$2.20^\circ \times 2.22^\circ$	$11.34^\circ \times 12.11^\circ$
Focal length	152.5 mm	717.4 mm	136 mm
CCD	1024×1024 pixels	2048×2048 pixels	2048×2048 pixels
Signal resolution	12 bit	16 bit	16 bit

1.2.1 Hedgehog: JPL/Stanford Hopping/Tumbling Platform

In order to overcome the challenges posed by the asteroid/cometary environments, Jet Propulsion Laboratory together with Stanford University are proposing a new mobility concept,

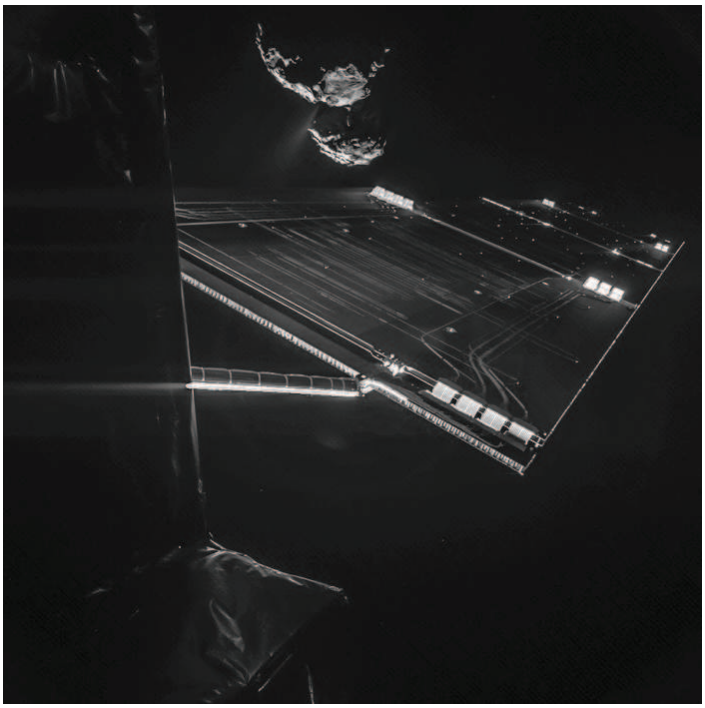


Fig. 1.8 Rosetta mission selfie at 16 km. Credit: ESA/Rosetta/Philae/CIVA.

called “Hedgehog”. The novelty of the project consists to develop a mission architecture based on the spacecraft/rover hybrid concept. Spacecraft/rover hybrids are characterized by small size and low power consumption, less than 5 kg for 15 Watts. The actual prototype is equivalent to an 8U design and is designed for mobility in low gravity environments (micro-g and milli-g). In the paper of [32] we can find a detailed contact dynamical analysis of the JPL platform prototype. The dynamics and control of tumbling/hopping platforms, and key design features (e.g., flywheel design and orientation, geometry of external spikes, and system engineering aspects) are discussed in [33]. The mission architecture is highlighted in [34].

Thanks to attitude-controlled hops, the rover hybrid is capable to achieving large surface coverage, by means of tumbling, it is able to achieve fine mobility and coarse instrument pointing is given by changing orientation relative to the ground. In Figure 1.9 we can see the JPL prototype of the rover, enclosing inside the cube three mutually orthogonal flywheels surrounded by external spikes give the manoeuvre capabilities. External propulsion is not needed, the mobility is given by accelerating and decelerating the flywheels. The mother spacecraft will be needed for communication purposes and as a support for the rover hybrid localization as we will see in Chapter 4.

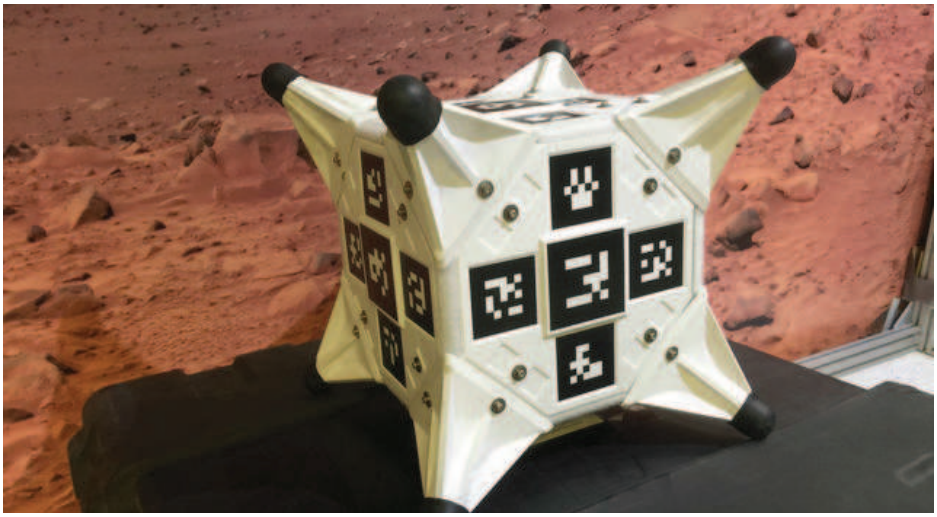


Fig. 1.9 Hedgehog gets around by spinning and stopping three internal flywheels using motors and brakes. Credit: NASA/JPL-Caltech/Stanford.

Chapter 2

Machine Vision Algorithms

The analyzed machine vision algorithms are presented in this chapter. The chapter is divided in two parts, a first shows the state of the art of Visual Odometry algorithms, the basis and the three different families od motion estimation techniques: 3D-to-3D, 3D-to-2D and 2D-to-2D. Moreover, the basis of Bundle Adjustment are give. The second part of the chapter analyses Visual SLAM (Simultaneous Localization and Mapping) algorithms and explains how ORB-SLAM2 works. ORB-SLAM2 is the software that has been modified to solve the synergistic localization and mapping problem between a mother and a daughter spacecraft.

As we have seen in Chapter 1 in GPS denied environment, like Mars or other solar system bodies, Visual Odometry, visual SLAM and more in general geometric vision is an asset to resolve robots egomotion problem. Geometric vision is a subdomain of computer vision that deals with the mathematical process behind the image formation given a three dimensional scene. Thanks to the optical projection theory, given camera parameters and its position relative to an object, it is possible to describe the object formation in the image space. In other word it exist a projection function which maps the three dimensional space to a two dimensional space. Geometric vision has a wide range of applications like camera egomotion tracking, camera calibration, object tracking, geometric consistency test in place recognition, large-scale reconstruction and many others. The interested reader is referred to the works of [35] and the dissertation of [36].

Figure 2.1 shows the structure from motion (SfM) problem, given a set of images taken at different positions and orientations it is possible to reconstruct the three dimensional environment and camera relative positions. In structure from motion problems images are unordered and the three dimensional environment is computed simultaneously. The fundamental building blocks of SfM is the computation of the relative pose between two view-points, it is then possible to retrieve a first information about the three dimensional scene and then to compute the absolute and relative pose of the other viewpoints. Considering

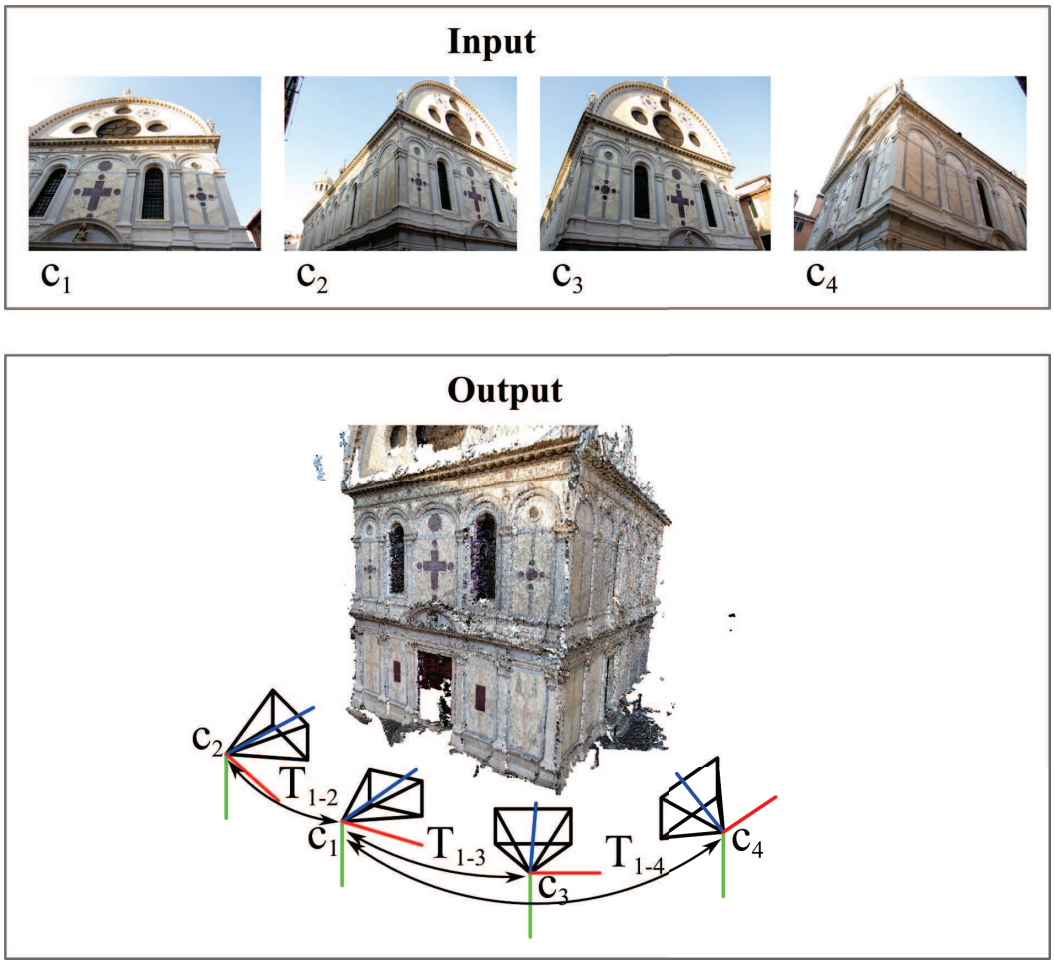


Fig. 2.1 The Structure from Motion problem

the unordered sequence of images a challenging part of the SfM problems is to firstly identify images that observes the same part of the scene. [37] shows the possibility to reconstruct entire cities from an extremely large set of photos, the sequence is composed of 150K images. [38] reconstructs the three dimensional model of the asteroid Lutetia with the images captured with the OSIRIS NAC telescope. Furthermore, SfM is widely use also in other applications like the archaeological research [39].

We can consider that Visual Odometry is a subset of SfM problem, where the relative pose computation between two view-points (the SfM building block) is applied sequentially to a temporally ordered series of images, the incremental transformations between ordered camera frames are computed in order to reconstruct camera trajectory. Visual Odometry gains interest in robotic applications with the reduction of the processing time. A tutorial on Visual Odometry is presented in [40] and [41].

Simultaneous localization and mapping (SLAM) attempts to solve the problem of a robot placed at an unknown environment, the robot incrementally build a map and localize itself relative to the map that he is building. The first SLAM algorithms uses laser scanners for online perception of the environment and to tracking robot motion. Mostly SLAM approaches are realized thanks a filter-based solution, like the Extended Kalman Filter, the Particle Filter or the Graph-Based SLAM. Recently the filter-based SLAM paradigm has been ported to monocular and stereo camera, we talk then about visual SLAM. Thanks to parallel computing, the state of the art of visual SLAM algorithm foreseen an egomotion estimation and a parallel geometric optimization in the background, like a Local or a Global Bundle Adjustment. It is possible to pose other constraints on the egomotion estimation like a motion model or the integration with other sensor measurements. [42] gives a tutorial about SLAM problems.

2.1 Camera Model

The pinhole camera model is fundamental for the most of the geometric vison problem discussed in this work. The model is developed for CCD like sensor and it describes the central projection of 3D points through the centre of projection onto the image plane.

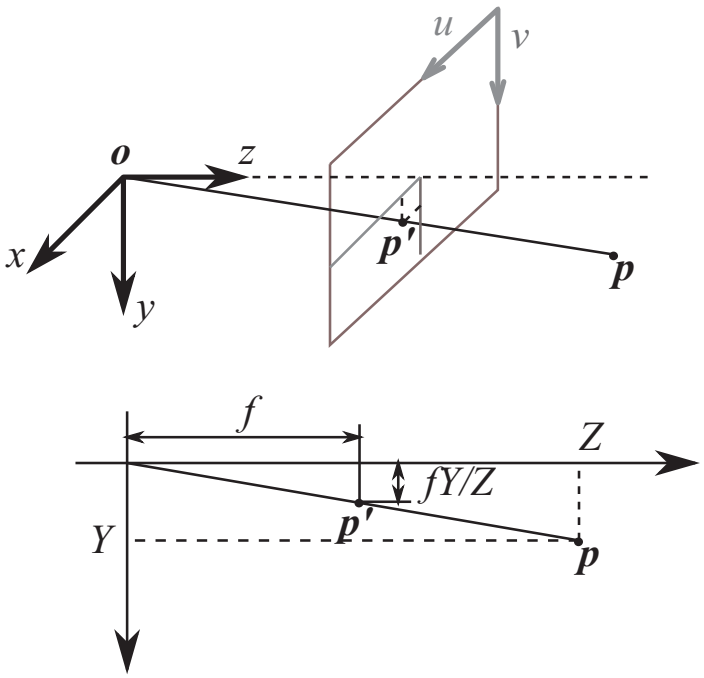


Fig. 2.2 Pinhole camera model. **o** is the camera centre and $\mathbf{p}'$ the projection on the image plane of the point **p**.

Let the pinhole camera centre being the origin of a reference frame (O,x,y,z), the z-axis is the same as the camera pointing direction, the image plane or focal plane is the plane $z = f$, see Figure 2.2. The line from the camera centre perpendicular to the image plane is called the principal axis or principal ray of the camera, and the point where the principal axis meets the image plane is called the principal point. A point in the 3D space $\mathbf{p} = (p_x, p_y, p_z)$ is mapped to the image plane where there is the intersection between the line joining the point $\mathbf{p}$ to the centre of the camera (u,v). Considering the principal point offset (c_x, c_y) the projection equation is:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f_x \frac{p_x}{p_z} + c_x \\ f_y \frac{p_y}{p_z} + c_y \end{pmatrix} \tag{2.1}$$

where f_x and f_y are the camera's focal lengths in the x and y directions. In the example Figure 2.2 $f = f_x = f_y$ is considered. Equation 2.1 can be written in terms of matrix multiplication as:

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \tag{2.2}$$

where $s \in \mathbb{R}_+$ is an arbitrary positive scalar associated with the depth of the point. During the projection the scale information is lost; it means that the projection of all the points that belong to the line from the point $\mathbf{p}$ to the camera centre is the image plane point (u,v). This is a mapping from Euclidean 3-space $\mathbb{R}^3$ to Euclidean 2-space $\mathbb{R}^2$. The focal length and the principal point offset are the *intrinsic parameters* of the camera, they can be estimated through the camera calibration process. They does not depends from the scene viewed, and if the camera focal length is fixed, they can be reused for all the image sequence.

The camera motion around the scene is described by the joint rotation-translation matrix $[\mathbf{R}|\mathbf{t}]$, or camera *extrinsic parameters*. The joint rotation-translation matrix could also express the rotation and translation of an object around a fixed camera. Considering the extrinsic camera parameters we obtain the *projection function*:

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} \tag{2.3}$$

Lens Distortion Real lenses present distortions that have to be considered in the projection model. Distortions arise mainly form the lens shape, we talk then about *radial distortion*, and from CCD alignment respect to the optical axis, we talk in this case about *tangential*

distortion. In this work we adopt the polynomial distortion model, the same used in the OpenCV library [43]. In literature it is possible to find other distortion models, like the FOV-model [44] suited for fish-eyes lenses and wide FOV cameras. The polynomial distortion model limited to the 3rd order, with the tangential distortion is given by :

$$\begin{aligned} x_d &= x_u(1+k_1r^2+k_2r^4+k_3r^3)+2p_1x_uy_u+p_2\left(r^2+2x_u^2\right) \\ y_d &= y_u(1+k_1r^2+k_2r^4+k_3r^3)+2p_2x_uy_u+p_1\left(r^2+2y_u^2\right) \end{aligned} \tag{2.4}$$

where $r^2 = x_u^2 + y_u^2$ and $x_u = p_x/p_z \quad y_u = p_y/p_z$. k_1, k_2, k_3 are radial distortion coefficients, p_1 and p_2 are tangential distortion coefficients.

In this work all the intrinsic and extrinsic camera parameters are calculated using Zhang calibration method [45].

2.2 Feature detection and matching

In SfM, VO and feature-based SLAM the first step begins with the identification of image salient regions, this computer vision process is called *feature detection*. For each feature is associated a descriptor. Then the features are matched the one with the others based on some similarity metric between their descriptors.

Corner detectors or *blob detectors* allow a precise measurement of the salient region in images coordinates. The first type of feature, the corner, correspond to image regions that have high intensity gradient in two orthogonal directions, generally is the intersection of one or more edges. The second type of feature, the blobs, correspond to image spot that have an intensity different from the neighbouring regions.

The attractive properties for a feature detector are listed in Tab. 2.1. A feature should be redetected in the next images that look to the same scene, this property is the detector repeatability. We talk about robustness when the detector it is not sensitive to image noise. Furthermore a feature should be detected after images photometric and geometric changes, like rotations, scale and affine transformations.

In the choice of a feature detector, localization accuracy, robustness and invariance properties have to be weighed against computational efficiency. For example, if the purpose of our application is to reconstruct a detailed three dimensional topographic map with sparse images the feature detector should be accurate and invariant to geometric changes. Instead it the rationale is the computational time we should opt for the efficiency against other performances. The last case is the case of many robotics application, where the lower

Table 2.1 Comparison of feature detectors: properties and performance, data from [41].

	Corner Detector	Blob Detector	Rotation Invariant	Scale Invariant	Affine Invariant	Repeatability	Localization Accuracy	Robustness	Efficiency
Haris	×		×			+++	+++	++	++
Shi-Tomasi	×		×			+++	+++	++	++
FAST	×		×	×		++	++	++	++++
SIFT		×	×	×	×	+++	++	+++	+
SURF		×	×	×	×	+++	++	++	++
CENSURE		×	×	×	×	+++	++	+++	+++

performances of the detector are compensated with smarter feature searching techniques, like reducing the searching window by using camera motion priors.

Among the corner detectors we count Harris [46], Shi-Tomasi [47] and Features from Accelerated Segment Test (FAST) [48]. Among the blob detectors we count Scale Invariant Feature Transform (SIFT) [49], Speeded Up Robust Feature (SURF) [50], CENter SUrround Extremas (CENSURE) [51] and many others.

After feature extraction we need to establish correspondence between different image features, we pass through the *feature descriptors* extraction phase for each local invariant keypoint. The information about feature surroundings is embedded in a floating point or binary value vector. One of the methods used to find the feature correspondence between two images, consists in the calculation of the vectorial distance between all the image descriptor, the smalles $L2$-norm corresponds to a feature matching. In the binary case we compute the Hamming distance. Two descriptors that provide the properties of scale and rotation invariance are the SIFT and SURF descriptors. One of the most computationally efficient descriptor is the Binary Robust Independent Elementary Features (BRIEF) [52], which uses binary string to represent the kepoint surroundings. The Binary Robust invariant scalable keypoints (BRISK) [53] combine the efficiency of a binary descriptor with the rotation invariant properties.

The Visual Odometry study presented in Section 3.3 have been realized as verification tool for the rovers ego-motion estimation, the algorithm rationale is the accuracy for this

reason the highest number of points is suited for relative pose measurement. The Harris detector the SIFT detector have been used. In the ORB-SLAM framework developed by [54], which has been used and modified in Chapter 4, Oriented FAST and Rotated BRIEF (ORB) [55] descriptor has been employed.

2.3 Visual Odometry

Visual Odometry is the process of reconstructing vehicle trajectory and attitude by processing images of an single or a multiple camera attached to it. VO can find application in ego-motion estimation, where the camera's motion is estimated incrementally with reference to its initial position. To ensure a low error in the final trajectory reconstruction, the incremental error of the step motion estimation errors have to been kept as small as possible, as it accrues rapidly.

VO, as has been highlighted in the introduction chapter, compared to wheel odometry is not affected by vehicle slippage on the soil, it can be used as integration or substitution of a GPS receiver, or an IMU. Its importance increases for navigation system in GPS denied environments. As reported by [40] a great part of modern VO system have a relative position error ranging between 0.1 and 2%.

Nowadays VO has a wide range of application from the UAV navigation to the underwater vehicle passing thorough the autonomous driving cars, without forgetting the Mars rovers! One of the first work about estimating the motion of a vehicle with images is the work of Moravec [56], and most of the earlier research was done with stereo systems for the NASA Mars Exploration program [57–59]. One of the first real time VO system s was introduced by Nistér [60].

Figure 2.3 shows a general scheme for a VO system, it is possible to identify three main VO family: for monocular cameras, stereo cameras or multi-camera (an example is given in [61]).

Stereo VO Most of the Stereo VO methods have in common that for each stereo pair the point cloud of the triangulated features is calculated, the motion estimated with a 3D-to-3D point cloud registration. In this case for each stereo pair the feature matching is performed intra stero-pair and inter stero-pair, in other word the same feature is matched in the four available images. Bu the stereo VO methods are not only limited to a 3D-to-3D estimation. Nistér et al. [60] estimated the relative motion using a 3D-to-2D camera-pose estimation problem. Moreover, they incorporated RANdom SAmple Consensus (RANSAC) outlier rejection into the motion estimation step. In Comport et al. [62] the motion estimation relies on 2D-to-2D images matches, this method avoids the triangulation of the 3D points.

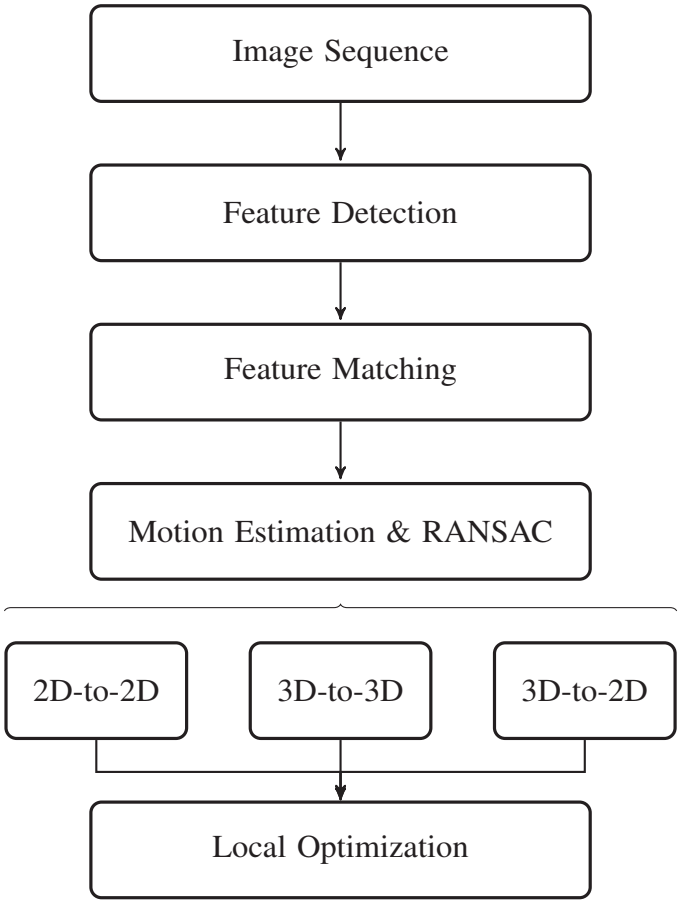


Fig. 2.3 Generalized block diagram showing the main components of a VO system. There are three relative pose estimation methods: 2D-to-2D, 3D-to-3D, or 3D-to-2D.

Monocular VO In monocular systems the absolute scale is unknown for the three dimensional scene and the trajectory. Usually the distance between the two first images is set to the unity, as consequence everything is scaled to this unit distance. As suggested by Mur-Artal et al. [54], if the scene is planar dominant, the relative transformation between the two images can be described by a homography $\mathbf{p}_c = \mathbf{H}_{cr}\mathbf{p}_r$ and they suggest to use the normalized DLT algorithm [35] to retrieve the homography matrix $\mathbf{H}_{cr}$. On the other hand a non-planar scene with enough parallax can be described by the fundamental matrix $\mathbf{p}_c^\top \mathbf{F}_{cr}\mathbf{p}_r = 0$, which is possible to retrieve thanks to the 8-point algorithms [35]. As a new image arrives its pose relative to the first two frames could be computed by exploiting the knowledge of the three dimensional structure, in this case it is necessary first to triangulate the scene points, or using the trifocal tensor [35].

2.3.1 Motion Estimation

As we have seen, in VO there are three categories of camera absolute and relative pose estimation: 3D-to-3D, 3D-to-3D, 2D-to-2D. If we have a set of images taken at a discrete time k, in the monocular case the sequence is notes as $I_{0:n} = \{I_0, \ldots, I_n\}$. If we analyse the sequence of a stereo camera we have the left and the right images at the instant k: $I_{l,0:n} = \{I_{l,0}, \ldots, I_{l,n}\}$ and $I_{r,0:n} = \{I_{r,0}, \ldots, I_{r,n}\}$.

The rigid body transformation in $\mathbb{R}^3$ between two cameras at two different instants k and $k-1$ is given by a 4×4 matrix of the following form:

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & \mathbf{t}_{k,k-1} \\ 1 & 0 \end{bmatrix} \tag{2.5}$$

where $R_{k,k-1} \in SO(3)$ is the rotation matrix and $\mathbf{t}_{k,k-1}$ the translation vector. $SO(3)$ being the Lie group of rotation matrices and $T_{k,k-1} \in SE(3)$ is a *Special Euclidean* group. The vector of poses $T_{1:n} = \{T_{1,0}, \ldots, T_{n,n-1}\}$ contains all the step poses. If we want to retrieve the camera pose relative to the initial frame we have to concatenate all the poses, for example $T_{n,0} = \prod_{k=1}^{k=n} T_{k,k-1}$. All the trajectory is contained in the vector $C_{1:n} = \{T_{1,0}, T_{2,0}, \ldots, T_{n,0}\}$

3D-to-3D In this case, both f_{k-1} and f_k are specified in 3-D. To triangulate the 3D points it is necessary a stereo camera system. The motion estimation is obtained by the minimization of the L_2 distance between the two feature sets, both specified as a 3D position:

$$\underset{T_{k,k-1}}{\operatorname{argmin}} \sum_i \left\| \widetilde{\mathbf{p}}_k^i - T_{k,k-1}\widetilde{\mathbf{p}}_{k-1}^i \right\| \tag{2.6}$$

where $\widetilde{\mathbf{p}}^i_{k-1}$ are the homogeneous coordinates ($\widetilde{\mathbf{p}} = (p_x, p_y, p_z, 1)^\top$) of the 3D point correspondent to the feature i A solution to this minimization problem is given by the Arun's method for aligning point clouds [63–65]. The minimal-case solution involves 3 non-collinear correspondences.

3D-to-2D The pose is estimated by minimizing the re-projection error of the triangulated 3D points [66, 67]:

$$\underset{T_{k,k-1}}{\operatorname{argmin}} \sum_i \left\| \mathbf{u}^i_k - \pi(\mathbf{p}^i_{k-1}, T_{k,k-1}) \right\|^2 \tag{2.7}$$

where $\pi(\mathbf{p}^i_{k-1}, T_{k,k-1})$ is the reprojection of the 3D point $\mathbf{p}^i_{k-1}$ into the image I_k through the transformation $T_{k,k-1}$. $\mathbf{p}^i_{k-1}$ can be estimated from stereo data or, in the monocular case, from triangulation of the image measurements $\mathbf{u}^i_{k-1}$ and $\mathbf{u}^i_{k-2}$. The minimal-case solution involves 3-point correspondences. One of the most used solution is the PnP algorithm proposed by Lepetit et al. [68]., which is a non-iterative solution whose computational complexity grows linearly with n, $n \geq 4$. Widely used is also the *Perspective-Three-Point* (P3P) solution of Kneip et al. [67], which aims at determining the position and orientation of the camera from three point correspondences. by using a RANSAC scheme for robust motion estimation in presence of outliers, to a lower number of points needed for the minimal case correspond a lower number of RANSAC iterations and a more efficient implementation of the VO system.

2D-to-2D Both the feature f_k and f_{k-1} are expressed in the 2D image space. The rotation and translation estimation between two subsequent frames pass trough the essential matrix E estimation. The minimal case solution involves five correspondences, an efficient implementation is proposed by Nistér in [66].

2.3.2 RANdom SAmple And Consensus

As we can see form the general scheme of Figure 2.3, the motion estimation process is embedded in a RANSAC scheme. In computer vision literature RANSAC is became a standard to fitting a model to experimental data in presence of a great part of outliers. Indeed, the probability of an erroneous feature correspondence between two images it is elevated. RANSAC was introduced by Fischler and Bolles in [69] in order to determine the absolute pose of a camera given an image depicting a set of landmarks with known locations. The different steps involved in a RANSAC scheme may be summarized as follow:

- Select randomly a minimum set of points required s to determine the model parameters. For example in the for the PnP algorithm of the 3D-to-2D case the minimal set is four points. These points are needed in order to compute a motion hypothesis $T_{k-1,k}$.

- Compute a motion hypothesis $T_{k,k-1}$.
- Project all the set of n points using the motion hypothesis $T_{k,k-1}$ and compute the error for each point $e^i_{k-1,k} = \left\| \mathbf{u}^i_k - \pi(\mathbf{p}^i_{k-1}, T_{k,k-1}) \right\|$. Verification of the motion hypothesis by the determination of how many points from the set of all points fit with a predefined tolerance $e^i_{k-1,k} < \varepsilon$. For a 3D-to-3D method the 3D error for each point has to be computed.
- If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold τ, re-estimate the model parameters using all the identified inliers and terminate.
- Otherwise, repeat all the steps for maximum of N times.

[69] gives also the minimum number of iterations N that it is necessary to find the correct solution:

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)} \tag{2.8}$$

where p is the required probability of success, s the minimum set of point required by the model and ε is an estimation of the point set outlier percentage. As we can see greater is s greater will be the number of iterations N required for the same values of p and ε.

2.3.3 Local Optimization

The final of a VO algorithm consists into optimize the first estimation of $T_{k,k-1}$, obtained through the RANSAC procedure, using all the n inliers. Many computer vision problems are non linear, in VO pose refinement could be performed through the Levemberg-Marquardt algorithm.

In the Stereo Visual Odometry system proposed in Chapter 3.3 the pose refinement is computed though the minimization of the the L_2 distance between the triangulated 3D points (3D-to-3D implementation). Equation 2.11 shows the non-linear cost function E_{nl}. We can see that each component of the error vector $\mathbf{e_i}$ of feature i is weighted taking into account landmark uncertainty. The uncertainties of the 3D points are represented by 3×3 covariance matrices as calculated by the Kline-McClintock formula, see GUM [70].

$$\mathbf{e}^i = \mathbf{p}^i_k - R_{k,k-1}\mathbf{p}^i_{k-1} - \mathbf{t}_{k,k-1} \tag{2.9}$$

$$\Omega^i = \Omega^i_k + R_{k,k-1}\Omega^i_{k-1}R^\top_{k,k-1} \tag{2.10}$$

$$E_{nl} = \sum_{i=1}^{n} (\mathbf{e}^{i\top} \Omega^{i\,-1} \mathbf{e}^{i}) \tag{2.11}$$

$\mathbf{p}_k^i$ are the three dimensional coordinates of the landmark i, expressed relative to the stereo camera during step k, $\mathbf{p}_{k-1}^i$ is the same 3D point expressed relative to the stereo camera during at the previous location $k-1$. $R_{k,k-1} \in SO(3)$ is the rotation matrix between the two positions and $\mathbf{t}_{k,k-1}$ is the system translation expressed in the frame of reference k-1; finally, Ω_k^i and Ω_{k-1}^i are the 3×3 covariance matrices of the same landmark in the two frames of reference.

In the monocular case is possible to refine the camera pose by minimizing the reprojection error $\mathbf{e}^i$.

$$\mathbf{e}^i = \mathbf{u}_k^i - \pi(\mathbf{p}_{k-1}^i, T_{k,k-1}) \tag{2.12}$$

where $\pi()$ is the projection function in Eq. 2.3. The cost function to be minimized is:

$$E_{nl} = \sum_{i=1}^{n} (\mathbf{e}^{i\top} \Omega^{i\,-1} \mathbf{e}^{i}) \tag{2.13}$$

where $\Omega^i = \sigma_i^2 \mathbf{I}_{2\times2}$ is the covariance matrix associated to the uncertainty of the feature in the 2D image. This minimization problem is the base of the localization framework of Chapter 4. In ORB-SLAM2 [54] Ω^i is the covariance matrix associated to the scale at which the feature is detected, and the Huber robust cost function is used.

In VO all the trajectory is given by the concatenation of all the step poses. Each estimation of the relative pose transformation $T_{k,k-1}$ has an uncertainty, the uncertainty of the pose estimation of the last frame in the initial reference frame is give by the propagation of all the step uncertainties. As shown by [41], the camera-pose uncertainty is always increasing when concatenating transformations. As follows, it is important to keep the step uncertainty as small as possible to reduce the drift.

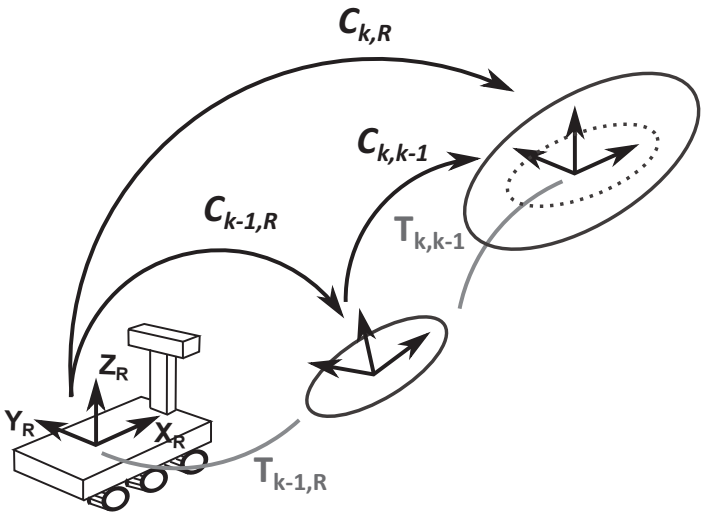


Fig. 2.4 The uncertainty of the pose estimation drifts over the time. The final uncertainty is a combination of the step uncertainty $C_{k,k-1}$ and the actual uncertainty $C_{k-,R}$.

2.4 Bundle Adjustment

A map point could be observed by multiple frames. Thus it is possible to set up a large optimization problem which attempts to optimize all the camera poses and the 3D points position, we talk about Global Bundle Adjustment. If we optimize only the poses of the frames that seen las frame 3D points we talk then about Local Bundle Adjustment. In the case that we would like to optimize m camera poses and n 3D points, the optimization step involve the computation of $2mn \times (6m+3n)$ Jacobian matrices, by considering the 6DOF $(t_x, t_y, t_z, \phi, \theta, \psi)$ for the camera pose and 3DOF (p_x, p_y, p_z) for the 3D points. The computational cost become very expensive with the number of frame and points. Moreover, do to the non linear nature of the problem the solution could easily converge to a local minima, so good initial conditions are needed. Bundle adjustment is used to refine the initial camera and structure parameters.

Assume that n 3D points are seen in m views. $\mathbf{u}_{i,j}$ is the 3D point projection of point i in the camera j. Each camera j is parametrized by a vector $\mathbf{a}_j$ and each 3D point i by a vector $\mathbf{b}_i$. $v_{i,j} = 1$ if the point i is visible in j view.

$$\underset{\mathbf{a}_i, \mathbf{b}_j}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{m} v_{i,j} \left\| \mathbf{u}_{i,j} - \pi(\mathbf{a}_j, \mathbf{b}_i) \right\|^2 \tag{2.14}$$

where $\pi(\mathbf{a}_j, \mathbf{b}_i)$ is the projection of the point i to the image j.

In $\mathbb{R}^3$, one possible parametrization for $\mathbf{a}_i$ is the 6-vectors $(\boldsymbol{\omega}, \boldsymbol{v})$, where $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ is the axis-angle representation of the rotation and $\boldsymbol{v}$ is the rotated version of the translation

t. Elements of $(\boldsymbol{\omega}, \boldsymbol{v})$ can be mapped to the $SE(3)$ group using the exponential mapping $\exp_{\mathrm{SE}(3)}$ (for further details see [71]):

$$\exp_{\mathrm{SE}(3)}(\boldsymbol{\omega}, \boldsymbol{v}) = \begin{bmatrix} \exp_{\mathrm{SO}(3)}(\boldsymbol{\omega}) & V\boldsymbol{v} \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} R & \mathbf{t} \\ 1 & 0 \end{bmatrix} \tag{2.15}$$

where $\exp_{\mathrm{SO}(3)}(\boldsymbol{\omega}) = I + \frac{\sin\theta}{\theta}(\boldsymbol{\omega})_\times + \frac{1-\cos\theta}{\theta^2}(\boldsymbol{\omega})_\times^2$, $V = I + \frac{1-\cos\theta}{\theta^2}(\boldsymbol{\omega})_\times + \frac{\theta-\sin\theta}{\theta^3}(\boldsymbol{\omega})_\times^2$ and $\theta = \|\omega\|_2$.

This large optimization problems could be solved using the Newton-Raphson or Levenberg-Marquart (LM) optimization methods. In the next section we will see how the LM algorithm works. Let h be the measurement function which maps a parameter vector $\mathbf{p}$ to an estimated measurement vector $\widehat{\mathbf{x}} = h(\mathbf{p})$. If we have an initial parameter $\mathbf{p}_0$ and a measured vector $\mathbf{x}$ are provided and it is desired to find the vector $\mathbf{p}^+$ that best satisfies the function h locally, that is, minimizes the squared distance $\varepsilon^\top \varepsilon$ with $\varepsilon = \mathbf{x} - \widehat{\mathbf{x}}$ the residual error. The basis of the LM algorithm is an affine approximation to h in the neighbourhood of $\mathbf{p}$. For a small $\|\delta\mathbf{p}\|$, h is approximated by:

$$h(\mathbf{p} + \delta\mathbf{p}) \approx h(\mathbf{p}) + J\delta\mathbf{p} \tag{2.16}$$

where J is the Jacobian of h. Like all nonlinear optimization methods, LM is iterative, initiated at the starting point $\mathbf{p}_0$, the vector parameter $\mathbf{p}$ is updated with the rule:

$$\mathbf{p}_l = \mathbf{p}_{l+1} + \delta\mathbf{p} \tag{2.17}$$

that converge toward a local minimum value $\mathbf{p}^+$ for h. Hence, at each iteration, it is required to find the step $\delta\mathbf{p}$ that minimizes the quantity

$$\|\mathbf{x} - h(\mathbf{p} + \delta\mathbf{p})\| \approx \|\mathbf{x} - h(\mathbf{p} - J\delta\mathbf{p}\| = \|\varepsilon - J\delta\mathbf{p}\| \tag{2.18}$$

The sought $\delta\mathbf{p}$ is thus the solution to a linear least-squares problem: the minimum is attained when $J\delta\mathbf{p} - \varepsilon$ is orthogonal to the column space of J. This leads to $J^\top(J\delta\mathbf{p} - \varepsilon) = 0$, which yields $\delta\mathbf{p}$ as the solution of the so-called normal equations:

$$J^\top J\delta\mathbf{p} = J^\top \varepsilon \tag{2.19}$$

where $J^\top J$ is the first order approximation to the Hessian of $\frac{1}{2}\varepsilon^\top\varepsilon$, $\delta\mathbf{p}$ is the Gauss-Newton step and $J = \frac{dh}{d\mathbf{p}}$. The LM method actually solves a slight variation of Equation 2.19:

$$N\delta\mathbf{p} = J^\top\varepsilon \tag{2.20}$$

where $N = J^\top J + \mu I$, $J^\top J$ is also called the Hessian matrix and $\mu > 0$ is the damping term.

We going to apply the example proposed by [72] to the Bundle Adjustment of a 3D-to-2D problem, the same example explains the local and the global bundle adjustment in ORB-SLAM [54]. Assume that $n = 4$ points are visible in $m = 3$ different views taken by the same cameras. We are supposing that feature 1 is seen by cameras 1, 2 and 3, feature 2 by 1 and 2, feature 3 by cameras 1, 2, 3, feature 4 by cameras 2 and 3. Thus, the measurement vector is $\mathbf{u} = (\mathbf{u}_{11}, \mathbf{u}_{12}, \mathbf{u}_{13}, \mathbf{u}_{21}, \mathbf{u}_{22}, \mathbf{u}_{31}, \mathbf{u}_{32}, \mathbf{u}_{33}, \mathbf{u}_{42}, \mathbf{u}_{43})^\top$ and the parameter vector is given by $\mathbf{p} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_3)^\top$. We can write the Jacobian as:

$$\frac{dh(\mathbf{p})}{d\mathbf{p}} = \begin{bmatrix}
\frac{d\pi(\mathbf{a}_1,\mathbf{b}_1)}{d\mathbf{a}_1} & 0 & 0 & \frac{d\pi(\mathbf{a}_1,\mathbf{b}_1)}{d\mathbf{b}_1} & 0 & 0 & 0 \\
0 & \frac{d\pi(\mathbf{a}_2,\mathbf{b}_1)}{d\mathbf{a}_2} & 0 & \frac{d\pi(\mathbf{a}_2,\mathbf{b}_1)}{d\mathbf{b}_1} & 0 & 0 & 0 \\
0 & 0 & \frac{d\pi(\mathbf{a}_3,\mathbf{b}_1)}{d\mathbf{a}_3} & \frac{d\pi(\mathbf{a}_3,\mathbf{b}_1)}{d\mathbf{b}_1} & 0 & 0 & 0 \\
\frac{d\pi(\mathbf{a}_1,\mathbf{b}_2)}{d\mathbf{a}_1} & 0 & 0 & 0 & \frac{d\pi(\mathbf{a}_1,\mathbf{b}_2)}{d\mathbf{b}_2} & 0 & 0 \\
0 & \frac{d\pi(\mathbf{a}_2,\mathbf{b}_2)}{d\mathbf{a}_2} & 0 & 0 & \frac{d\pi(\mathbf{a}_2,\mathbf{b}_2)}{d\mathbf{b}_2} & 0 & 0 \\
\frac{d\pi(\mathbf{a}_1,\mathbf{b}_3)}{d\mathbf{a}_1} & 0 & 0 & 0 & 0 & \frac{d\pi(\mathbf{a}_1,\mathbf{b}_3)}{d\mathbf{b}_3} & 0 \\
0 & \frac{d\pi(\mathbf{a}_2,\mathbf{b}_3)}{d\mathbf{a}_2} & 0 & 0 & 0 & \frac{d\pi(\mathbf{a}_2,\mathbf{b}_3)}{d\mathbf{b}_3} & 0 \\
0 & 0 & \frac{d\pi(\mathbf{a}_3,\mathbf{b}_3)}{d\mathbf{a}_3} & 0 & 0 & \frac{d\pi(\mathbf{a}_3,\mathbf{b}_3)}{d\mathbf{b}_3} & 0 \\
0 & \frac{d\pi(\mathbf{a}_2,\mathbf{b}_4)}{d\mathbf{a}_2} & 0 & 0 & 0 & 0 & \frac{d\pi(\mathbf{a}_2,\mathbf{b}_4)}{d\mathbf{b}_4} \\
0 & 0 & \frac{d\pi(\mathbf{a}_3,\mathbf{b}_4)}{d\mathbf{a}_3} & 0 & 0 & 0 & \frac{d\pi(\mathbf{a}_3,\mathbf{b}_4)}{d\mathbf{b}_4}
\end{bmatrix} \tag{2.21}$$

$\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{a}_j}$ is a 2×6 Jacobian matrix and $\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{b}_i}$ a 2×3 matrix. They can be calculated by the chain rule as in [73]:

$$\begin{aligned}
\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{a}_j} &= \frac{\partial\pi(\mathbf{b}')}{\partial\mathbf{b}'}\Big|_{\mathbf{b}'=T_j\mathbf{b}_i}\frac{\partial T\mathbf{b}_i}{\partial T}\Big|_{T_j}\frac{\partial TT_j}{\partial T}\Big|_{I}\frac{\partial \exp_{\mathrm{SE}(3)}(\mathbf{a}_j)}{\partial\mathbf{a}_j}\Big|_{\mathbf{0}} \\
&= \begin{bmatrix}
\frac{f_x}{p_{z,j}} & 0 & -f_x\frac{p_{x,j}}{p_{z,j}^2} & -f_x\frac{p_{x,j}p_{y,j}}{p_{x,j}^2} & f_x\left(1+\frac{p_{x,j}^2}{p_{z,j}^2}\right) & -f_x\frac{p_{y,j}}{p_{z,j}^2} \\
0 & \frac{f_y}{p_{z,j}} & -f_y\frac{p_{y,j}}{p_{z,j}^2} & -f_y\left(1+\frac{p_{y,j}^2}{p_{z,j}^2}\right) & f_y\frac{p_{x,j}p_{y,j}}{p_{x,j}^2} & f_y\frac{p_{x,j}}{p_{z,j}^2}
\end{bmatrix}
\end{aligned} \tag{2.22}$$

$$\begin{aligned}\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{b}_i} &= \frac{\partial\pi(\mathbf{b}')}{\partial\mathbf{b}'}\Big|_{\mathbf{b}'=T_j\mathbf{b}_i}\frac{\partial T\mathbf{b}_i}{\partial\mathbf{b}_i}\Big|_{\mathbf{0}} \\ &= \begin{bmatrix}\frac{f_x}{p_{z,j}} & 0 & -f_x\frac{p_{x,j}}{p_{z,j}^2} \\ 0 & \frac{f_y}{p_{z,j}} & -f_y\frac{p_{y,j}}{p_{z,j}^2}\end{bmatrix}R_i\end{aligned} \tag{2.23}$$

where $(p_x,\ p_y,\ p_z)$ are the components of the vector $\mathbf{b}_i$, and $(p_{x,j},\ p_{y,j},\ p_{z,j})$ are the components of the vector $T_j\mathbf{b}_i$. If we consider the uncertainties of our measurements, that in a 3D-to-2D framework corresponds to the uncertainty of the 2D feature position, Eq. 2.19 becomes:

$$(J^\top\Omega_{ij}^{-1}J+\mu I)\delta\mathbf{p} = J^\top\Omega_{ij}^{-1}\varepsilon \tag{2.24}$$

where $\Omega_{i,j}=\sigma_{i,j}^2 I_{2\times 2}$ is the covariance matrix associated to the features uncertainty. By considering the example of Equation 2.21 the Hessian, given by $(J^\top\Omega_{ij}^{-1}J)$, is:

$$H=\begin{bmatrix}U_1 & 0 & 0 & W_{11} & W_{21} & W_{31} & W_{41} \\ 0 & U_2 & 0 & W_{12} & W_{22} & W_{32} & W_{42} \\ 0 & 0 & U_3 & W_{13} & W_{23} & W_{33} & W_{43} \\ W_{11}^\top & W_{12}^\top & W_{13}^\top & V_1 & 0 & 0 & 0 \\ W_{21}^\top & W_{22}^\top & W_{23}^\top & 0 & V_2 & 0 & 0 \\ W_{31}^\top & W_{32}^\top & W_{33}^\top & 0 & 0 & V_3 & 0 \\ W_{41}^\top & W_{42}^\top & W_{43}^\top & 0 & 0 & 0 & V_4\end{bmatrix} \tag{2.25}$$

where

$$U_j=\sum_{i=1}^{4}\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{a}_j}^\top\Omega_{ij}\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{a}_j} \tag{2.26}$$

$$V_i=\sum_{j=1}^{3}\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{b}_i}^\top\Omega_{ij}\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{b}_i} \tag{2.27}$$

$$W_{ij}=\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{a}_j}^\top\Omega_{ij}\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{b}_i} \tag{2.28}$$

in this particular case $W_{23}=W_{41}=0_{2\times 2}$.

The iterative scheme to solve a minimization problem with the Levenberg-Marquardt algorithm can be summarized this way:

1. Given the initial guess/current state estimate $\mathbf{p}_l$, calculate for each measurement $\varepsilon_{ij,l}=\mathbf{u}_{ij}-h(\mathbf{p}_l)$ and the Jacobian matrix $J_{ij,l}=\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{a}_j}\Big|_l$ or $J_{ij,l}=\frac{d\pi(\mathbf{a}_j,\mathbf{b}_i)}{d\mathbf{b}_i}\Big|_l$ from Equation 2.22 and 2.23.

2. Build the linear system 2.24 with $\boldsymbol{\varepsilon}^\top = \sum_{i,j} J_{ij,l}^\top \Omega_{ij} \varepsilon_{ij,l}$ and $H = \sum_{i,j} J_{ij,l}^\top \Omega_{ij} J_{ij,l}$. Note that the uncertainty of the measurements Ω_{ij} does not depends by the iteration l.

3. Solve the linear system $(J^\top \Omega_{ij}^{-1} J + \mu I)\delta\mathbf{p} = J^\top \Omega_{ij}^{-1} \varepsilon$ for the increment $\delta\mathbf{p}$, update the state $\mathbf{p}_l = \mathbf{p}_{l+1} + \delta\mathbf{p}$ and iterate.

In Levenberg-Marquardt iterations the computation of H^{-1} is computational expensive. Different calculation strategies exist for the inversion of the matrix H. As shown by [72] the nature of this matrix is sparse due to the lack of interaction among certain subgroups of parameters, and this can be exploited to achieve considerable computational savings.

In the framework of Visual SLAM techniques for synergistic localization between the mother and daughter spacecraft, the optimization problems have been solved using the g^{2}o library [71]. Also the Ceres Solver library is widely used by the SfM and robotics community for modelling and solving large, complicated optimization problems [74].

2.5 Simultaneous Localization and Mapping

The simultaneous localization and mapping problem solution is seen by the robotic community the means to make a robot truly autonomous. If we place a robot in an unknown environment, thanks to its sensors the robot is able to reconstruct a consistent map of the environment and a to localize itself with reference to this map. The tutorials of Durrant-Whyte and Bailey [42] and [75] provides a brief history of the early development of SLAM problems, the formulation of the problem and the most common solutions, issues in computation, convergence, and data association. In Cadena et al. [76] we can find a review of the state of the art SLAM codes and the open challenges and the newest research issues in SLAM.

Figure 2.5 shows a graphical model of the SLAM problem, a robot is moving through an environment and is taking relative measurement about the position of a series of unknown landmarks. We want to estimate robot path and landmarks relative positions. The following quantities are considered in the problem: $\mathbf{x}_i$ the robot state of represented by position and attitude at a time instant i and $\mathbf{X}_{0:i} = \{\mathbf{x}_0, \ldots, \mathbf{x}_i\}$ the history of robot state; $\mathbf{u}_i$ the input vector, applied at time instant $i-1$ to drive the robot from the state $\mathbf{x}_{i-1}$ to the state $\mathbf{x}_i$, $\mathbf{U}_{0:i} = \{\mathbf{u}_0, \ldots, \mathbf{u}_i\}$ is the history of inputs; $\mathbf{l}_j$ is the jth landmark, its position is supposed to be time invariant and $\mathbf{l} = \{\mathbf{l}_1, \ldots, \mathbf{l}_j\}$ is the set of all the landmarks; $\mathbf{m}_{i,j}$ is the observation of the jth landmark made from the robot at instant i.

It is possible to reformulate the SLAM problem as an estimation of the posterior probability distribution over the robot's trajectory and the landmark set, given all the measurements

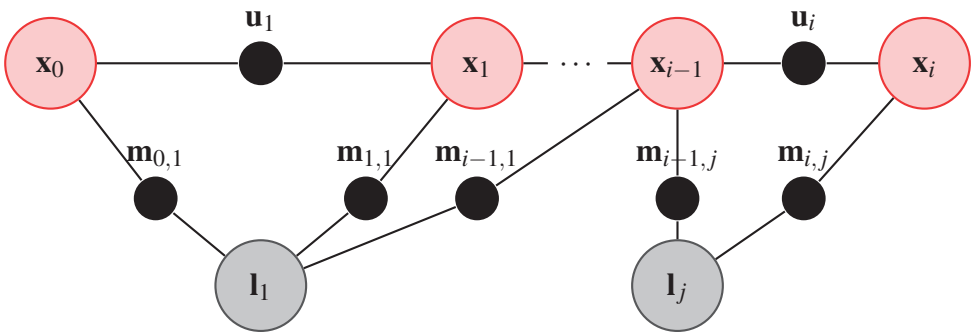


Fig. 2.5 A graphical model of the SLAM problem. $\mathbf{x}_i$ indicate the robot pose, $\mathbf{l}_j$ are landmarks positions directly observable by the robot, $\mathbf{m}_{i,j}$ are the landmarks measurements and $\mathbf{u}_i$ the control vector. Through these quantities, we want to estimate the path of the robot and the landmarks map.

[42, 77], the control inputs and the robot initial state:

$$p(\mathbf{x}_i, \mathbf{l} | \mathbf{M}_{0:i}, \mathbf{U}_{0:i}, \mathbf{x}_0) \tag{2.29}$$

The probability distribution is computed for all times i.

To solve the SLAM problem we have also to introduce a motion model and an observation model. The motion model relies input measurements $\mathbf{u}_i$ to robot state $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$. Supposing that motion model is subject to Gaussian noise, its description in terms of probability distribution is:

$$p(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{u}_i) \sim \mathcal{N}(g(\mathbf{x}_{i-1}, \mathbf{u}_i), R_i) \tag{2.30}$$

It is then defined by a normal distribution centred at $g(\mathbf{x}_{i-1}, \mathbf{u}_i)$, where $g()$ is the kinematic model of robot motion and R_i a 3×3 covariance matrix.

The probability of making an observation $\mathbf{m}_{i,j}$ when the vehicle location and landmark locations are known is generally described in the form:

$$p(\mathbf{m}_{i,j} | \mathbf{x}_i, \mathbf{l}_j) \sim \mathcal{N}(h(\mathbf{x}_i, \mathbf{l}_j), Q) \tag{2.31}$$

where $h(\mathbf{x}_i, \mathbf{l}_j)$ is the measurement function and Q the covariance matrix describing the sensor uncertainty.

[77] identifies three main mathematical framework developed up to date: Extended Kalman Filter (EKF) SLAM, Particle Filter SLAM and the Grpah-Based SLAM.

EKF SLAM

In EKF SLAM the robot estimate is represented by a multivariate Gaussian:

$$p(\mathbf{x}_i, \mathbf{l}|\mathbf{M}_{0:i}, \mathbf{U}_{0:i}, \mathbf{x}_0) \sim \mathcal{N}(\mu_i, \Sigma_i) \tag{2.32}$$

where μ_i is a vector containing robot position and orientation, and the environment landmark positions. By considering a robot moving on a 2D environment, μ_i dimension would be $3+2N$, indeed, we need two variables to define robot position and one variable fo the robot orientation, $2N$ variables for the N landmarks in the map. Σ_i is the robot state covariance matrix, representing the uncertainty in μ_t estimate, it is a $(3+2N) \times (3+2N)$ matrix. Supposing that the $g()$ and $f()$ function are linear in their arguments the Kalman filtering is applicable, $g()$ and $f()$ will be linearised by using the Taylor expansion.

The major issue related to EKF SLAM techniques is the quadratic nature of the covariance matrix. As the robot moves new landmarks are added to the map and then to the state vector, the covariance matrix grows quadratically. This pose a grat limitation also for medium scale maps because the processing time and memory consumption are $O(N^2)$ in the size of the map.

Particle Methods SLAM

In particle methods the posterior probability of robot position is represented by a set od particles. Particle methods were introduced in the SLAM literature by [78] with FastSLAM. For example, FastSLAM maintains each time the robot moves a set of K particles:

$$\mathbf{X}_i^{[k]} \quad \mu_{i,1}^{[k]}, \ldots, \mu_{i,N}^{[k]} \quad \Sigma_{i,1}^{[k]}, \ldots, \Sigma_{i,N}^{[K]} \tag{2.33}$$

k denote the index of the sample. Each particle contains the robot path estimation $\mathbf{X}_i^{[k]}$, an estimation of the the landmarks position $\mu_{i,j}^{[k]}$ with the relative variance $\Sigma_{i,j}^{[K]}$. j is the landmark index ($1 < j < N$).

The initialization is performed by setting each particle at the robot known position, at this state the map is empty. When a new input is given to the robot, or a new odometry reading is received, for each particle new position variables are generated stochastically. New particle positions are generate through the motion model:

$$\mathbf{x}_i^{[k]} \sim p(\mathbf{x}_i|\mathbf{x}_{i-1}^{[k]}) \tag{2.34}$$

For each new measurement $\mathbf{m}_{i,j}$ the filter computes the measurement probability:

$$w_i^{[k]} \sim \mathcal{N}(\mathbf{m}_{i,j}|\mathbf{x}_i^{[k]}, \mu_{i,j}^{[k]}, \Sigma_{i,j}^{[k]}) \tag{2.35}$$

$w_i^{[k]}$ is the importance weight, it measure how important is the particle in the light of the new sensor measurement. The weight is normalized so that the sum of all particle weights is 1. Then the resampling step is performed: a set of new particle is created, the probability of drawing a new particle is based on the normalized importance weight. The landmarks estimate $\mu_{i,j}^{[k]}$ and $\Sigma_{i,j}^{[K]}$ is updated for the new set of particles based on the measurement $\mathbf{m}_{i,j}$ using the Extended Kalman Filter rules. As the robot moves and time passes good particles survive while bad estimates of the state are discarded.

One of the advantage of particle methods over EKF methods is to break down the posterior over maps into low-dimensional Gaussians. Moreover, using tree methods to represent the landmark estimates is possible to improve algorithm efficiency: the update can be performed in $O(log(N))$, logarithmic in the number of landmarks, and linear in the number of particles M, $O(M)$.

Graph-Based SLAM

The SLAM method used in this work, ORB-SLAM, is a Graph-Based method. In these methods landmarks $\mathbf{l}_j$ and robot state $\mathbf{x_i}$ are thought as nodes in a graph. Two consecutive positions $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ are connected by an edge which represents the information obtained by the input kinematics or the odometry reading $\mathbf{u}_i$. Other edges are set up between the robot positions $\mathbf{x}_i$ and the j-th landmark. Figure 2.5 shows a nodes and edges in a SLAM graph. These edges are soft constraints in the graph. The best estimates for the map and the full path is retrieved by relaxing these constraints.

Often the graph is treated as a spring-mass model, to find a solution is equivalent to compute the state of minimal energy of the model. The graph correspond to the log-posterior of the full SLAM problem:

$$\log p(\mathbf{x}_i, \mathbf{l}|\mathbf{M}_{0:i}, \mathbf{U}_{0:i}) = \text{const} + \sum_i \log p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{u}_i) + \sum_{i,j} \log p(\mathbf{m}_{i,j}|\mathbf{x}_i, \mathbf{l}_j) \tag{2.36}$$

where $p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{u}_i)$ are the motion constraints and $p(\mathbf{m}_{i,j}|\mathbf{x}_i, \mathbf{l}_j)$ are the landmark measurements constraints.

Assuming kinematic model and sensors with Gaussian noise we obtain the following quadratic expression:

$$\begin{aligned}\log p(\mathbf{x}_i, \mathbf{l} | \mathbf{M}_{0:i}, \mathbf{U}_{0:i}) = & \text{const} \\ & + \sum_i \|\mathbf{x}_i - g(\mathbf{x}_{i-1}, \mathbf{u}_i)\|^\top R_i^{-1} \|\mathbf{x}_i - g(\mathbf{x}_{i-1}, \mathbf{u}_i)\| \\ & + \sum_{i,j} \|\mathbf{m}_{i,j} - h(\mathbf{x}_i, \mathbf{l}_j)\|^\top Q_{i,j}^{-1} \|\mathbf{m}_{i,j} - h(\mathbf{x}_i, \mathbf{l}_j)\|\end{aligned} \tag{2.37}$$

Thanks to the sparse associations of the graph representing the SLAM problem, the update time of the graph and the amount of memory is linear with the number of landmarks, $O(N)$. Moreover, we can easily add or remove constraints between nodes and optimize only a subset of the graph.

Graph based methods are divided in two parts the *front-end* and the *back-end*. The front-end part converts sensor measurements into graph node and edge informations, is the part responsible of building the graph. The optimization process take place in the SLAM back-end.

Bundle Adjustment techniques, see Chapter 2.4, are an affordable method to optimize Graph-Based SLAM problems when the sensitive element is a camera.

2.5.1 Visual SLAM

The more recent monocular visual SLAM open source codes have been analysed and compared in order to retrieve the best candidate for the hopping/tumbling platform application. Visual SLAM codes can be divided into two main categories: feature based methods and direct methods. Feature-based methods are based on the minimization of the re-projection errors. The estimation of the camera motion is performed by matching and tracking a set of sparse interest points. As indicated in [40], a general scheme for a feature based method could be:

- Image acquisition.
- Feature detection with an interest point operator.
- Feature matching or tracking.
- Motion estimation (3D-to-3D, 3D-to-2D, 2D-to-2D).
- Mapping.
- Optimization.

On the other hand, direct methods are based on the minimization of the photometric error. The camera motion is retrieved directly from the intensities of the pixels of the images, all the pixels information are used to retrieve the camera pose. The assumption is the so-called photo-consistency constraint, the intensities of the projections of a world point $P_W = (X, Y, Z)$ on two frames are the same. Due to the great amount of data these methods are generally slower and work best for small motions and sufficient image overlap. A comparison between the two methods are outlined in Table 2.2.

Table 2.2 Feature-based visual SLAM and direct SLAM methods characteristics.

Feature based	**Direct methods**
• Can only use and reconstruct feature points	• Can use and reconstruct the whole image
• Faster	• Slower (but good for parallelism)
• Flexible: outliers can be removed retroactively	• Inflexible: difficult to remove outliers retroactively
• Robust to inconsistencies in the model/system	• Not robust to inconsistencies in the model/system
• Good initialization not needed	• Needs good initialization

Table 2.6 shows the comparison between the analyzed monocular visual SLAM codes. PTAM (Parallel Tracking and Mapping) [79] authors introduce for the first time the idea of splitting tracking and mapping into two separate tasks, processed in parallel threads. Indeed, the optimization of a big landmarks map is not possible in real-time, but the tracking of the map is kept at the frame rate. This software was specifically designed to track a hand-held camera in a small workspace. C2TAM [80] is a collaborative SLAM software, based on PTAM, where robots build sub maps and transfer the local information to a central station that performs the map optimization. REMODE [81] and DTAM [82] rely on a dense method. They can outperform feature-based methods in scenes with poor texture, defocus, and motion blur, but they are very computationally expensive, thus, they are prohibitive for this application. LSD-SLAM [83] is a semi-dense method, it overcomes the high computation requirement of dense methods by exploiting only pixels with strong gradients. Moreover, this algorithm is able to manage scale-drifting. DPPTAM [84] is a direct method, in this algorithm the authors made the assumption that homogeneous color regions belong to approximately planar area. That assumption is a limit for our operational environment constraints. SVO [85] is a semi-direct method: it uses both sparse features (such as corners or edges) and direct methods. The absence of loop closure mechanism is a limit for our application.

Indeed, re-localization is one of the major characteristics that we are looking for in a visual SLAM code for the tumbling/hopping navigation. The possibility to relocalize the hybrid rover is based on the place recognition module of the code. For this reason, we decided to discard all the visual SLAM codes that do not present this characteristic. After this pre-selection, a choice has to be done between ORB-SLAM (or the more recent ORB-SLAM2), a feature based SLAM code and LSD-SLAM. [54] shows that their code is more accurate in an indoor environment. For that reason and considering that direct SLAM methods need a god initialization, we decided to use ORB-SLAM2 code to realize the visual SLAM code for the tumbling/hopping platform.

	Monocular vision	Computationally and memory efficient	Loop Closure and Place Recognition	Large Topographic changes	Limited Inertial Navigation	Different Camera Parametrization	Invariance to sunlight direction	Open Source Code	Map Reuse Capabilities	Automatic Map Initialization	
Feature Based											
PTAM (2007)	x	x			x			x			Works on a dominant plane
C2TAM (2014)	x	x	x		x	x		x	x	x	Utilization of the depth sensor for visualization purpose
ORB-SLAM (2015)	x	x	x	x	x			x		x	Feature detector strictly embedded in the code
ORB-SLAM2 (2016)	x	x	x	x	x	x		x	x	x	Ful bundle adjustment after loop closure
Direct											
DTAM (2011)	x				x			x		x	
REMODE (2014)	x				x			x		x	Dense map reconstruction
LSD-SLAM (2014)	x	x	x	x	x			x		x	ORB_SLAM2 shows better performances
DPPTAM (2015)	x			x	x			x		x	Dense map reconstruction
Semi-Direct											
SVO (2014)	x	x		x	x			x		x	Highly sensitive to lighting situations

Fig. 2.6 Monocular visual SLAM open source algorithms comparison.

2.5.2 ORB-SLAM

ORB-SLAM is based on PTAM algorithm, it has many improvements in comparison, like a loop closing mechanism, an adequate handling of occlusions and a low invariance to the viewpoint of the re-localization. Moreover, there is no need of human intervention for map initialization. This algorithm is a graph-based SLAM method, where the graph (camera poses and map points) is built with a keyframe-based strategy (see Figure 2.7). After map initialization, new frames are tracked using a 3D-to-2D approach, as soon as possible

new frames are promoted to keyframes, which provide new map points by triangulation of salient features. Frames are tracked with reference to keyframes. The insertion of too many keyframes grows the computational complexity of the optimization phase, for that reason ORB-SLAM2 provided an efficient way to select keyframes, to triangulate map points and to match them. Table 2.3 summarize the *mappoint* and *keyframe* classes.

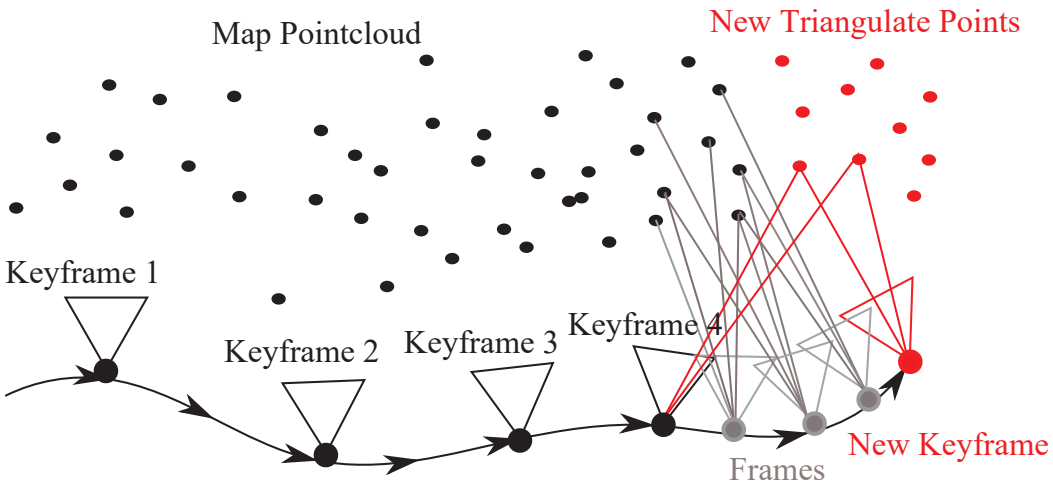


Fig. 2.7 Keyframe-based monocular visual SLAM concept.

Table 2.3 shows the parallels threads that track and optimize the frame poses and the map points position. Table 2.4 shows in which way the keyframes are connected among them.

ORB features allow real-time performance without GPUs, indeed, the use of a binary descriptor reduces the computation time during feature comparison. Moreover, ORB features provides good invariance to changes in viewpoint and illumination.

ORB-SLAM2 uses a third part library, DBoW [86], for place recognition. Thanks to DBoW2 it is possible to perform fast localization on an image sequence. This place recognition library is based on the *bag of words* model, the possibility to process a set of binary descriptors (like ORB) is available.

In the bag of words model, image features are converted into "words". We can retrieve two similar images by checking the frequency of the repeated words. Using a bag of word model the computational cost is lighter compared to check feature by feature on the whole sequence of images. Word generation is based on a vocabulary that has been trained previously with a dataset of images.

Vocabulary has to be trained in order to achieve the image descriptors clustering. The vocabulary words are the leaf nodes of the tree. The *inverse index* stores the weight of the words in the images in which they appear. The *direct index* stores the features of the images and their associated nodes at a certain level of the vocabulary tree. *Inverse index* is used

Table 2.3 Mappoint and Keyframe classes

Mappoint $\mathbf{p}_i$	**Keyframe $\mathbf{K}_i$**
• 3D position in the world coordinates system $\mathbf{X}_{(w,i)}$	• The camera pose $\mathbf{T}_{iw}$, rigid transformation from the world to the camera coordinate system
• The viewing direction $\mathbf{n}_i$, which is the means of the unit vector between the point and the Keyframe that see it	• Camera intrinsic parameters
• Mappoint ORB descriptor D_i, ORB descriptor that minimize the distance between the associated Keyframe's keypoint descriptor	• All the ORB features extracted in the frame, associated or not to a map point. The keypoint's coordinates are undistorted if a distortion model is provided
• The maximum d_{max} and minimum d_{min} distances at which the point can be observed. This distance is based on the associated keypoint scale	

Table 2.4 Covisibility graph and Essential graph

ORB-SLAM graphs	**Description**
Covisibility graph	An edge between two keyframes exists if they share the observation of the same map points (at least 15)
Spanning tree	It is a connected subgraph of the covisibility graph with a minimal number of edges. A link is formed between keyframes which share most points observation
Essential graph	It contains the spanning tree and the subset of edges from covisibility graph with high covisibility (100 shared points)

to retrieve similar images. *Direct index* is used to efficiently obtain point correspondences between images, during feature matching phases only features that belong to the same node are compared, reducing the computation time. A vocabulary tree is divided into branches k and levels L (see Figure 2.8).

ORB-SLAM2 default vocabulary has been generated with 10K images from a dataset containing sequences of indoor and outdoor images, it has 6 levels and 10 clusters per level. The conversion from the asteroid mock-up images features to vocabulary words does not work in a proper way, because there is not clustered descriptors that represent an asteroid feature. For these reasons, a custom vocabulary has been created. The vocabulary has been created with a series of images of comet 67P, asteroid Ceres, asteroid Itokawa, asteroid Eros,

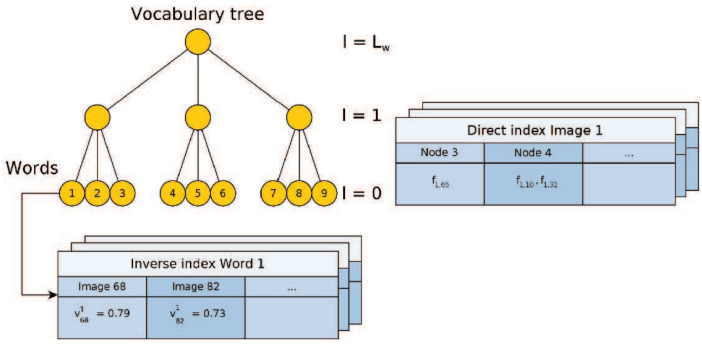
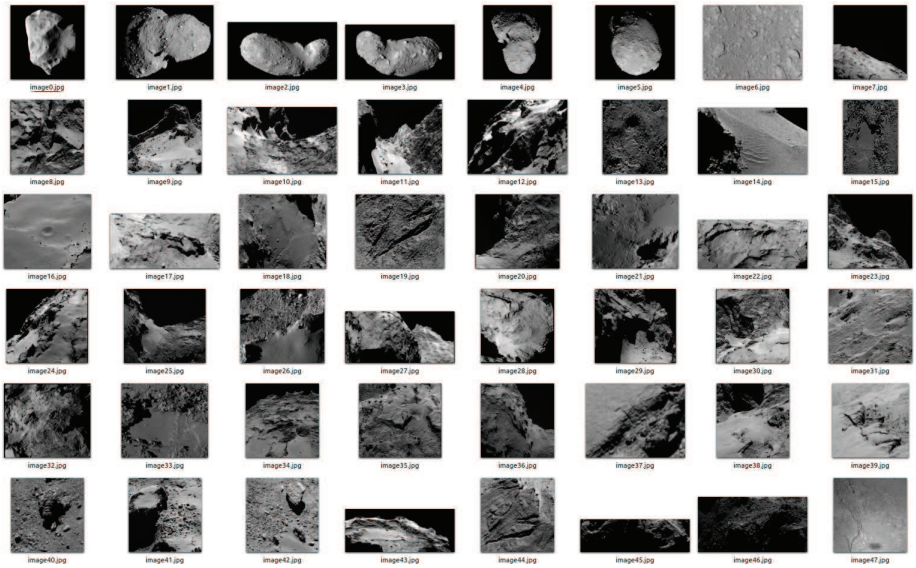
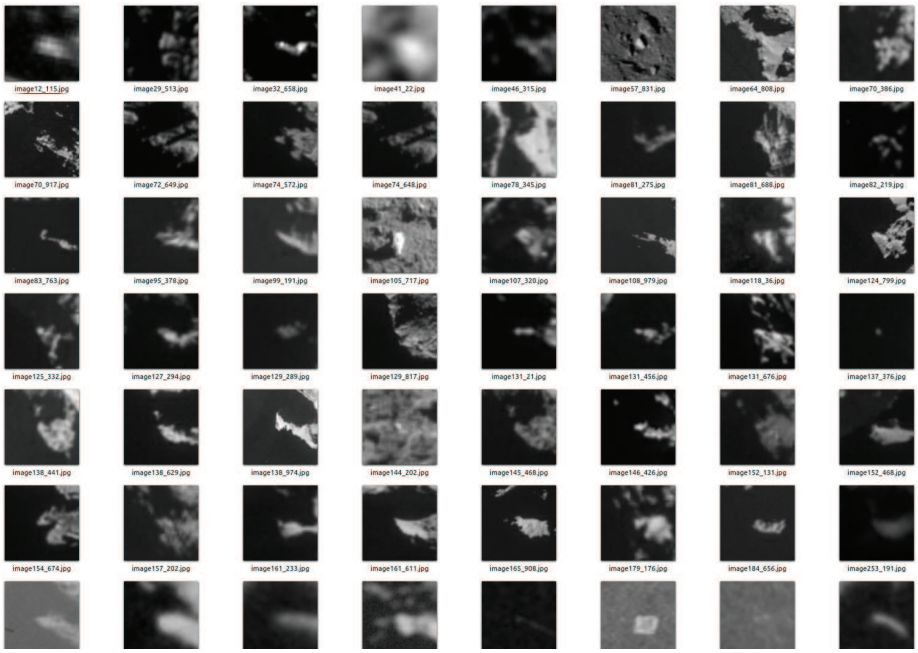


Fig. 2.8 Example of vocabulary tree and direct and inverse indexes. *Image courtesy of [86]* ©2012 IEEE

images of the Moon taken during Apollo missions, and the JPL asteroid mock-up, Figure 2.9 shows a subset of the collected images.



(a) Subset of images used for vocabulary generation.



(b) Vocabulary features that belong to the same cluster.

Fig. 2.9 Asteroid vocabulary training.

Table 2.5 ORB-SLAM threads

Tracking	**Local Mapping**	**Loop Closing**	**Global Bundle Adjustment**
• Process new frame	• Check if there are keyframes in the queue	• Check if there are keyframes in the queue	• This thread is launched from the Loop Closing thread
• Extract ORB features	• BoW conversion and insertion in Map	• Detect loop candidates and check co-visibility consistency (3D-to-3D)	• Perform the Global Bundle Adjustment, only the initial KeyFrame is fixed
• Initial pose estimation from last frame or Relocalization (2D-to-3D)	• Check recent MapPoints	• Compute similarity transformation $[sR \mid t]$	
• Track local map	• Triangulate new MapPoints	• Perform loop fusion and global pose graph optimization	
• Check if we need to insert a new keyframe (local mapping is busy?)	• Find more matches in neighbour keyframes and fuse point duplications • Local Bundle Adjustmentm • Check redundant local Keyframes	• Launch a new thread to perform Global Bundle Adjustment	
• Run continuously	• Run continuously • Stopped if a Loop Closure is detected	• Run continuously • If a Loop Closure is detected send a stop signal to Local Mapping, wait until Local Mapping has effectively stopped • If a Loop Closure is detected abort the Bundle Adjustment thread	• Lunched if a Loop Closure is detected • Lunched if a Loop Closure is detected

Chapter 3

Planetary Rover Global and Local Localization

The study of algorithms and methods for rover global localization are here presented. This algorithms have been studied and applied for the ExoMars 2020 localization at Rover Operative Control Centre in ALTEC. The adopted methodologies highlight computer vision methods. Global localization techniques uses HiRISE images and Digital Elevation Models. A method, which uses the rover panoramic horizon curve and the site Digital Elevation model, is presented. During traverse a classical stereo Visual Odometry algorithm has been studied, some tests have been performed in laboratory environment and using a Martian-like visual dataset.

3.1 Global Localization

Global positioning on extraterrestrial planets is needed to correlate rover scientific measurements with orbiters remote measurements and to validate planetary local and global models. Orbiter surface images and detailed Digital Elevation Models (DEM), produced from stereophotogrammetry or stereo photoclinometry techniques, provides significant support to the landing site selection and to plan rover operations. After rover touchdown, one of the key tasks requested to the operations center is the accurate measurement of the rover position on the inertial and fixed coordinate system, such as the J2000 frame and the Mars Body-Fixed (MBF) frame.

Based on orbital mechanics and astrodynamics, after landing, the Rover position is known with an uncertainty ranging between a hundred to a few kilometres depending on the used Entry Descent and Landing (EDL) method. Spirit landing ellipse has major and minor axes

of 78 km and 10 km [17]. The EDL architecture developed for Mars Science Laboratory [87] has reduced the landing ellipse to a 20-kilometre ellipse. For ExoMars, the landing ellipse is 104 km by 19 km [2]. Starting from a landing position evaluated with an uncertainty of some kilometres a possible sequence of further localizations could be:

1. Sextant: $\sim$1 km uncertainty. To obtain the necessary data this technique could take up to a sol, it could be performed just after rover egress phase. The position is retrieved in the MBF frame.

2. Doppler: $\sim$250 m uncertainty. The Rover does not move during some satellite passages, this method could take few sol, and could be performed during rover egress phase or just after. Thanks to the UHF Two-Way Doppler Tracking technique the rover could be localized up to an accuracy of 10 m. However, the conversion from the inertial frame to the MBF frame led to an uncertainty in the position of 250 m, and all topographic maps are expressed in the MBF frame.

3. Triangulation: $\sim$100 m uncertainty. This method consists on the cartographic triangulation of craters and hills and retrieves the rover position relative to an orbiter image of the landing site. It can be performed as soon as the rover panoramic and navigation cameras start to be fully operative.

4. Orbital imaging: $\sim$3.5 m uncertainty (depending on orbiter telescope resolution and calibration). As shown by [88], using HiRISE camera images is possible to localize the rover directly on the surface images. This is possible if the orbiter camera is available, during satellite allowable passages and when localization is already performed within $\sim$1km.

As the orbiter image used for localization is georeferenced we can considered that the rover is globally localized.

3.1.1 Map-based localization using rover surface panorama and orbiter images

This localization method is a refinement of rover position estimation performed by a human operator [20]. The location is calculated by correlation of hills and craters between a NAVCAM or PANCAM panorama and a Mars referenced image (like HiRISE image). We have applied this method for Spirit lander localization. We assume that the HiRISE image has been Mars referenced on Mars Orbiter Laser Altimeter (MOLA) topographic model previously [89]. Figure 3.1 shows the flowchart to generate HiRISE georeferenced images.

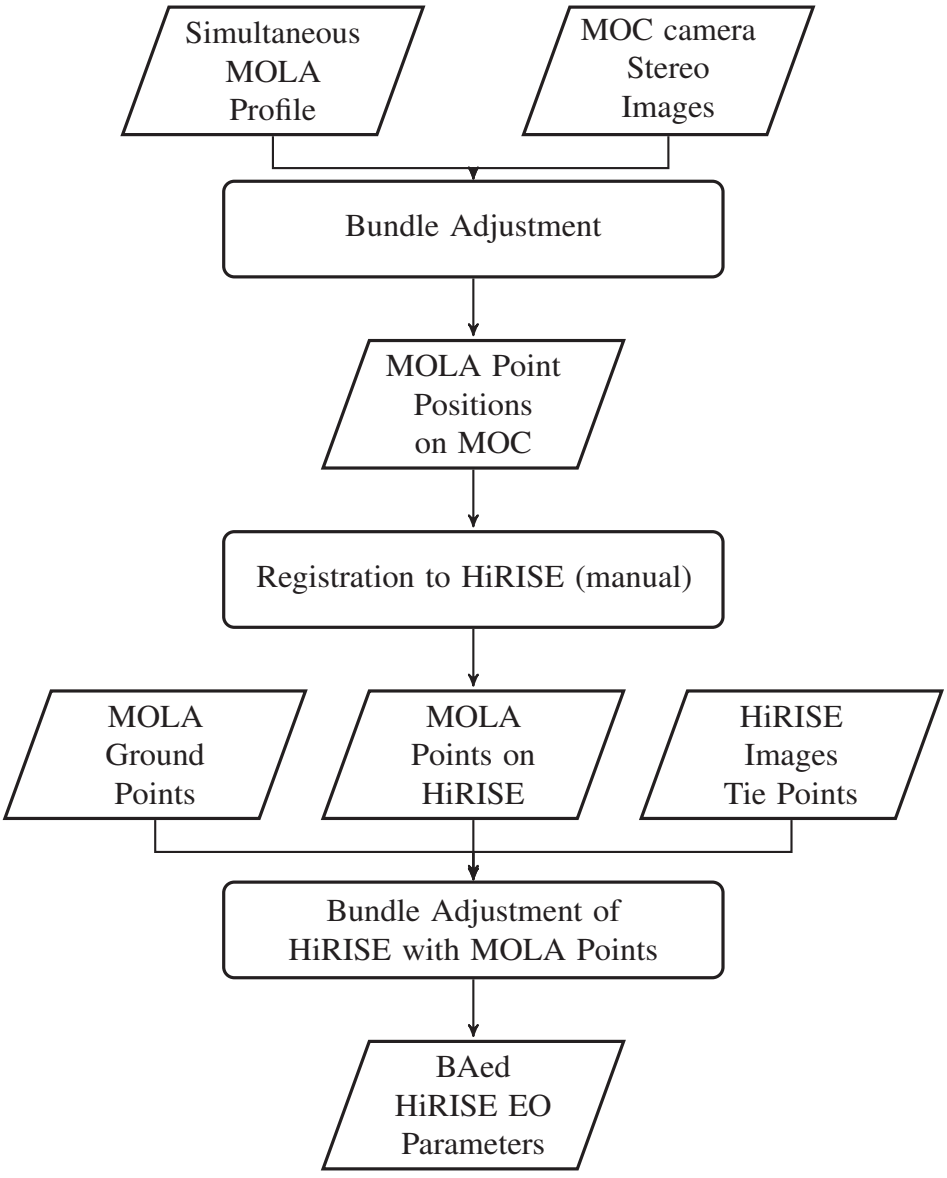


Fig. 3.1 Flowchart of HiRISE registration on MOLA points. *Image courtesy of [88]*

Li et al. [88] shows the step by step procedure used to georeferencing the HiRISE orbiter images:

1. Photogrammetric processing of HiRISE stereo-images, DEM and orthoimage generation. The generated 3-D model is based on orbital spacecraft orientation and orbital position data. The orthoimage may need to be georeferenced to Mars global topographic model. MOLA is considered the most accurate data source for Mars global topographic mapping.

2. Registration of MOLA points onto HiRISE stereo images. Direct integration of HiRISE to MOLA is difficult, so firstly MOLA and MOC data are integrated, than each MOLA point are manually transferred onto HiRISE stereo images by comparing image features of MOC and HiRISE. MOC (Mars Orbiter Camera) and MOLA are mounted together on Mars Global Surveyor satellite, so integration of MOLA with MOC is less difficult.

3. Based on the adjusted HiRISE EO parameters an orthoimage (0.25 m resolution) is generated.

In this example we will use NASA/JPL/University of Arizona processed HiRISE images, which are widely used in Mars geography to localize the rover position, using as input a surface panoramic image from the rover and the knowledge of the azimuth of the platform. Table 3.1 shows the proposed global localization method inputs and outputs. By visual inspection of a panorama it is possible to identify hills and craters in rover horizon. A NAVCAM panorama of landing site, processed by MIPL (Multimission Image Processing Laboraory) team [90], has been used for the test.

Table 3.1 Correlation method inputs and output

Input
NAVCAM or PANCAM panorama
Rover azimuth α
Previous estimated position (λ_0, ϕ_0)
Orbiter image (Mars referenced)
Landing site DTM
Output
New rover position $(\lambda_{new}, \phi_{new})$

In order to test this method we have used the images at the landing site collected by the NASA MER rover Spirit. The rover camera images have been retrieved from the from the

MER Analyst's Notebook website http://an.rsl.wustl.edu// [91]. The orbiter images used for localization are the orthorectified HiRISE NASA MarsReconnaissance Orbiter (MRO) mission camera HiRISE [92], available online from https://hirise.lpl.arizona.edu/. Thanks to direct observation of the landing site by HiRISE camera orbiter was possible to estimate the error of our method.

Spirit Localization

Figure 3.2 shows the NAVCAM cylindrical projection mosaic taken at Spirit landing site (2nn001edn00cyl00p1501l000m2).

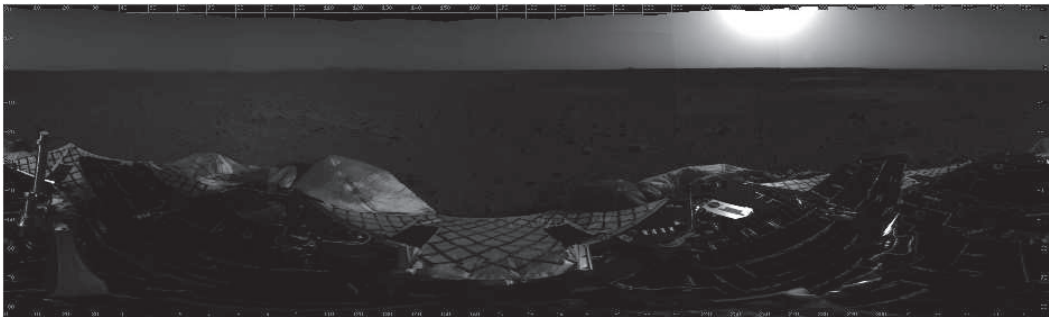


Fig. 3.2 NAVCAM Cylindrical projection mosaic of Site 0 Position 0, from Spirit on Sol1. Image Credit: NASA/JPL/Cornell.

Fig. 3.3 Mountain features in Site 0 Position 0 mosaic.

The following equations correlate the azimuth α and elevation ϑ of the observed objects with the panorama image:

$$\alpha = \frac{i}{\text{IMAGE\_RESOLUTION}} + \alpha_0 \tag{3.1}$$

$$\vartheta = \frac{\text{IMAGE\_ELEVATION\_LINE} - j}{\text{MAP\_RESOLUTION}} \tag{3.2}$$

where (i, j) are the pixel coordinates and α_0 is the starting azimuth. In the image label we can find informations about the IMAGE_RESOLUTION, the START_AZIMUTH and the PROJECTION_ORIGIN_VECTOR (expressed with reference to the SITE_FRAME) [90]. The image size is 557×1866 pixels, the pixel resolution is 5.18515 pixel/deg, the starting azimuth is 0 deg, and $(-0.556355, 0.113974, -0.896827)$ m is the projection origin vector.

It is possible to know α_0 by knowing sun position relative to the rover and the current time [15].

By starting to observe the panorama it is possible to identify some landmarks, that it will possible to observe also in the orbiter image, like hills and craters. By knowing their pixel positions, thanks to Equations 3.1, we retrieve their azimuth.

In the Fig.3.3, we have identified five features (coloured lines), that belong to surrounding hills and mountains. The following azimuths are identified: 65.96°, 101.25°, 114.75°, 124.59°, 216°.

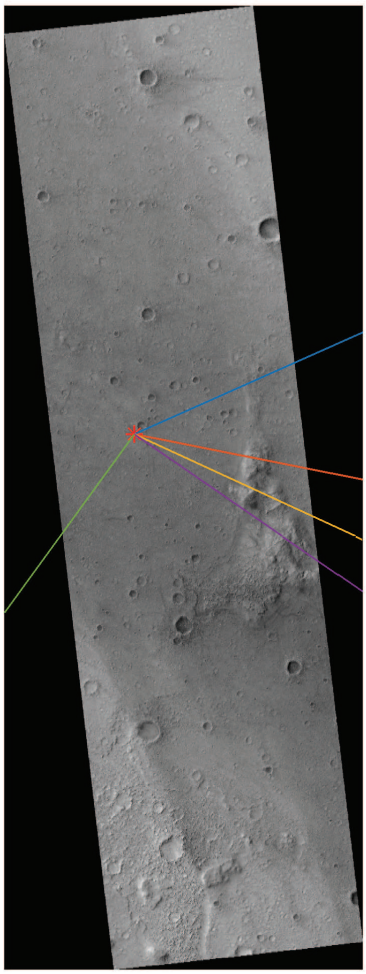


Fig. 3.4 Mars Exploration Rover landing site at Gusev Crater (PSP_001513_1655_RED). Red star shows Spirit lander position as estimated from two-way Doppler techniques. Coloured lines are the direction of the mountainous landmarks identified in the landing site panorama.

The location of Spirit landing site, estimated by fitting direct-to earth (DTE) two-way X-band Doppler and two passes of UHF two-way Doppler between MER-A and Mars Odyssey, is 14.571892°S 175.47848°E [20]. This position is used to identify the corresponding mountainous features in the orbiter image, as shown in Fig.3.4. The figure shows lander two-way Doppler position estimation, and hills view direction, superimposed to the equirectangular projection of HiRISE image.

The following equations shows the correlation between a pixel of the image (u, v) and the corresponding latitude and longitude (λ, φ):

$$\begin{aligned} u &= r(\lambda - \lambda_W)) \\ v &= r(\varphi_{max} - \varphi)) \end{aligned} \tag{3.3}$$

where r is the map resolution. In the HiRISE image of Mars Exploration Rover landing site at Gusev crater (PSP_001513_1655_RED.JP2) the latitude goes from -14.779191204696° to -14.405986429528°, and the longitude from 175.42815231742° to 175.56889973542°, the image size is 88471×32859 pixels. Thus, the image resolution is 237056 pixel/deg.

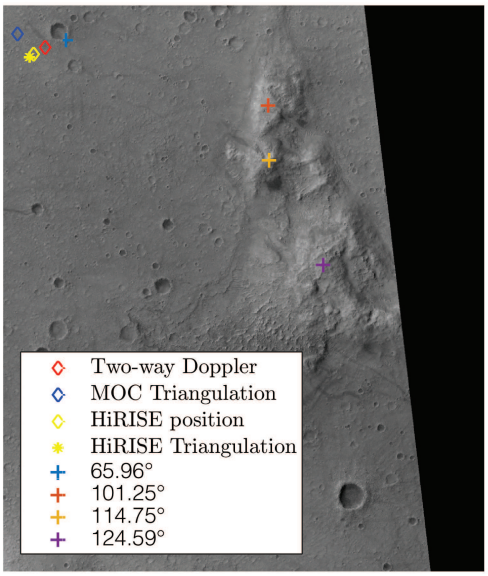


Fig. 3.5 Hills and craters used for cartographic triangulation of Spirit landing site (HiRISE PSP_001513_1655_RED image), the yellow star is the estimated position, lander's position estimated with two-way Doppler method is highlighted by a red square, the blue square is the estimation made by [20] with MOC images. The yellow square is the landing position captured by HiRISE, and used as ground truth.

Table 3.2 shows the peaks location on the map. The peak corresponding to 216° azimuth direction is outside PSP_001513_RED image. By triangulating the located points we can estimate the new rover position $P(X,Y)$ [93]:

$$\tan\alpha_i = -\frac{X_i - X}{Y_i - Y} \tag{3.4}$$

$$\begin{pmatrix} 1 & \tan\alpha_i \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} = X_i + \tan\alpha_i Y_i \tag{3.5}$$

where α_i is the azimuth location of the landmark and (X_i, Y_i) is the landmark location on the map (in pixel). By using the following notation: $A_i = \begin{pmatrix} 1 & \tan\alpha_i \end{pmatrix}$, $B_i = X_i + \tan\alpha_i Y_i$, $A = [A_1, \ldots, A_n]^\top$ and $B = [B_1, \ldots, B_n]^\top$ Eq. 3.5 becomes:

$$AP = B \tag{3.6}$$

a direct solution can be obtained thanks to a least square minimization:

$$P = (A^\top A)^{-1} A^\top B \tag{3.7}$$

Table 3.2 Location of azimuth landmarks in PSP_001513_1655_RED HiRISE image.

Azimuth	**Orbiter Image Location**
65.96°	(12907, 38982) pixel
101.25°	(22552, 42100) pixel
114.75°	(22605, 44706) pixel
124.59°	(25173, 49693) pixel

Using Table 3.2 data, we find the new rover position estimation, which is $P = (11198,\ 39790)$ pixels, using the inverse of Equation 3.3 we find that $P = (14.5738387°S,\ 175.4753884°E)$. Considering that the HiRISE images resolution is 0.25 m/pixel, the distance between the estimated positions and the HiRISE detected location are: 161.33 m by using the Two-Way Doppler technique, 307.55 m by using the cartography method with MOC images and 55.55 m by using the cartographic method with HiRISE images.

3.1.2 Map-based localization using the Panoramic Horizon and orbiter images Digital Elevation Models

This algorithm, through an exhaustive search, attempts to estimate rover position with a high resolution Digital Elevation Model (DEM) and panoramic images. We consider a vector of N template positions and orientations $\mathbf{X} = \{X_1, \ldots, X_N\}$, the state of the rover is defined by $X_i = (\mathbf{p}_i, \alpha_i)$, where $\mathbf{p}_i = [x_i, y_i]$ defines the position, and α_i the heading relative to the map. The rover will be localized in one of the template positions. A vector of skylines is rendered from the DEM for each template position, each rendered skylines is compared to the measured skyline, which is retrieved from a panoramic image of the surface. The algorithm is capable to retrieve rover's heading (α_i), platform roll and pitch angles are supposed to be known before capturing panoramic images.

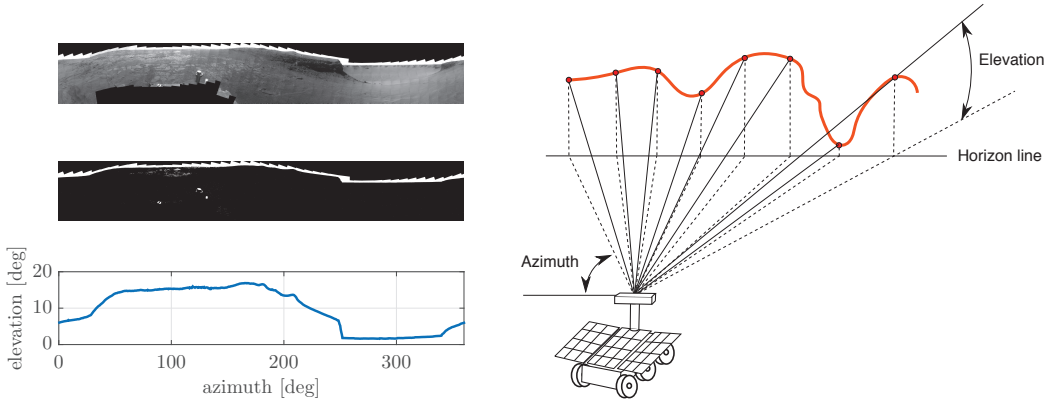
Algorithm pipeline

The used algorithm is summarized in the following steps (see also [94, 95]):

1. Skyline measurement. A segmented panorama of the landing site, or of the current rover location, is generated by combining images with overlapping field of view. The skyline is defined by a vector of elevation samples m_ϕ measured for each azimuth ϕ. In order to compute the skyline vector, the grey-scale panorama has been converted to a black and white binary image by applying a luminance threshold. The pixels where is located the limit between the two regions is the local skyline. Fig.3.6a shows the skyline automatic extraction passages. The conversion between image coordinates, expressed in pixel and the azimuth/elevation coordinates is performed by knowing the camera intrinsic parameters. By sampling the skyline every 1 deg, $\phi = [0, \ldots, 359]$ deg, the skyline vector $\mathbf{m} = [m_0, \ldots, m_{359}]$ is then obtained.

2. Skyline rendering from DEM. A Digital Elevation Model (DEM) is a digital representation of terrain surface, for each position $\mathbf{p} = [x; y]$ it is associated an altitude $z(\mathbf{p})$. The HiRISE DEM resolution is $\sim$1 m, and as an example, the expected vertical precision EP of the Victoria crater DEM (used below for Opportunity localization), assuming 0.2-pixel matching error, is 0.22 m [96]. Given an azimuth angle ϕ_j the interpolation line between the viewing direction and the DEM is:

$$a_{j,k} = z\left(\mathbf{p} + k\begin{bmatrix}\cos(\phi_j + \alpha)\\ \sin(\phi_j + \alpha)\end{bmatrix}\delta p\right) - d(k\delta p) \tag{3.8}$$



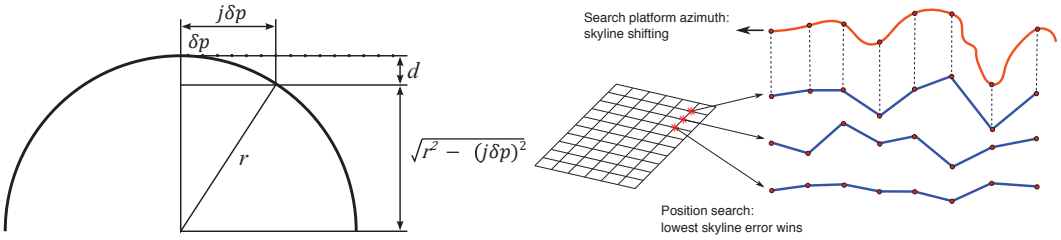
(a) (b)

Fig. 3.6 (a-Top) Panoramic image taken by opportunity from sol 1530 to sol 1545 on the slopes of Victoria crater (credit NASA/JPL/Cornell). (a-Middle) Conversion of panorama to a binary image for automatic skyline extraction. (a-Bottom) Measured skyline converted form pixel to azimuth/elevation coordinates. (b) Skyline segmentation.

where α is the rover heading, $k = 0, \ldots, M$ are the number of step where the DEM is interpolated along the viewing direction and $d(k\delta p) = r - \sqrt{r^2 - (k\delta p)^2}$ is the planet curvature correction. r is the planet radii and $(k\delta p)$ is the interpolation step. Knowing the height of the camera above the ground h, the elevation angle $e_{j,k}$ is given by:

$$e_{j,k} = \arctan\left(\frac{a_{j,k} - (a_{j,0} + h)}{k\delta p}\right) \tag{3.9}$$

The elevation at the horizon s_j is the maximum elevation $\max(e_{j,k})$. The vector $\mathbf{s}(\mathbf{p}, \alpha) = [s_0, \ldots, s_{359}]$ is then obtained. Fig.3.8 shows the horizon on the DEM used for Spirit localization at the landing site and Fig.3.9 the corresponding skyline.



(a) (b)

Fig. 3.7 (a) Altitude correction based on the curvature of the planet. (b) Best candidate location searching.

3. An exhaustive search is performed in order to find the best candidate location and the platform azimuth. The best candidate position is evaluated on a grid of template

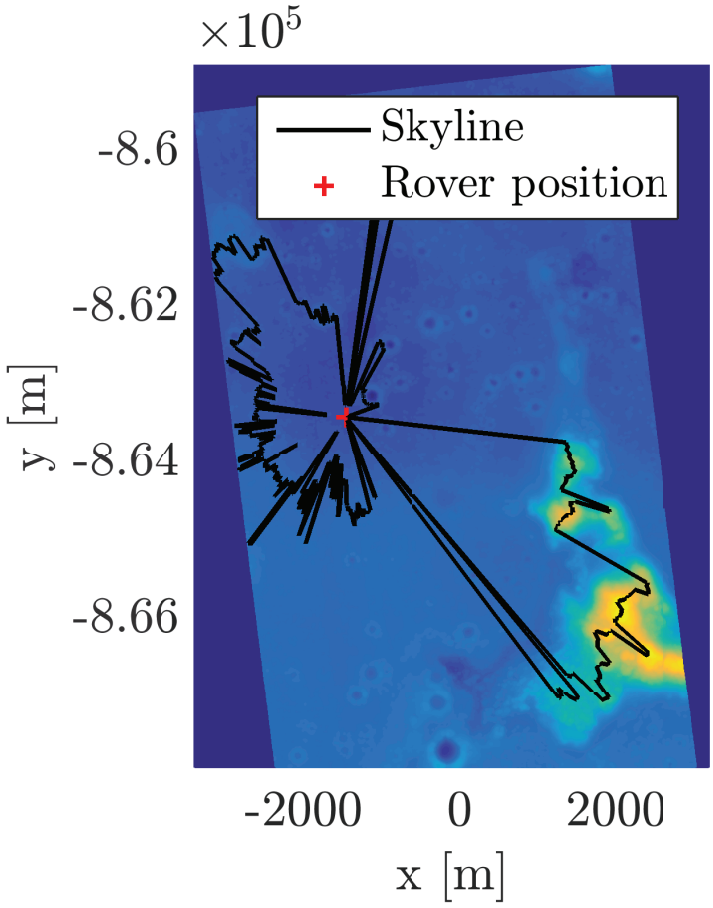


Fig. 3.8 Horizon viewed from Spirit landing site on the HiRISE DEM used for the skyline rendering. The rendered skyline is limited by the DEM surface coverage.

positions, a database of template skylines is build for each point of the grid. The squared error between the measured skyline and the rendered skyline is evaluated at each position of the grid and for each sampled azimuth.

$$\varepsilon^2(\mathbf{p},\alpha) = \sum_j \left(m_j - s_j(\mathbf{p},\alpha)\right)^\top \left(m_j - s_j(\mathbf{p},\alpha)\right) \tag{3.10}$$

Fig.3.7b shows the error computation for each template position of the grid.

4. The best candidate location $(\mathbf{p}^*, \alpha^*)$ is the one which has the lowest error:

$$(\mathbf{p}^*, \alpha^*) = \underset{(\mathbf{p},\alpha)}{\operatorname{argmin}}\, e^2(\mathbf{p},\alpha) \tag{3.11}$$

Spirit landing site localization

In the Spirit landing site panorama it is possible to see the Columbia Hills Complex, see Fig.3.9. This dataset allows us to evaluate the algorithm performance when it is possible to see a series of mountain landmarks in the distance. The mosaic view[1] available on the MER Analyst's Notebook website has been used as panorama for the skyline extraction.

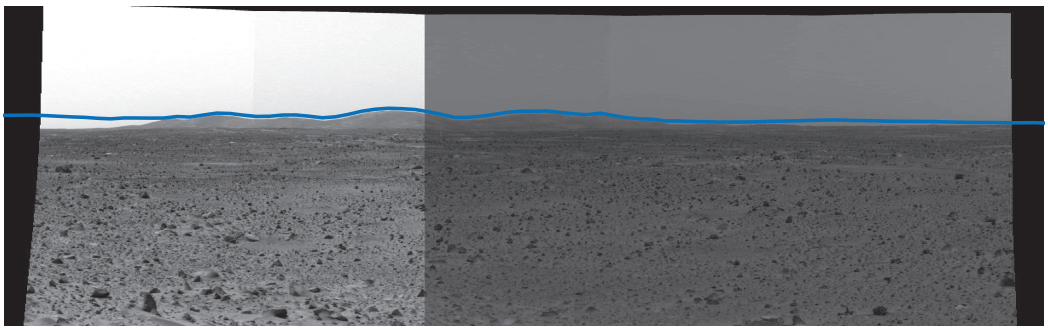


Fig. 3.9 Skyline rendered at the landing site position (blue line)

Equation 3.1 and equation 3.2 correlate the azimuth and elevation of the observed objects with the panorama mosaic pixels.

Equation 3.12 gives the relation between latitude/longitude and DEM coordinates in meters. The coordinates of Spirit landing site HiRISE DEM[2] are expressed with reference to the $central\_meridian = 175.5$ deg and $standard\_parallel = -10$ deg. The DEM, which cover an area of 6.6×11 km has a size of 6524×10847 pixel and a resolution of $\mu = 1.01$ m/pixel.

$$\begin{aligned} x &= x_0 + u\mu \\ y &= y_0 - v\mu \end{aligned} \tag{3.12}$$

(x_0, y_0) are the coordinates of the upper-left corner of the DEM expressed in meters and (x, y) are the coordinates in meters corresponding to the image point (u, v).

Template positions are selected on a regular spaced grid disposed inside a square area of 1000×1000 m centred on an initial position, see Fig.3.11. The initial position is the location of Spirit landing site estimated by fitting direct-to earth (DTE) two-way X-band Doppler and two passes of UHF two-way Doppler between MER-A and Mars Odyssey, which is 14.571892°S 175.47848°E [20].

As we can see from Fig.3.10 only a subset of mountainous features is contained in the HiRISE DEM, only the Columbia hills complex is covered by the DEM model. In this case,

[1] The used mosaic view is 2pp003iff02cyl00p2211l222m4

[2] The used DEM is DTEEC_001513_1655_001777_1650_U0

using the whole skyline could lead to a wrong result. In order to overcome the limited DEM size related issues, we have used only the skyline within a range.

In order to identify the best skyline range for localization purposes, we rendered N skylines corresponding to N different positions using the pipeline detailed in Section 3.1.2. As first guess we use $N = 50$ random locations. Fig.3.10 shows the skylines rendered at 50 random positions around the estimated landing site location. We can see that rendered skylines approach the measured skyline around the Columbia Hill complex ($\alpha = 95.2°/129.7°$), instead rendered skylines does not match the measured skyline around Grissom Hill and Apollo 1 Hills. Skyline range between 91° and 145° has been considered in this example for rover localization.

The algorithm has been tested for different grid, DEM resolution and view line step (δp), with an angular resolution of 0.5 [deg]. Table 3.3 shows the results. The best candidate location is the one which has the lowest root mean square error between the measure skyline and the rendered skylines.

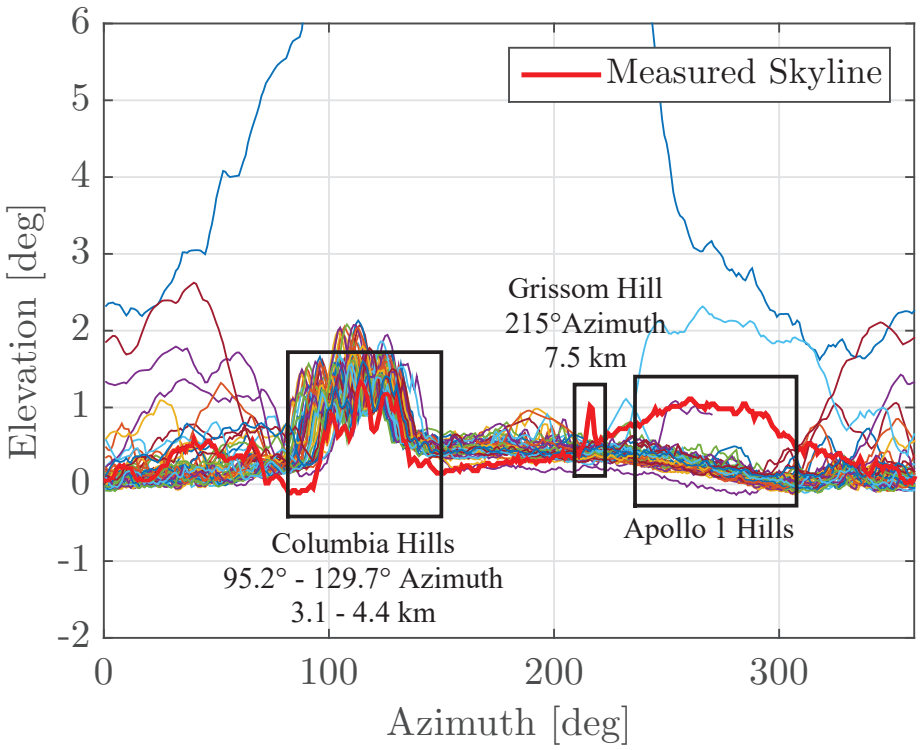


Fig. 3.10 Rendered skylines around the estimated landing site, red line shows the measured skyline.

As explained in [97], some artefacts are present in USGS HiRISE DEMs, as the long line that we can see in Fig.3.12. The seams are caused by the characteristic of HiRISE images,

Table 3.3 Localization error and measured azimuth, for different grid, DEM resolutions and view line step.

Grid [m]	DEM [m]	δp [m]	Error [m]	$\alpha_{\mathbf{mes}}$ [deg]
4	4	10	55	90
8	4	10	51	90
8	4	20	51	90
8	4	30	51	90
8	4	100	51	90
8	8	10	51	90
8	8	20	51	90
8	8	30	51	90
8	8	100	51	90
8	16	10	72	89.5

which are made up to 10 individual images (HiRISE focal plane is composed by 10 different CCD). These lines may lead to an error in the DEM reconstruction up to 1-2 meters and, consequently, to a bias in elevation angle estimation, (as we can see in Figure 3.13).

Fig.3.11 shows the five best candidate location for the grid 8 [m], DEM 8 [m], $\delta p = 10$ [m] and $\delta\alpha = 0.5$ [deg]. Fig.3.12 shows the five best location on the DEM.

We consider as candidate location the position that shows the lowest difference between the measured and the rendered skyline. Considering Table 3.3 results for a DEM resolution of 8 [m] and a searching grid of 8 [m], the best candidate position is $P = (-13883,\ -863399)$ [m] (DEM reference frame). By using the inverse of Equation (3.12) we find that $P = (14.5680859°S \quad 175.4762191°E)$. The distance between the estimated landing positions and the HiRISE detected landing location on PSP_0011513_1655_RED_A_01_ORTHO are 289 [m] with the Two-way Doppler method and 51 [m] with the method described hereby. The panorama starting azimuth is 0 deg, the azimuth measured thanks to the skyline matching techniques is -1 deg. Figure 3.14 shows Spirit lander location on HiRISE map-projected image, its coordinates has been used to evaluate the distances of estimated positions of the table above.

Localization error seams to not decrease with the reduction of the view line step. This may be caused by some artefacts present on the DEM. As we can see in Fig.3.15, some skylines points are rendered in correspondence of HiRISE DEM artefacts (see reference [97]). These points will not be present in the measured skyline and will increase the difference with the rendered skyline. Fig.3.12 shows that spirit landing site is closer to this line. Most probably if we increase the view line step the first interpolation point on the DEM is beyond the DEM artefact.

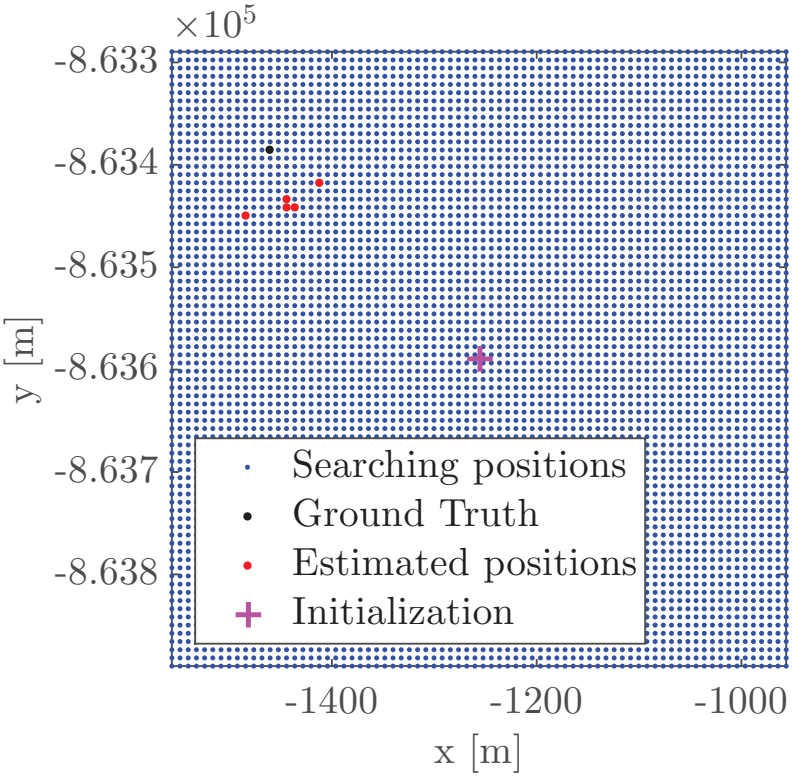


Fig. 3.11 Localization algorithm results using $[m_{91},\dots,m_{145}]$. (Blue dot) Location of the rendered skylines, prior positions of the map. (Magenta cross) Estimated location with two way Doppler. (Black dot) Spirit landing site on HiRISE image *PSP*_0011513_1655_*RED*_*A*_01_*ORTHO.IMG*, reference position. (Red dot) Best five position estimates.

Localization error seams to not decrease with the reduction of the view line step. This may be caused by some artefacts present on the DEM. As we can see in Figure 3.15, some skylines points are rendered in correspondence of HiRISE DEM artefacts (see reference [97]). These points will not be present in the measured skyline and will increase the difference with the rendered skyline. Figure 3.12 shows that spirit landing site is closer to this line. Most probably if we increase the view line step the first interpolation point on the DEM is beyond the DEM artefact.

Opportunity localization

Two different operative scenarios have been investigated for the Opportunity dataset. The DEM of Victoria crater produces using HiRISE images has been used. Two panorama has been analysed: one taken from outside the crater and the other taken from inside the crater.

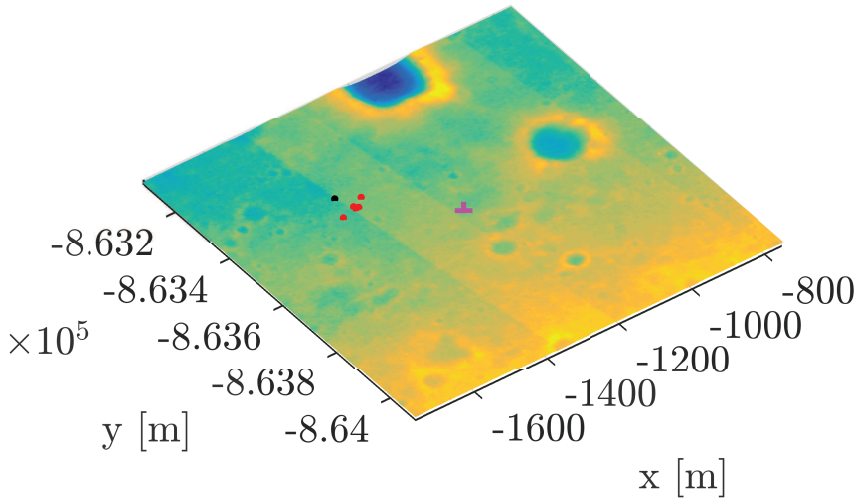


Fig. 3.12 HiRISE DEM of Spirit Landing site with estimated landing positions. (Magenta cross) Two-way X-band Doppler estimation of landing position. (Black) Spirit landing site on HiRISE image *PSP*_0011513_1655_*RED*_*A*_01_*ORTHO.IMG*, reference position. (Red dot) Best five guessed positions.

During sol 952 Opportunity taken a panorama from outside Victoria crater. The algorithm has been run with a skyline obtained obtained form Figure 3.16, as we can see the scene is almost plat. With these conditions has not been possible to localize the rover.

From sol 1530 to 1545 Opportunity rover was located in the crater slopes, Figure 3.17 shows the position of the rover inside the Victoria crater and prior positions of the searching grid. Figure 3.16 shows the panoramic image used for rover localization and Figure 3.6a the image converted in a binary image based on greyscale threshold.

Figure 3.17 shows the reference position and the algorithm estimated position on Victoria crater DEM. Table 3.4 reports localisation errors and azimuth measurements. In order to estimate the accuracy, rover position at sol 1530-1545 on traverse map (Figure 3.19) has been taken as reference.

The global localization algorithm based on the skyline matching, and presented in this chapter, it is part of the framework under development at ALTEC S.p.A. for the Mars global localization of ExoMars 2020. The contest of utilization is highlighted in Figure 3.20.

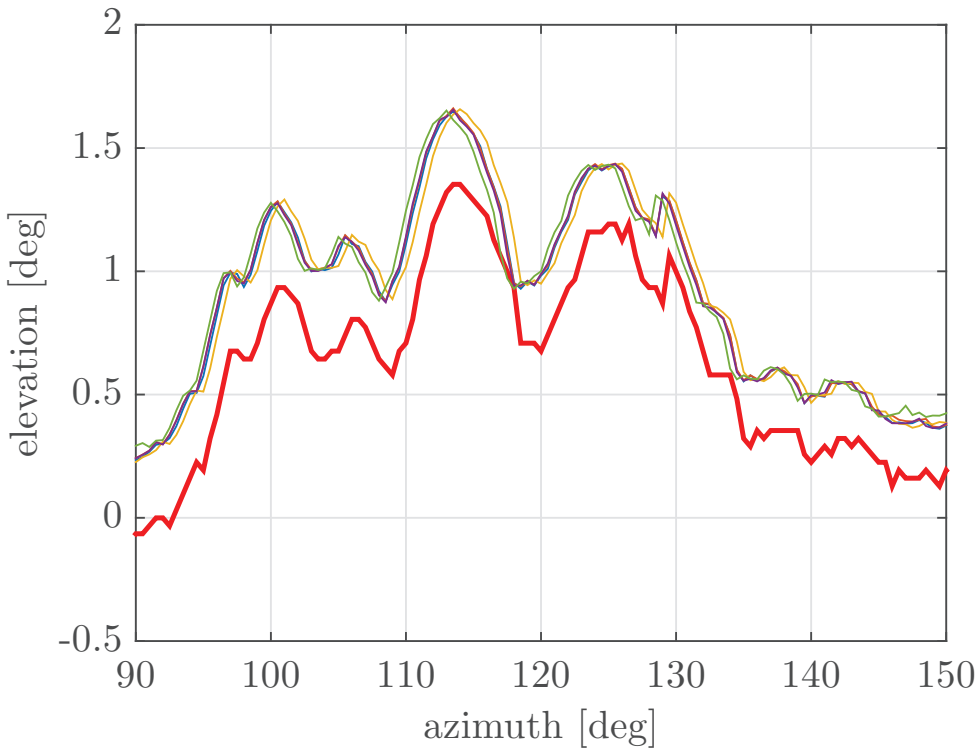


Fig. 3.13 Comparison between detected skyline (red line) and the five best matched positions skylines.

Fig. 3.14 Spirit lander location on HiRISE image.

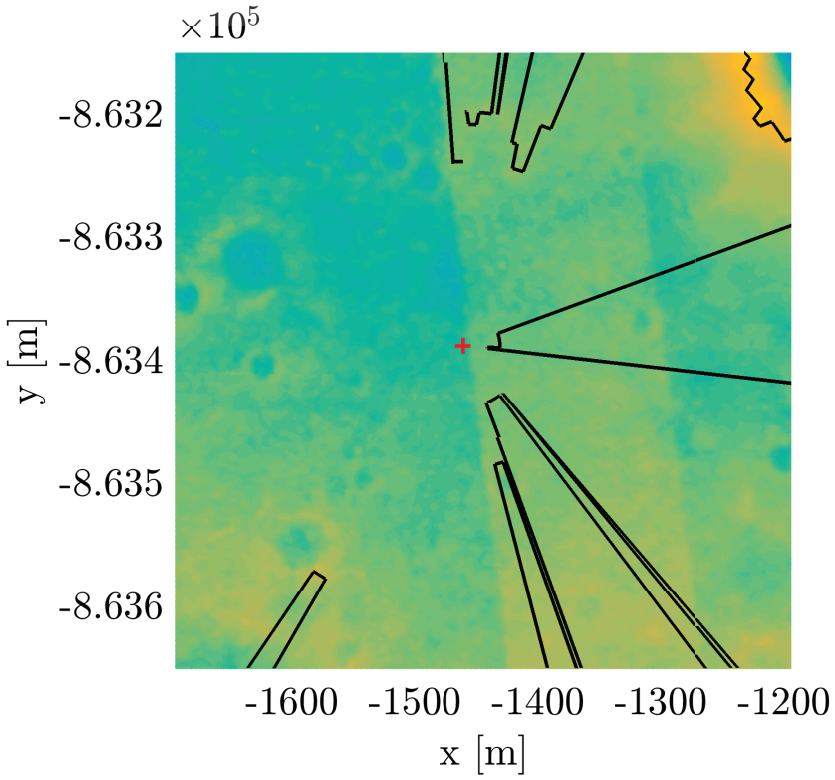


Fig. 3.15 Horizon and DEM around the landing position. Some horizon points correspond to a DEM artefact, like a CCD seams, which limits the possibility to reduce the localization error.

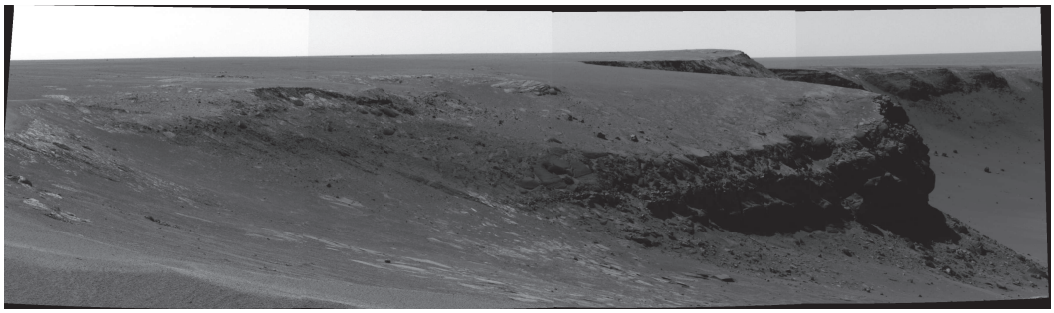


Fig. 3.16 Panoramic image taken by Opportunity during sol 952 of Victoria Crater.

Table 3.4 Four best position estimation for Victoria crater localization scenario, DEM resolution 1 [m], prior grid spacing 4 [m] and skyline resolution 0.5 [deg].

$e^2(\mathbf{p},\alpha)$ **[rad]**	**Error [m]**	$\alpha_{\mathbf{mes}}$ **[deg]**
0.2782	28	184
0.2865	22	181
0.3533	24	179
0.4254	25	185.5

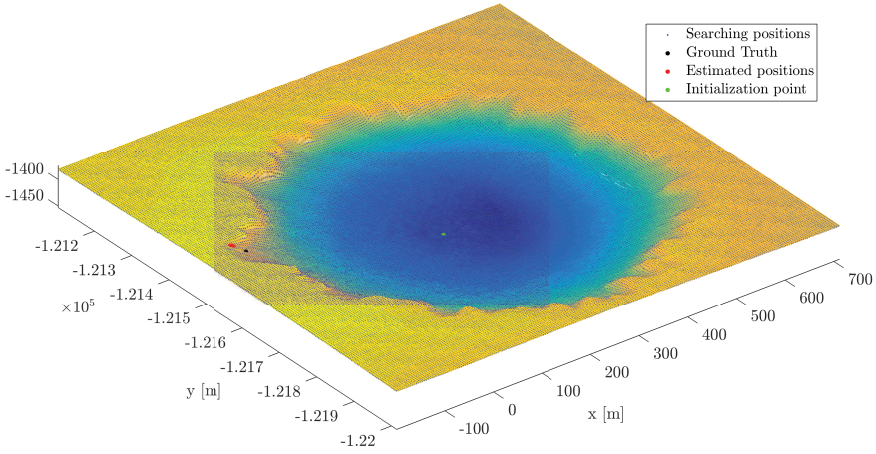


Fig. 3.17 HiRISE DEM of Victoria crater and Opportunity rover estimated position.

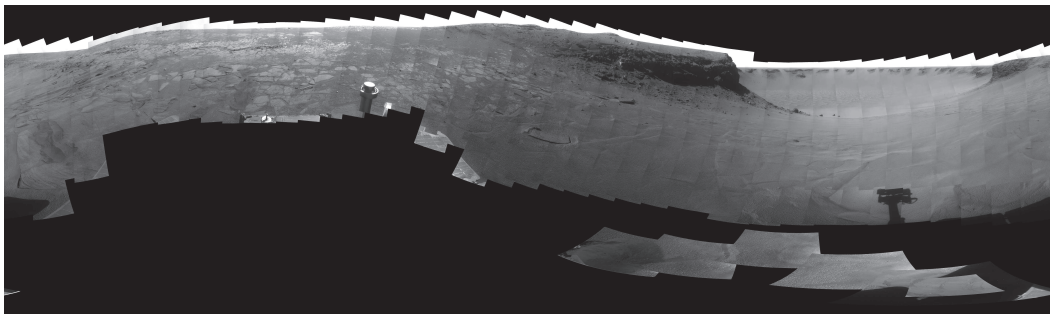


Fig. 3.18 Panoramic image taken by opportunity from sol 1530 to sol 1545.

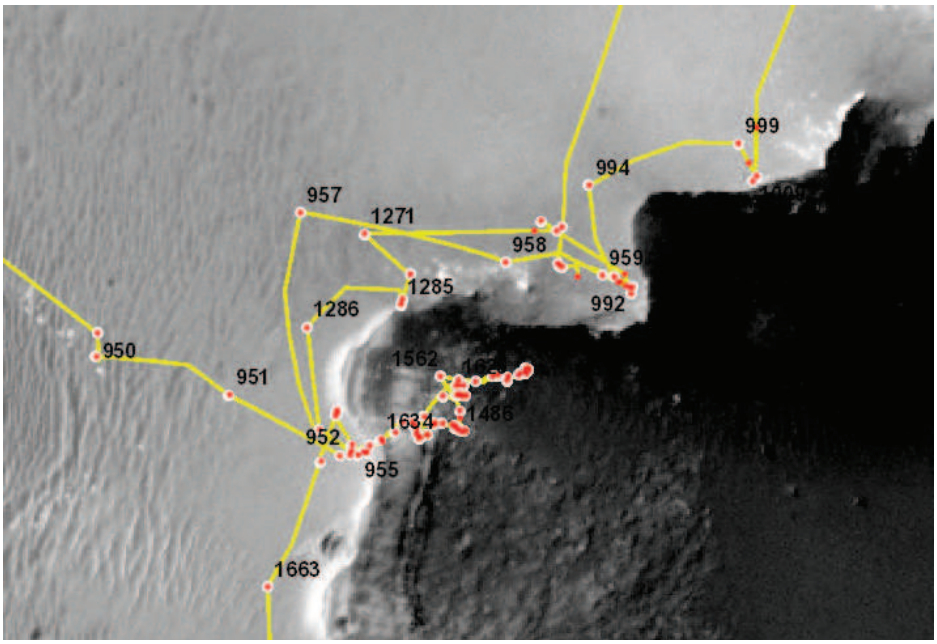


Fig. 3.19 Opportunity rover location on traverse map (on Analyst's Notebook web site).

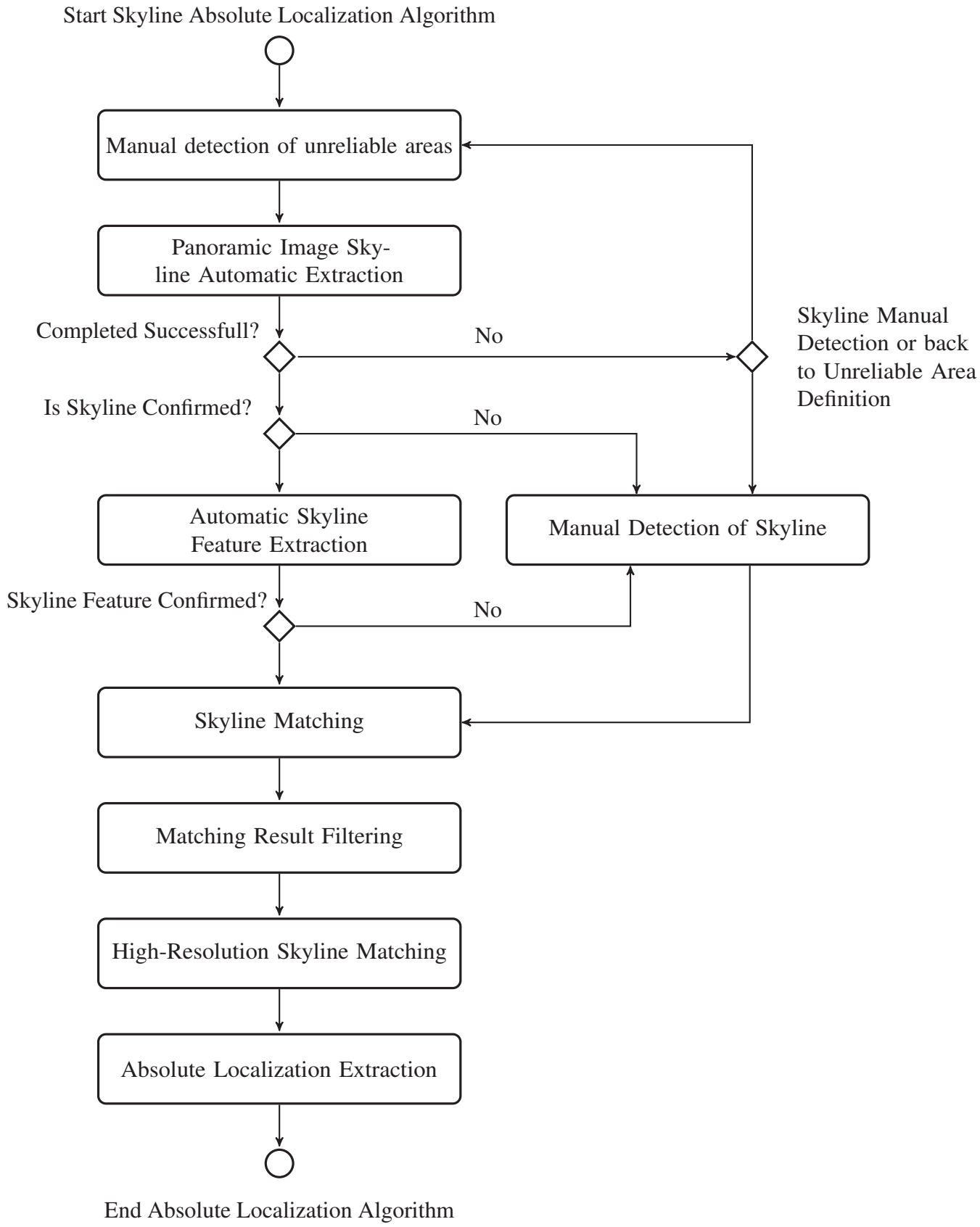


Fig. 3.20 Block diagram of the global localization framework under development at ALTEC S.p.A. *Courtesy of ALTEC S.p.A.*

3.2 Localization During Traverse

During rover traverse ROCC should be provided with Visual Odometry algorithms that refine the on-board estimated path with stereo images processed off-line.

We have investigated a stereo Visual Odometry method based on a 3D-to-3D method. The pseudo code of the algorithm is highlighted in Algorithm 1. It has been realized considering the theoretical aspects described in Chapter 2.3. First of all the intrinsic and extrinsic camera parameters are calculated using Zhang calibration method [45]. At each motion step, the left and right images are captured by the stereo pair. Images are then processed in order to find the keypoints, which are the projection of the 3D physical landmarks in the camera's image plane. Furthermore, a Harris detector [46] and the Scale Invariant Feature Transform (SIFT) detector [49] are used in order to exploit both corner and blob features of the scene, and so to have a high number of 3D points. The matching between stereo pairs and subsequent images is performed with a SIFT descriptor. This is the major difference between the employed algorithm and that of [98].

After 2D feature detection and matching, the 3D coordinates of the landmark are calculated via the middle point algorithm [59]. The detected 2D features are then filtered: the features that are less than 1 pixel apart and the features which have a re-projection error ($\mathbf{err_{i-j}}$) $\sqrt{\mathbf{err_{1-2}^2} + \mathbf{err_{3-4}^2}} > 4$ are removed. Since each motion step is relatively small, a single physical landmark should have its image projections relatively close to one another, thus, corresponding features that have a distance larger than a given threshold value of 500 pixels are discarded. Then, the maximum number of features used for computing the displacement and rotation is limited to the 1000 points, with the lowest descriptor distance.

The system pose is calculated in two steps, as in [59]. First, the motion is calculated by a least square estimation coupled with a RANdom SAmple Consensus (RANSAC) process to remove outliers. Then, a nonlinear minimization procedure is performed in order to take into account the uncertainty of landmarks. Only 3D points that are not removed during the RANSAC procedure are passed to the non-linear phase.

Levemberg-Marquardt algorithm is used to refine the stereo camera pose (rotation matrix and translation), the L_2 distance between the triangulated 3D points is minimized. Equations 2.9, 2.10 and 2.11 show the non-linear cost function E_{nl}. We can see that each component of the error vector $\mathbf{e_i}$ of feature i is weighted taking into account landmark uncertainty. The uncertainties of the 3D points are represented by 3×3 covariance matrices as calculated by the Kline-McClintock formula, see GUM [70].

The stereo VO algorithm has been tested on Mars/Moon analogue site image dataset, called The Devon Island Rover Navigation Dataset [99]. The dataset provides ground truth

Algorithm 1: Pseudo code of the tested stereo Visual Odometry algorithm for rover relative localization during traverse.

Input : $I^l_{0:n} = \{I^l_0, \ldots, I^l_n\}$ and $I^r_{0:n} = \{I^r_0, \ldots, I^r_n\}$
Output : $T_{1:n} = \{T_{1,0}, \ldots, T_{n,n-1}\}$
Parameters : N_{RANSAC}: number of RANSAC iterations, τ_{RANSAC}: threshold, K_l, K_r: Camera calibration parameters

`// Frame 0 Initialization`
[$\mathbf{u}^l_0$]=DETECT_HARRIS&SIFT_FEATURES(I^l_0);
[$\mathbf{u}^r_0$]=DETECT_HARRIS&SIFT_FEATURES(I^r_0);
`// Frame k>0 iterate`
for *(*$k = 1$,k++*)* **do**
  [$\mathbf{u}^l_k$]=DETECT_HARRIS&SIFT_FEATURES(I^l_k);
  [$\mathbf{u}^r_k$]=DETECT_HARRIS&SIFT_FEATURES(I^r_k);
  [ind^r_k, ind^l_k, ind^r_{k-1}, ind^l_{k-1}, N]=INTER&INTRA_FEAUTRE_MATCHING($\mathbf{u}^r_k$, $\mathbf{u}^l_k$,$\mathbf{u}^r_{k-1}$,$\mathbf{u}^l_{k-1}$);
  [$\mathbf{p}_{k-1}$]=STEREO_TRIANGULATE($\mathbf{u}^l_{k-1}(\text{ind}^l_{k-1})$, $\mathbf{u}^r_{k-1}(\text{ind}^r_{k-1})$, K_l, K_r);
  [$\mathbf{p}_k$]=STEREO_TRIANGULATE($\mathbf{u}^l_k(\text{ind}^l_k)$, $\mathbf{u}^r_k(\text{ind}^r_k)$, K_l, K_r);
  `// start RANSAC`
  $n_{c,max} \leftarrow 0$;
  for *(*$i = 1$, $i \leq N_{RANSAC}$, i++*)* **do**
    choose random keypoint indices [ind_1, ind_2, ind_3];
    $\widetilde{\mathbf{p}}_{k-1} \leftarrow \left[\mathbf{p}^{\text{ind}_1}_{k-1}, \mathbf{p}^{\text{ind}_2}_{k-1}, \mathbf{p}^{\text{ind}_3}_{k-1}\right]$;
    $\widetilde{\mathbf{p}}_k \leftarrow \left[\mathbf{p}^{\text{ind}_1}_{k}, \mathbf{p}^{\text{ind}_2}_{k}, \mathbf{p}^{\text{ind}_3}_{k}\right]$;
    `// find transformation with random samples`
    [$\widetilde{R}_{k,k-1}, \widetilde{\mathbf{t}}_{k,k-1}$]=POINT_CLOUDS_ALIGNMENT($\widetilde{\mathbf{p}}_k, \widetilde{\mathbf{p}}_{k-1}$);
    foreach $p^{ind}_k \in \mathbf{p}_k$ *and* $p^{ind}_{k-1} \in \mathbf{p}_{k-1}$ **do**
      if $(p^{ind}_k - R_{k,k-1}p^{ind}_{k-1} - \mathbf{t}_{k,k-1}) < \tau_{RANSAC}$ **then**
        $n_c \leftarrow n_c + 1$;
        store index *ind*;
      end
      if $n_c > n_{c,max}$ **then**
        $n_{c,max} \leftarrow n_c$;
        $R_{RANSAC} \leftarrow \widetilde{R}_{k,k-1}$;
        $\mathbf{t}_{RANSAC} \leftarrow \widetilde{\mathbf{t}}_{k,k-1}$;
      end
    end
  end
  `// non linear optimization, RANSAC mition estimation is used as initial condition for the LM algorithm`
  [$R_{k,k-1}, \mathbf{t}_{k,k-1}$]=LEVENBERG_MARQUART($\mathbf{p}^{ind_{RANSAC}}_k$, $\mathbf{p}^{ind_{RANSAC}}_{k-1}$, R_{RANSAC}, $\mathbf{t}_{RANSAC}$);
end

information s on the captured stereo couples by means of a differential GPS, an inclinometer and a sun sensor Figure 3.21 shows a couple of images of the dataset.

Figure 3.22 shows a comparison between the ground truth and the trajectory reconstructed by means of the VO software. The error of this stereo Visual Odometry method is 2.04%.

Fig. 3.21 Stereo Images from The Devon Island Rover Navigation Dataset [99].

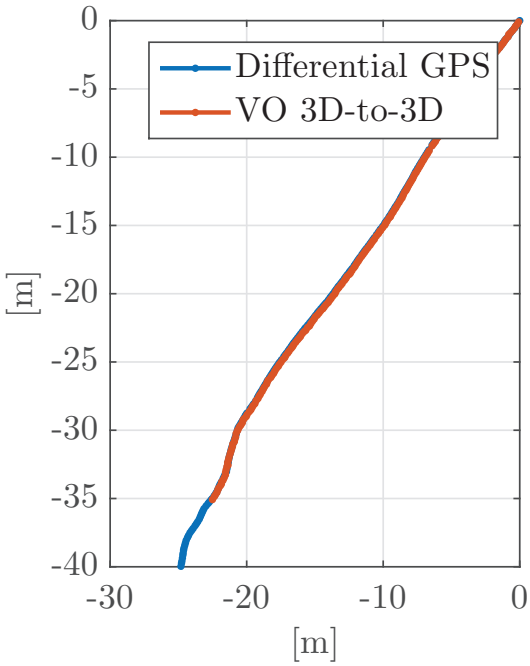


Fig. 3.22 Estimated trajectories using images from The Devon Island Rover Navigation Dataset. Comparison between the differential GPS signal and a 3D-to-3D VO method.

3.3 Visual Odometry System Performance for Different Landmark Average Distances

As we have done in [100], hereby, we have performed an analysis of how the landmark distance affects the VO motion estimation. This is important in VO studies in order to set up possible solutions to reduce Visual Odometry drift.

Efforts are being made to reduce error in motion measurements operated by VO systems. Uncertainty modeling for 3D captured landmarks, as shown by [57], need to be anisotropic. It is well known that the position uncertainty of the triangulated points, along the viewing ray, increases as the two camera rays become more parallel [101, 65]. As a result, a stereo camera whose baseline is small, the triangulated point uncertainty is greater along the viewing ray compared to the other directions. Moreover, in a stereo camera system, the 3D points are less precisely localized as the landmark moves away from the stereo camera. For this reason, as discussed in [102, 103], the cost function to be minimized during the pose optimization takes into account Mahalanobis distances. Each 3D error is weighted by its inverse covariance matrix, this generally improves the accuracy obtained by a 3D-to-3D approach.

Reference [104], considers the anisotropic uncertainty of the acquired landmarks, showing that is possible to reduce the error of the position estimates recovered from stereo VO by changing the perspective of the stereo camera in relation to the moving platform. In [103], we applied the same VO algorithm as described here to the measurement of a vehicle trajectory in order to investigate how camera's FOV affects the whole camera trajectory error and uncertainty. [105] points out that stereo VO has two sources of bias: the distribution of landmarks and incorrect modeling of their associated uncertainty.

Hereby, we focus on comparing the measurement uncertainties obtained by a visual odometry system when the average features distance of the scene changes. In order to change the average feature distance, the VO system has been tested in seven different positions progressing further into the laboratory environment. In order to give a direct experimental comparison, only the system position and pose were changed, while all other influencing parameters were kept constant (for example same cameras, same FOV, same relative positions of cameras, same elevation angle of cameras, same imposed rotary and linear motions) while a rigorous uncertainty analysis, according to [70, 106], has been performed. Although the cost function for the pose optimization takes into account Mahalanobis distances, the translational pose uncertainty still grows with landmarks average distance.

The VO method used [98] is described in Algorithm 1, it is a 3D-to-3D method based upon NASA's Mars rovers approach [59] and [58]. Compared to [98] we used a different feature detector, moreover in [98] the analysis is limited to how the motion amplitude affects

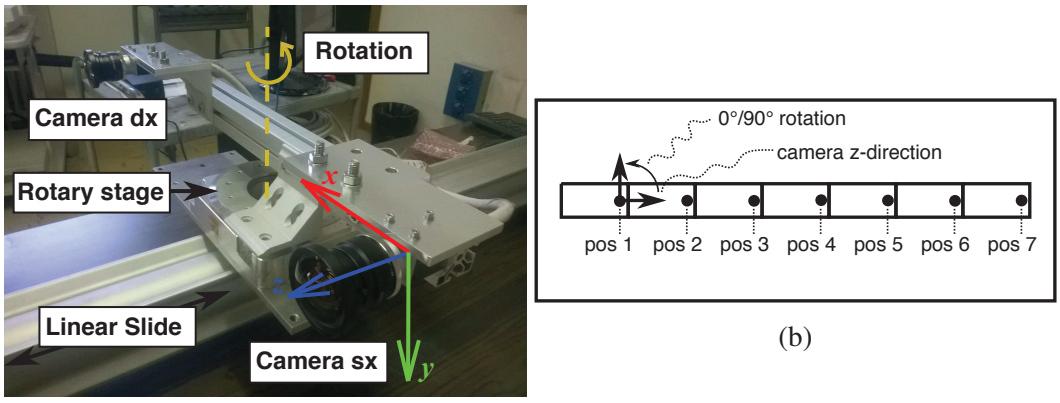
Fig. 3.23 Features extracted by the Harris and the SIFT detector. Two detectors have been used in order to have an high number of 3D points.

a single step error and uncertainty. In the following sections the experimental set-up and the obtained results are discussed.

3.3.1 Experimental set-up and results

Figure 3.23 shows the scene with the observed features, by utilizing two different descriptor algorithm the features have a more dispersed distribution and increases the overall number of features.

Experimental set-up consists of a stereo camera mounted on a high precision motor-driven rotary stage, which was, atop a linear slide; thus translations and rotations could be independently achieved. A stepper motor drove the rotary stage, which resolution is equal to 0.0003°, its position repeatability is 0.02°. The linear slide had a resolution equal to 1 mm; reading to the nearest division gave an error of no more than ± 0.5 mm. We can take these to be a uniformly distributed uncertainty. Experimental set-up and Visual Odometry 1-sigma uncertainties are highlighted in Table 3.5, while Figure 3.24a shows the employed set-up. In Table 3.5 σ_Z is expressed in a range because, as we can see in Figure 3.27, the



(a) (b)

Fig. 3.24 (a) Experimental set-up. (b) Rotary stage position in the laboratory environment.

Table 3.5 Uncertainties of the Visual Odometry system compared to the experimental set-up uncertainties.

Visual Odometry system	
σ_Z	0.5 - 3 [mm]
σ_θ	$2.5 \cdot 10^{-2}$ [deg]
Experimental set-up	
$\sigma_{linear\,slide}$	0.3 [mm]
$\sigma_{rotary\,stage}$	$5.7 \cdot 10^{-3}$ [deg]

translational step uncertainty changes by changing the experimental set-up position.Thanks to the motor-driven rotary stage used in the experimental set-up, each imposed rotation from frame to frame are known with an uncertainty of at least one order of magnitude better than that of the VO's estimate. The linear slide resolution is limited to 1 mm, so its uncertainty is not one order of magnitude better than the Visual Odometry system as requested by a reference system, but at least it is lower.

During the translation test, stereo camera images are acquired exactly every 50 mm for a total span of 1350 mm along the z-axis, as shown in Figure 3.24a; while the rotation test goes from 0° to 90° around the y-axis, in increments of 1°, as driven by the stepper motor. Rather than considering the uncertainty arising from the initial rigid transformation between the linear slide's reference frame and that of the camera, and having to propagate it through the final position uncertainty; during the tests we only measured the magnitude of the translation vector.

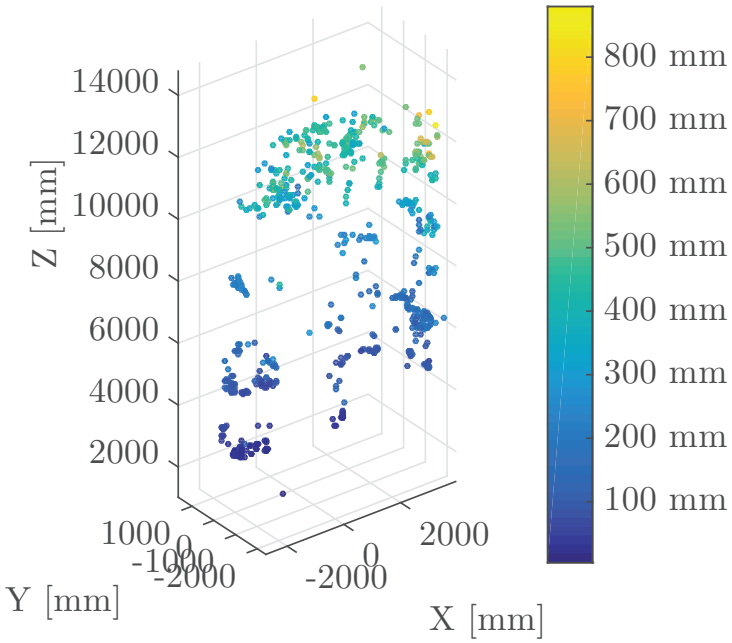


Fig. 3.25 Position 1 landmark point cloud, each point is colored with the value of the uncertainty ellipsoid major axis.

In order to see how the average feature distance affects the rotation and the translation measurements we have repeated the test in seven different positions, as depicted in Figure 3.24b. The experimental set-up is manually brought to each new position, as it approaches the wall the feature distances, with reference to the stereo-camera system, decrease. Each matched feature is triangulated, so we know the point position within the camera's reference frame, from this it is then possible to measure the average feature distance as a function of stereo-camera movement; as shown in Figure 3.26c.

Figure 3.25 shows the point cloud corresponding to the triangulated features. For each point, the correspondent uncertainty ellipsoid has been calculated. The uncertainty ellipsoids of the 3D points are represented by 3×3 covariance matrices calculated by the Kline-McClintock formula, see GUM [70]. As is visible, the 3D points ellipsoid major axis grows with the distance from the stereo camera centre Table 3.6 shows the variances of the camera properties that have been propagated to variances in the landmark 3D positions. The uncertainty of the intrinsic ($\sigma_{focal\,length}, \sigma_{optical\,center}$) and extrinsic camera parameters ($\sigma_\theta, \sigma_\alpha, \sigma_\beta, \sigma_X, \sigma_Y, \sigma_Z$) have been retrieved from the Zhang calibration procedure [45]. The half width divided by $\sqrt{3}$ of a uniform distribution has been taken as standard uncertainties for the intrinsic and extrinsic camera parameters.

Table 3.6 Camera characteristics and intrinsic and extrinsic parameters uncertainties, they have been propagated to obtain the variances in the landmark 3D positions.

Camera characteristics	
Image size	1084×2040 [pixels]
Focal length	6 [mm]
FOV	$86^\circ \times 53^\circ$
Baseline	546 [mm]
$\sigma_{focal\,length}$	1 [pixel/mm*mm]
$\sigma_{optical\,center}$	1 [pixel]
$\sigma_\theta, \sigma_\alpha, \sigma_\beta$	$5 \cdot 10^{-2}$ [deg]
σ_X	2 [mm]
σ_Y, σ_Z	1 [mm]

Translations

Linear translation test results obtained for the seven tested positions are summarized by the following figures. Figure 3.26a illustrates the measured total errors for each motion step, which are the differences between the measured and imposed total displacements. Here we have considered, as an error, the difference between the modulus of the measured distance ($\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$), and the imposed displacement. Figure 3.26b shows the standard uncertainty evaluated by the Monte Carlo approach for the total position along the z axis (the z-axis being parallel to the linear slide). Figure 3.26c depicts the number of features used during the non-linear phase of the VO flowchart to estimate the motion step. Figure 3.26d shows the features average distance as function of the imposed displacement. These uncertainties are obtained combining all the single uncertainties of each motion step from the first one to the considered position. Relative displacement error is less than 0.41% for all the tested positions. For the linear translation tests, the relative standard uncertainty is less than 1.39% along the x-axis and 0.9% in the z-direction.

The uncertainty grows faster in the first part of the sequence, when the camera moves forward the feature distances decrease, so, as is shown in Figure 3.25, also their uncertainty decreases. As result the step uncertainty is greater when the features are located far away from the camera, as shown in Figure 3.27. The uncertainty of the 3D-to-3D matched features are still affecting the motion estimates despite during motion optimization Mahalanobis distance has been taken into account.

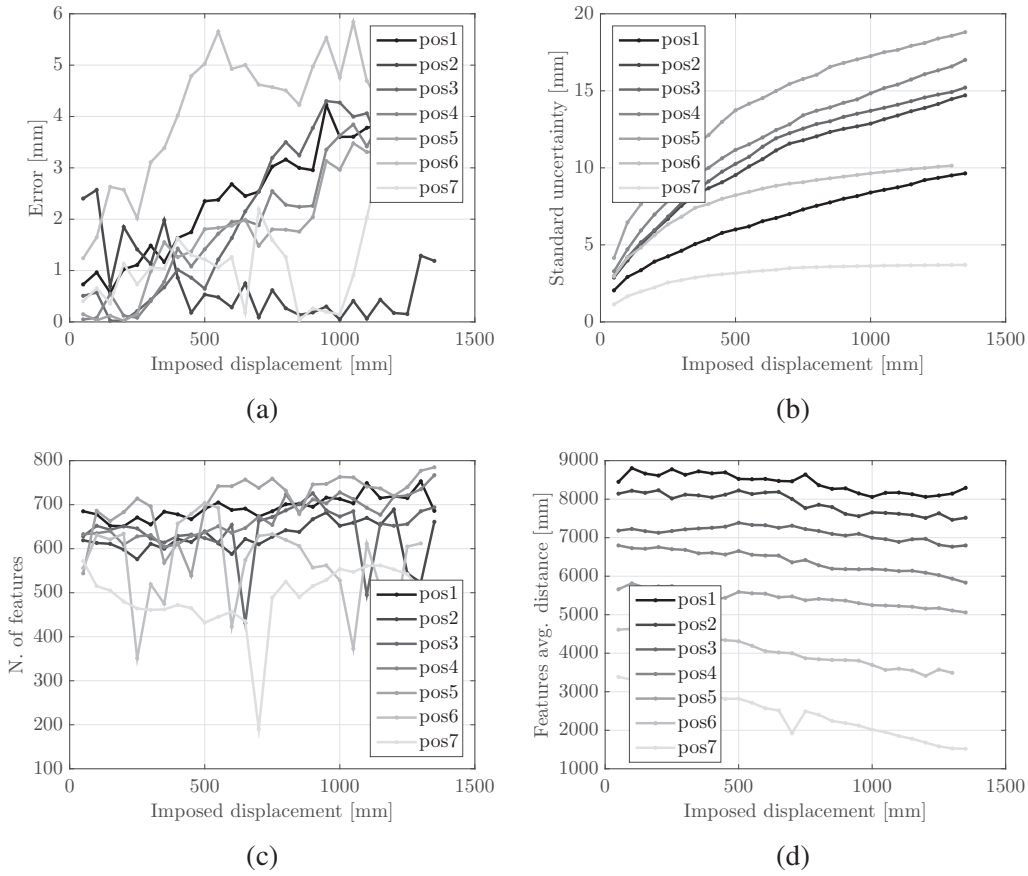


Fig. 3.26 (a) Total displacement error in the linear translation test; (b) total position standard uncertainty along z-axis in the linear translation test; (c) number of features used in the non-linear optimization, during the linear translation test; (d) features average distance in the linear translation test.

Rotations

$0°/90°$ rotation test results obtained for the seven tested positions are summarized by the following figures: Figure 3.28a illustrates the measured total errors for each motion step, which are the differences between the measured and the imposed total rotation. Figure 3.28b shows the standard uncertainties evaluated by the Monte Carlo approach for the total rotation around the y-axis (z-axis is parallel to the linear slide, while the x-axis is orthogonal, see Figure 3.24a). These uncertainties are obtained by combining all the single uncertainties of each rotation step from the initial position to that of the point being considered. Landmark distances do not seem to affect the uncertainty of the measurements. The feature average distance decreases around $\theta = 60°$, when the camera starts to face the closest wall side, parallel to z-axis. In correspondence of the same point, and for the same reason, the number

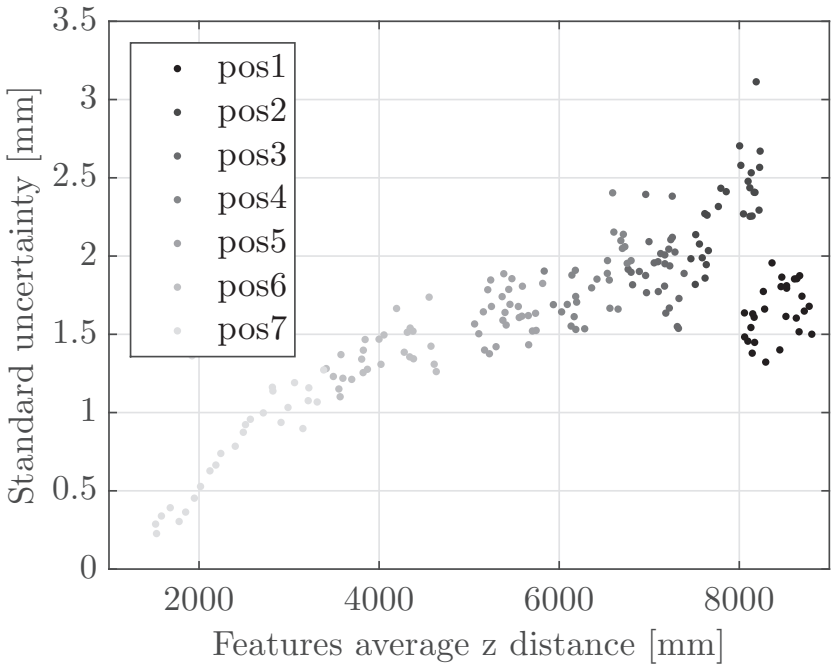


Fig. 3.27 Translation step standard uncertainty along z-axis as a function of features average distance.

of matched features decreases. Relative rotation error is less than 0.27% for all the tested positions and the relative standard uncertainty is less than 0.38%.

Despite 3D landmark uncertainty growing with distance from stereo-camera center, the rotation uncertainty does not. If landmarks are uniformly distributed within the stereo camera's FOV, rotation uncertainty does not depend on feature distance. This is because pixel position measures angles.

Figure 3.29 shows the step rotation uncertainty, evaluated using the experimental data in all the considered positions, as a function of features average distances. The mean value of rotation step standard uncertainty is 0.0249 [deg], this is the mean value for the seven 0/90 and -90/0 counter-clockwise tests.

In the case of an unbalanced features distribution rotation uncertainty grows. As we can see in Figure 3.29 when z_{ave} is between 2000 and 4000 [mm] the rotation step uncertainty grows. The distribution of features in camera FOV is unbalanced when z_{ave} is between 2000 and 4000 [mm] because the distance between the stereo camera and the laboratory wall is small, so the number of features matched correctly decreases between two consecutive images.

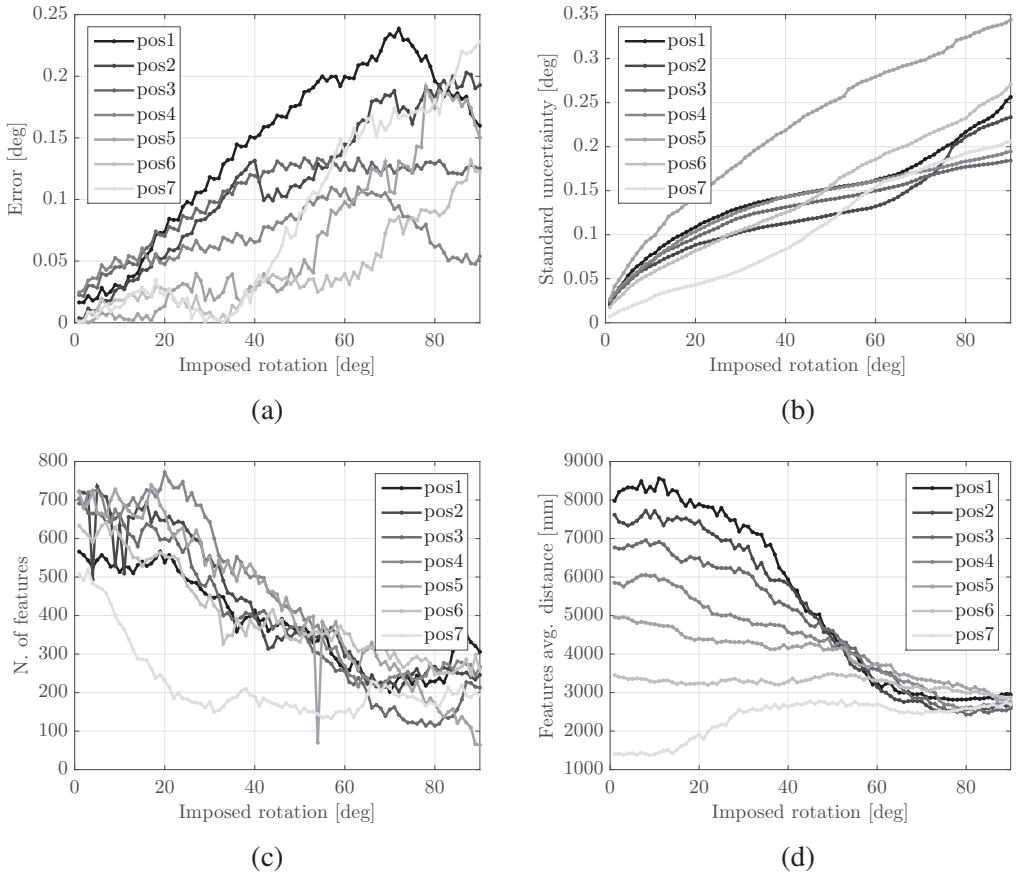


Fig. 3.28 (a) Total rotation error in the 0/90 rotation test, absolute value; (b) total rotation standard uncertainty around y-axis in the 0/90 rotation test; (c) number of features used in the non-linear optimization, during the 0/90 rotation test; (d) features average distance in the 0/90 rotation test.

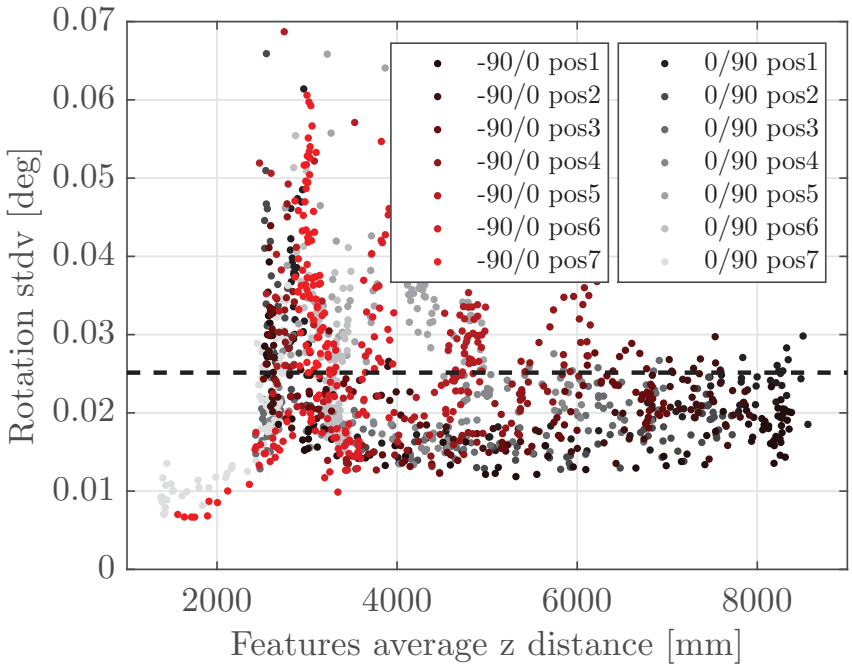


Fig. 3.29 Rotation step standard uncertainty along y-axis as a function of features average distance, for counterclockwise rotations. Dashed-line: average value.

Chapter 4

Visual SLAM techniques for small spacecraft and interplanetary exploration

Hopping/tumbling mobility enables exploration and large coverage of comets, asteroids and other small bodies. These new mobility systems require new approaches for localization. Localization problem can be split into two main area: global localization and relative localization. Global localization means that we retrieve the position of the rover in the frame of reference of the small body under exploration. Could be useful to localize globally a rover in order to enable the correlation of rover's measurements with observations from other platforms, like the mother spacecraft. Moreover, given a target location in the small body reference frame and by knowing rover's pose, global localization allows the rover to drive autonomously. Relative localization is the localization of the platform with reference to platform initial position, it is known also as tracking problem. It is helpful to keep tracking of the rover's trajectory when its global position is not known. Moreover, relative localization is useful for hazard avoidance and path re-planning.

In this chapter, we present a synergistic localization and mapping approach between the mother-craft and a deployed daughters. Before the daughter deployment into the small body surface, the mother spacecraft provides the map of the small body surface. The a priori map of the small body is built by using a narrow FOV camera on-board the mother-craft. The daughter is relocalized during long-range hops, where the platform could reach altitudes as high as of 100 m for a small-sized asteroid [33]. Figure 4.1 shows the hopping/tumbling rover concept.

The limited size and power resources available for the platform had led the choice of the navigation sensors and therefore the localization algorithm. The navigation algorithms

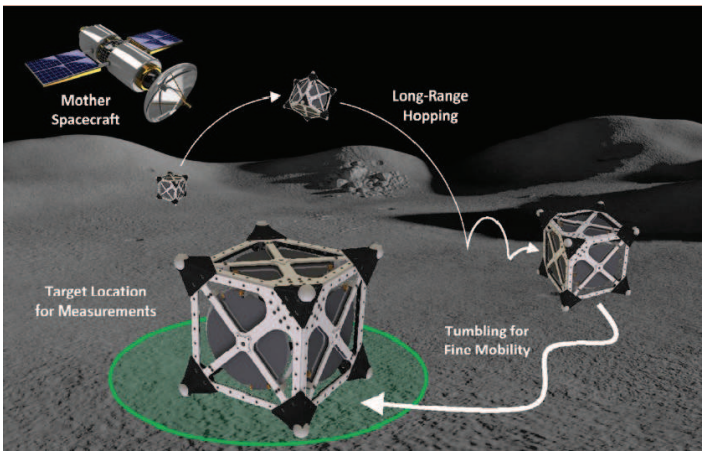


Fig. 4.1 Spacecraft/rover hybrids concept for the exploration of comets, asteroids and small bodies. *Image courtesy of [34]*

have to rely on a monocular camera, moreover, due to the limited knowledge of the asteroid gravity model, the possibility to use an inertial navigation system is limited. A key aspect that had led the choice of the algorithm is the presence of loop closure and place recognitions mechanisms, these mechanisms are the basis of the localization on the prior map and of the drift reduction during tumbling. Table 4.1 summarizes the requirements for the localization algorithm.

Asteroid and comets present a challenging environment for visual localization, due to large-scale changes, frequent occlusions, high-contrast, rapidly changing shadows and featureless terrain. Due to the absence of air scattering on asteroid/comets surface, shadows are very sharp, a visual characteristic of an area could change suddenly during small body rotation.

In Chapter 2.5.1 we have evaluated various open-source monocular visual SLAM software. Between them, ORB-SLAM2 has been chosen and adapted for this application. Indeed, the place recognition module of this software enables re-localization on a prior map and loop closure, which is an asset to reduce the uncertainty drift in SLAM trajectory estimation.

In Section 2.5.2 the process used for the realization of a bag-of-words vocabulary for visual localization on asteroids, comets, and moon is presented. Section 4.2 describes the main method of collaborative localization between orbiter and daughter spacecraft: the possibility to share the same map, that has been previously realized and stored.

The illumination related problems are presented in Section 4.3. Finally, a metrological evaluation of this method accuracy is presented. Tests have been performed by considering the exploration scenario of the hybrid rover/spacecraft on a medium size asteroid.

Table 4.1 Requirements for localization algorithm

Requirements for algorithm	Rationale
• Monocular vision	• Limited space on the platform • Weight constraints
Computationally efficient and Memory efficient	• Limited on-board computational capabilities
• Loop closure • Place recognition	• Drift reduction during tumbling • Localization of the platform
• Large topographic changes	• Operations in an asteroid environment
• Limited inertial navigation	• Asteroid gravity model unknown
• Possibility to set different cameras parameters • Map reuse capabilities	• Synergistic approach between mother spacecraft/tumbling rover
• Landmarks invariance to sunlight direction	• Different time between the mapping and the surface operation phase
• Fish-eye camera model	• Tumbling rover FOV camera, large image footprint
• Open source code	• Avoid license issues

4.1 ORB-SLAM Vocabulary parameters optimization

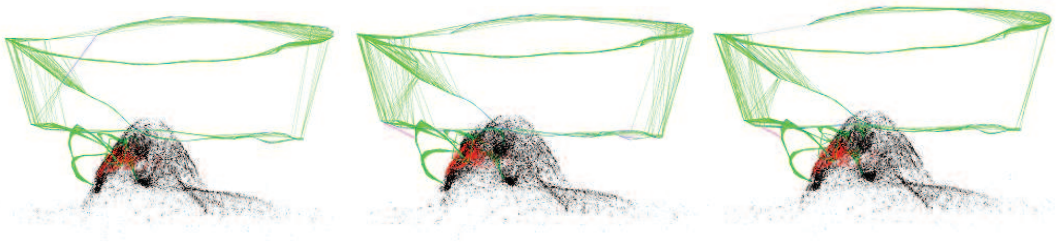
By processing the image sequences of an asteroid mock-up asteroid, place recognition shows poor performances and multiple failures in most of the cases if we use the default ORB-SLAM2 vocabulary. The consequence of a place recognition failure is the discarding of a loop closure. For this reason, we decided to build a custom vocabulary based on asteroid, comets and moon images.

A greater number of vocabulary branches and levels allow a more efficient place recognition. The number of generates nodes, for feature clusterization, is given by $n = k^L$, where k is the number of branches and L the number of levels. If we augment the number of nodes the average number of features for each cluster decrease, bringing some issue for a correct clusterization. By adding some other images to our asteroid vocabulary like the lunar mock-up images, MRO images and images from Mercury we obtain a vocabulary composed by 1510 images.

We change the ORB extractor parameters in order to retrieve a number of features equal to 2000 for each image. Finally, we have trained two vocabularies one with 5 levels and 10 branches, and the other one with 6 levels and 8 branches. This two vocabulary will be used for testing ORB-SLAM2 performances. Table 4.2 shows the parameters of the generated vocabularies.

Table 4.2 Used vocabulary parameters and performances

	Default	**Custom 1**	**Custom 2**
N of training images	10000	1510	1510
Features/Images	1000	2000	2000
Vocabulary Tree Levels	6	5	6
Cluster/Level	10	10	8
N of nodes	10000000	100000	262144
Feature/Node	10	20	12
Loop closure	1	2	2
Frames before localization (constant illumination)	10	14	1
Frames before localization (rotating asteroid)	43	41	26



(a) Default (b) Custom 1 (c) Custom 2

Fig. 4.2 Vocabulary testing on asteroid mock up data-set

New vocabulary has been tested on the asteroid mock up data set, results shown in Figure 4.2. Using the default vocabulary the tracking is lost two times, instead, with our asteroid vocabulary, the track is lost only one time. With both the vocabulary ORB-SLAM2 algorithm is able to re-localize the camera. Performances between vocabulary 1, 2 and 3 are slightly different.

4.2 Relocalization on the Mothercraft Map

For the tumbling/hopping platform relocalization, mapping and saving functions have been added to ORB-SLAM2. The output of the saving function is a YML file with keyframes, mappoints and keypoints information. For a map composed by 206 keyframes and 16674 mappoints the file size is 35.7 MB. The file contains all the information needed to process a new sequence on the previous map. Figure 6 shows a scheme of that process.

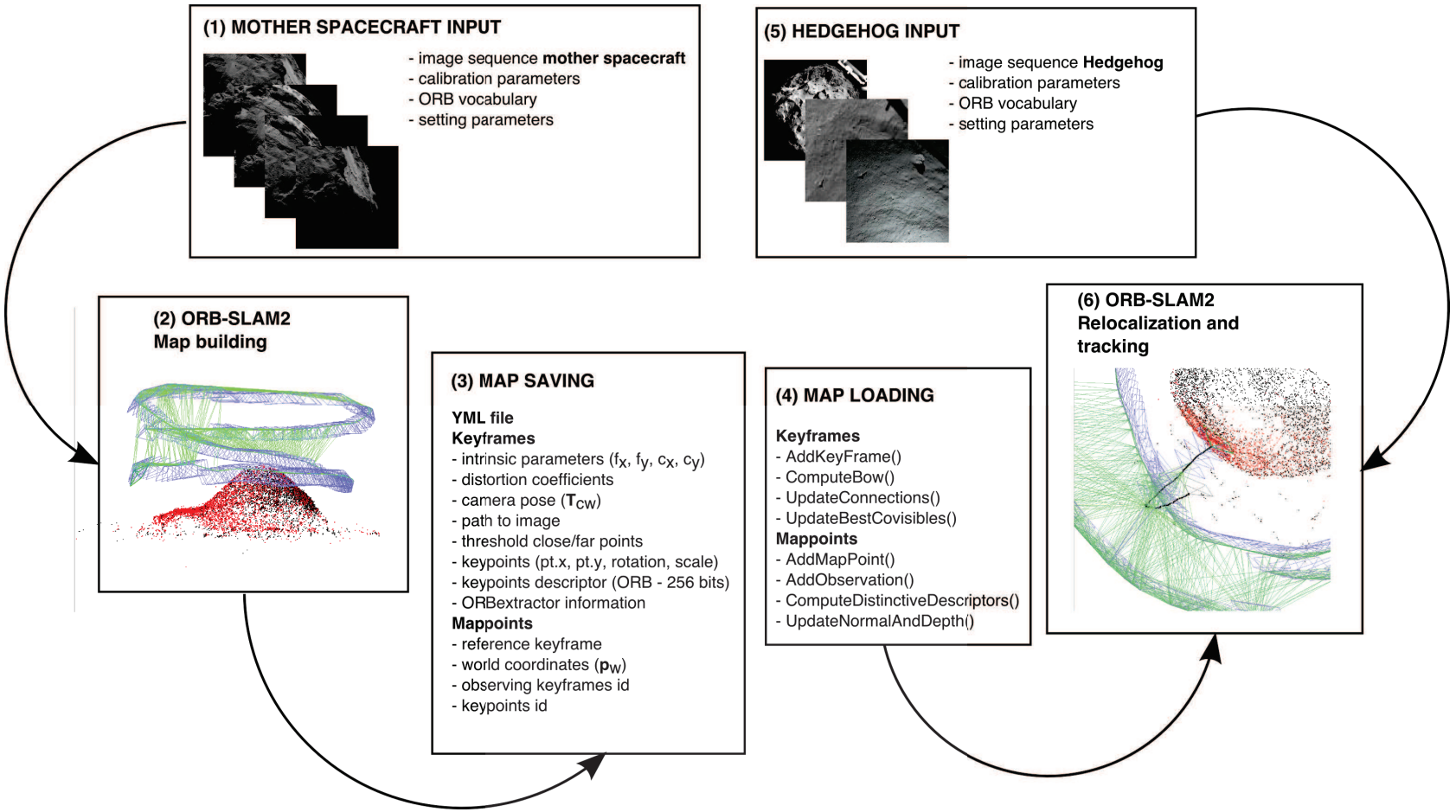


Fig. 4.3 Map saving and map loading function realized for the application.

The saving/loading functionality has permitted to us to evaluate the localization capabilities for a series of sequences taken at different distances from the comet/asteroid, the utilization of different cameras and different illumination conditions. The ORB-SLAM2 code has been modified to allow large FOV camera localization. The modification consists in:

- Increase the number of levels of the ORB extractor.
- Reduce threshold on min N of inliers for relocalization from 50 to 10.
- Do not allow scale filtering during tracking.

Figure 4.4 shows the performances of relocalization algorithm, is possible to relocalize camera with a huge difference between the camera intrinsic parameters.

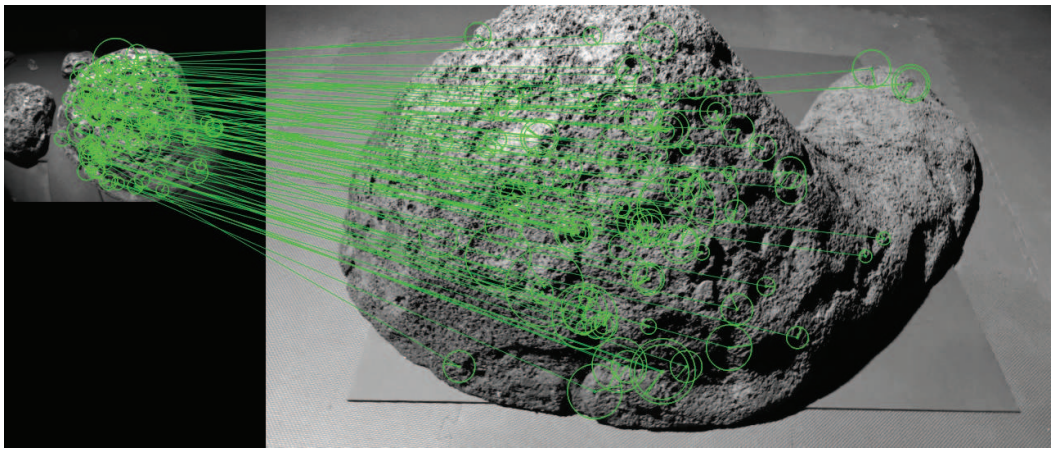


Fig. 4.4 Relocalization of a large FOV camera image ($f = 255 \quad [pixel/mm * mm]$, $640 \times 480 \quad [pixel]$) on a map realized with a narrow FOV camera ($f = 1527 \quad [pixel/mm * mm]$, $1920 \times 1060 \quad [pixel]$).

4.3 Robustness of ORB-SLAM to illumination

Comets and asteroids present a very challenging environment for feature matching and place recognition. The absence of the atmosphere and the related scattering effects led to the formation of net shadows, which move simultaneously to the asteroid rotation. The shadow of a rock or a crater could become a feature which position changes with the sunlight direction. Localization algorithm robustness to illumination changes is a key property for small body navigation and lunar Entry Descending and Landing problems [107].

For these reasons, a series of experiments under different illumination conditions has been performed. The experimental set-up allows us to rotating the asteroid and changing the light incidence direction. During the tumbling rover operations, the illumination conditions could be very different from the mapping phase. In order to evaluate at which angle the localization on the previous map fail a series of new debug functionalities have been introduced into ORB-SLAM2.

Figure 8 shows the relocalization of the rotating asteroid sequence (illumination changes) on a previously recorded map. After a rotation of about 12 deg, there are 13 features matching, after about 22deg there are no more features matching and the track on the previous map is lost. The frame to frame tracking still works in the new sequence.

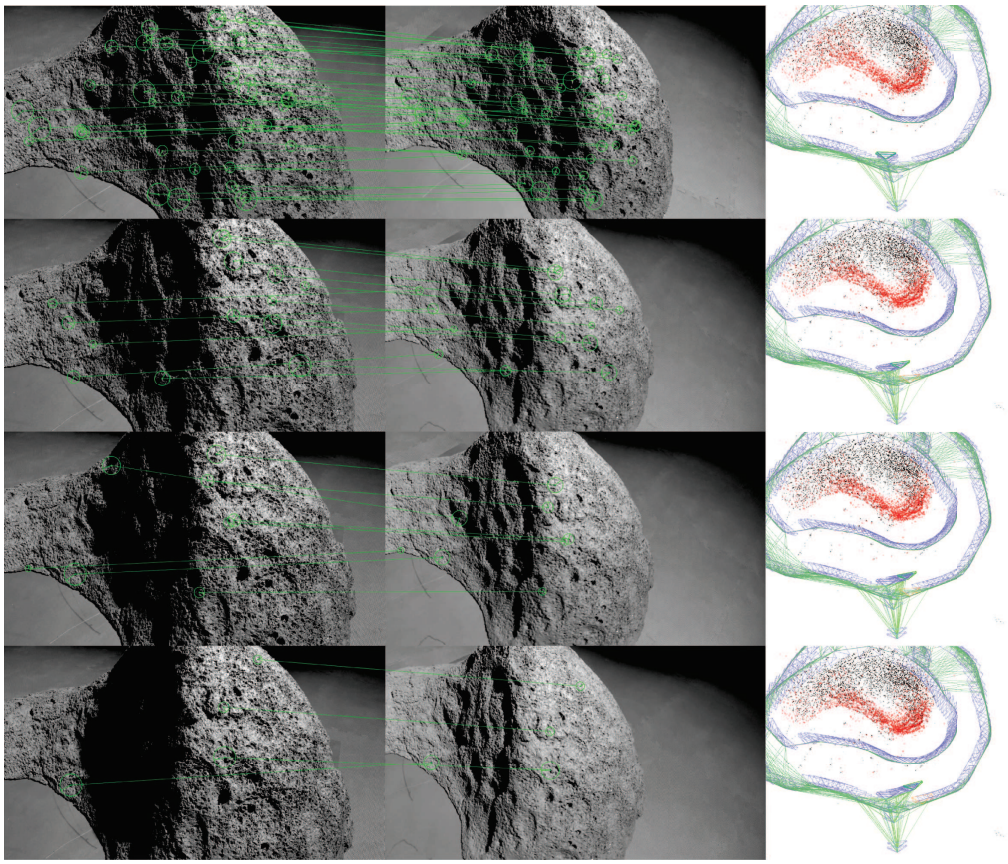


Fig. 4.5 Feature matching between the keyframes of the rotating asteroid sequence (left) and the map keyframes (center). Map keyframes have been realized by keeping the illumination conditions constant. The right image shows the new frames localized on the map.

4.4 Mapping using prior camera pose

During the mapping phase the pose of the spacecraft relative to the asteroid is known by other measurements [29, 30]. During the experiments with the asteroid mock-up, the prior pose of the camera will be estimated thanks to a Vicon system.

In order to add this functionality, ORB-SLAM2 code has been modified. Now is possible to store the frame pose to a text file (.yml) and load into the Frame class the image associated camera pose. In order to optimize the frames pose by knowing the prior pose position we have to add the prior adjustment thread.

The Frames that have a prior pose are "promoted" to keyframes and sent to the prior adjustment thread. Table 4.3 shows the parallels threads that track and optimize the Frame poses and the map points position with the addition of the "Prior adjustment" thread.

Table 4.3 ORB-SLAM2 threads. The "prior adjustment" thread has been added to adjust camera's trajectory when there is a prior knowledge of the camera pose.

Tracking	**Local Mapping**	**Loop Closing**	**Global Bundle Adjustment**	**Prior Adjustment**
• Process new frame	• Check if there are keyframes in the queue	• Check if there are keyframes in the queue	• This thread is launched from the Loop Closing thread	• Check if there are keyframes in the queue
• Extract ORB features	• BoW conversion and insertion in Map	• Detect loop candidates and check co-visibility consistency (3D-to-3D)	• Perform the Global Bundle Adjustment, only the initial keyframe is fixed	• Perform global pose graph optimization
• Initial pose estimation from last frame or Relocalization (2D-to-3D)	• Check recent mappoints	• Compute similarity transformation $[sR \mid t]$		• Perform a global bundle adjustment considering prior poses
• Track local map	• Triangulate new mappoints	• Perform loop fusion and global pose graph optimization		
• Check if we need to insert a new keyframe (local mapping is busy?)	• Find more matches in neighbour keyframes and fuse point duplications • Local Bundle Adjustmentm • Check redundant local keyframes	• Launch a new thread to perform Global Bundle Adjustment		
• Run continuously	• Run continuously	• Run continuously	• Lunched if a Loop Closure is detected	• Run continuously
	• Stopped if a Loop Closure is detected	• If a Loop Closure is detected send a stop signal to Local Mapping, wait until Local Mapping has effectively stopped	• Lunched if a Loop Closure is detected	• If a new keyframe is added in the queue send a stop signal to Local Mapping, wait until Local Mapping has effectively stopped
	• Stopped if a new keyframe is added in Prior Adjustment thread	• If a Loop Closure is detected abort the Bundle Adjustment thread		• If a new keyframe is added in the queue abort the Bundle Adjustment thread

The prior adjustment thread performs a pose graph optimization over the essential graph. Both the prior pose of the prior keyframes and the pose issued by the tracking phase are added into the optimization thread. The prior pose of the keyframes is fixed. A $\mathrm{Sim}(3)$ transformation is added between the prior pose and the tracked pose, other keyframes are connected each other by means of the essential graph, they are connected by $\mathrm{Sim}(3)$ transformations. This optimization distributes the error between the keyframe prior pose and the tracked pose over the essential graph.

An error in an edge (the connection between two keyframes) is defined as:

$$\mathbf{r}_{i,j} = \log_{\mathrm{Sim}(3)}(\mathbf{S}_{i,j}\mathbf{S}_{j,w}\mathbf{S}_{i,w}^{-1}) \tag{4.1}$$

where $\mathbf{S}_{i,j}$ is the relative $\mathrm{Sim}(3)$ transformation between the connected keyframes computed before the pose graph optimization, the scale is set to 1. As shown by [108] $\mathrm{Sim}(3)$ is a Lie group and $\mathfrak{sim}(3)$ is the corresponding Lie algebra, represented by a 7-vector $(\boldsymbol{\omega}, \boldsymbol{\nu}, \sigma)$. $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ is the axis-angle representation of the rotation, $\boldsymbol{\nu}$ is the rotated version of the translation $\mathbf{t}$ and $\sigma = \log s$, where s is the scale. The exponential map from $\mathfrak{sim}(3)$ to $\mathrm{Sim}(3)$ is given by:

$$\mathbf{S} = \exp_{\mathrm{Sim}(3)} \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{\nu} \\ \sigma \end{pmatrix} = \begin{bmatrix} sR & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{4.2}$$

Also, an inverse relation exists:

$$\begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{\nu} \\ \sigma \end{pmatrix} = \log_{\mathrm{Sim}(3)} \begin{bmatrix} sR & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{4.3}$$

$$\boldsymbol{\omega} = \frac{\theta}{2\sin\theta}(R - R^{\top}) \tag{4.4}$$

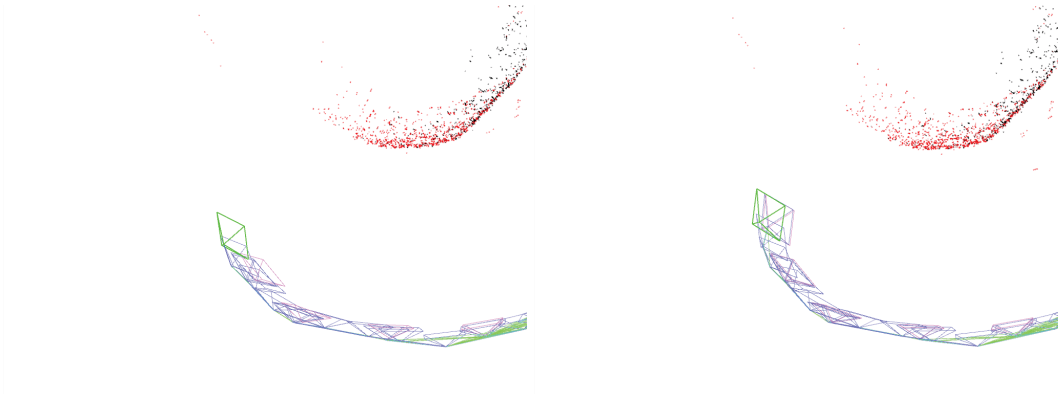
where

$$\theta = \arccos\left(\frac{\mathrm{tr}(R) - 1}{2}\right) \tag{4.5}$$

The rotated version of the translation $\mathbf{t}$ is given by:

$$\boldsymbol{\nu} = W^{-1}\mathbf{t} \tag{4.6}$$

$$W = \frac{a\sigma + (1-b)\theta}{\theta(\sigma^2 + \theta^2)}(\boldsymbol{\omega})_{\times} + \left(c - \frac{(b-1)\sigma + a\theta}{\sigma^2 + \theta^2}\right)\frac{(\boldsymbol{\omega})_{\times}^2}{\theta^2} + cI \tag{4.7}$$



(a) Before Pose Graph Optimization (b) After Pose Graph Optimization

Fig. 4.6 Keyframe poses before and after the prior adjustment, the prior pose of a keyframe is highlighted in magenta

$$a = s\sin\theta \qquad b = s\cos\theta \qquad c = \frac{e^{\sigma}-1}{\sigma} \tag{4.8}$$

The cost function of the essential graph optimization is given by:

$$\chi^2(\mathbf{S}_2,\ldots,\mathbf{S}_m) = \sum_{i,j} \mathbf{r}_{i,j}^{\top}\Omega_{\Delta\mathbf{S}_{i,j}}\mathbf{r}_{i,j} \tag{4.9}$$

where $\Omega_{\Delta\mathbf{S}_{i,j}}$ is the information matrix:

$$\Omega_{\Delta\mathbf{S}_{i,j}} = \mathrm{diag}(\sigma_r^2,\sigma_r^2,\sigma_r^2,\sigma_t^2,\sigma_t^2,\sigma_t^2,\sigma_s^2) \tag{4.10}$$

σ_r, σ_t, σ_s are respectively the rotation, the translation and the scale standard deviation. These values are settable in that version of ORB-SLAM code. In the results section we will show how these parameters influence the map construction. In the case of the connection between the prior keyframe pose and the tracked pose the $\mathrm{SE}(3)$ transformation between the two is set to the identity, and the scale to 1. In ORB-SLAM2 the optimizations are performed using g2o libraries [71].

After the essential graph optimization, a global bundle adjustment is performed. An $\mathrm{SE}(3)$ edge is added between the keyframe and its corresponding prior pose. The prior information of the prior keyframes is also taken into account during the windowed bundle adjustment.

4.5 Tests and results

A series of experiment in order to test ORB-SLAM2 capabilities to localize a large FOV camera on a map realized with a narrow FOV camera have been done. The large FOV camera represents the tumbling rover camera. Figure 4.7 shows a concept of tumbling/hopping rover operations. During phase *A* it is possible to localize the rover on the orbiter prior map. Phase *B* highlights rover operations, tumbling movements and small hops, during that phase the tracking is kept relative to previous frames, the estimate poses uncertainty grows with the traveled distance. In order to reduce the collected uncertainty, a second big hop is performed in order to close the loop on the satellite map, phase *C*. During tests, the localization capabilities have been tested also by changing the illumination conditions.

The mapping phase will be limited to the planar case; we assume that the orbital plane of the mother spacecraft will be similar to the equatorial plane of the small body. The comet is rotating around its orbital plane perpendicular direction and during the mapping phase the mother spacecraft is adequately far away to be considered not influenced by the asteroid gravity, so the distance between the spacecraft and the asteroid centre does not change.

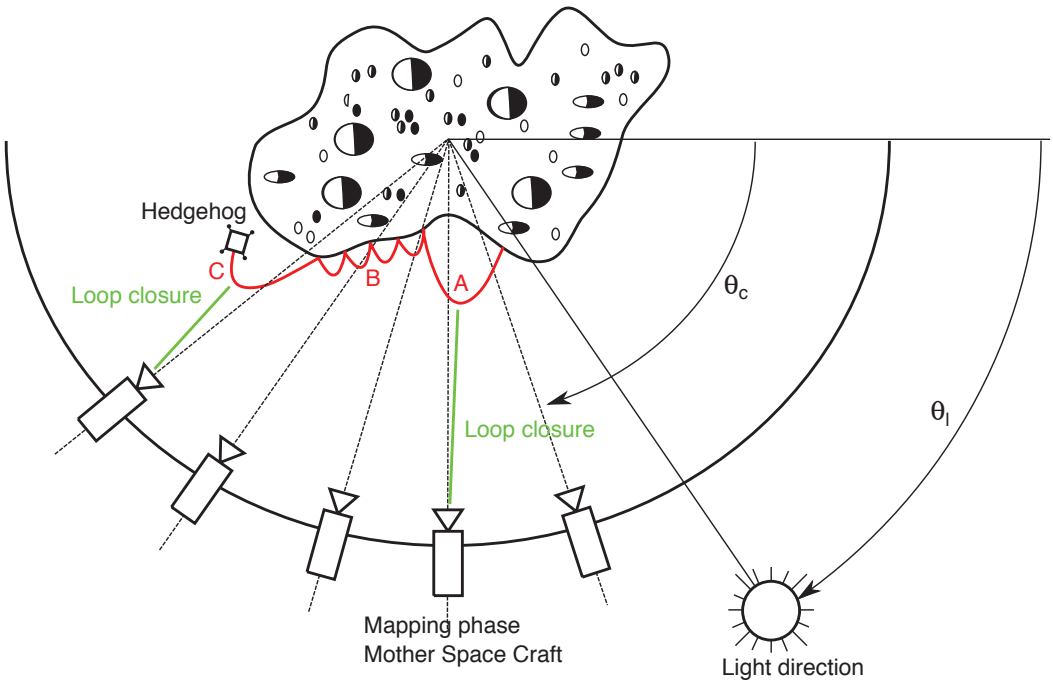


Fig. 4.7 Scheme of the experimental set-up.

As reference we consider a target comet which has a diameter of 1 km, we want to scale the localization problem to the mock-up asteroid, which diameter is 1 m. Table 4.5 shows the image footprint, the ground resolution, and how many images are necessary to map the

asteroid diameter with a camera that has similar properties to the candidate rover camera. For comparison, Rosetta's camera properties are highlighted in Table 4.4. In order to scale the problem to the asteroid mock up we have considered that to simulate the 10 m hop the image footprint need to be 1/10 of the asteroid diameter. Table 4.6 shows the results.

Table 4.4 Rosetta's cameras parameters.

	NAC	**WAC**
Field of view [deg]	2.20×2.22	11.35×12.11
Image size	2048×2048	2048×2048
Footprint @ 20 [km]	0.77×0.78 [km]	4.01×4.29 [km]
Footprint @ 100 [km]	3.84×3.88 [km]	20.07×21.46 [km]
Resolution @ 20 [km]	0.4 [m/pixel]	2 [m/pixel]
Resolution @ 100 [km]	2 [m/pixel]	10 [m/pixel]

Table 4.5 Performances of the tumbling rover camera, 160 deg FOV 640×480 pixels.

Distance	**Image footprint**	**Resolution**	**Images to cover a 1 km diameter asteroid**
10 [m]	0.11×0.085 [km]	0.2 [m/pixel]	10
50 [m]	0.57×0.42 [km]	0.9 [m/pixel]	2
100 [m]	1.13×0.85 [km]	1.8 [m/pixel]	1

Table 4.6 Large FOV camera performances on the asteroid mock up.

Hop maximal altitude	**Image footprint to map the asteroid mock up**	**Distance from asteroid mock-up**
10 [m]	0.11 [m]	0.01 [m]
50 [m]	0.57 [m]	0.05 [m]
100 [m]	1.13 [m]	0.10 [m]

The large FOV camera and the asteroid rotation will be tracked by means of a Vicon system, which accuracy is of 2 mm. This is a limit of the experimental set up because the lowest hops peak is of the order of 10 mm. The camera employed to create the asteroid mock-up map is a GoPro Hero4 Black, used in "Narrow field of view" mode. The resolution of images captured by the camera is 1920×1080 pixel. The image sequence for the localization experiments are listed below.

Sequence 1 – Mapping at fixed illumination conditions The narrow-angle FOV camera (GoPro Hero 4) is mounted on a cart that is manually moved around the steroid mock up in an "orbit like" way. The light direction is parallel to the asteroid orbital plane. Three mapping distances have been tested. A 30 fps video sequence is captured. The motion of the camera is recorded by a Vicon system. The knowledge of the pose is exploited in order to create the prior map.

Sequence 2 – Relocalization on the previous map – fixed illumination conditions Tumbling/hopping rover camera (160 deg FOV 640×480 pixels) is moved from the asteroid surface up to the mapping orbit. The motion of the camera is tracked by a Vicon system in order to have the ground truth and evaluate relocalization accuracy.

Sequence 3 – Mapping by changing the illumination conditions The narrow-angle FOV camera (GoPro Hero 4) is mounted on a tripod, the distance between the tripod and the asteroid is kept constant. The asteroid is rotated by keeping the light source angle of incidence constant. This allows the formation of a circular orbit around the asteroid. A 30 fps video sequence has been captured. In the meanwhile, the pose of the asteroid is recorded with the Vicon system. In addition, the pose of the illumination source is tracked during the test.

Sequence 4 – Relocalization on the previous map – variable illumination conditions Tumbling/hopping rover camera (160 deg FOV 640×480 pixels) is moved by performing multiple hops on the comet surface. The relocalization capabilities are tested by loading a map with different illumination conditions. The images are taken by a video sequence.

The used camera has been calibrated with the Zhang method described in [45]. Calibration results are summarized in Table 4.7.

4.5.1 Vicon/ORB-SLAM synchronization

In order to analyze algorithm performances and generate the prior map, the timestamp of the camera need to synchronize to the Vicon timestamp. Moreover, the trajectory output of the two systems are in a different reference frame, that need to be aligned. The ORB-SLAM2 trajectory is expressed with reference to the first keyframe pose.

Vicon/GoPro camera alignment In order align the Vicon timestamp with the GoPro timestamp and so align the two reference frame, first of all, an initial guess of the time-shift has been estimated manually by identifying some interest point of the time-displacement

Table 4.7 Camera experiment parameters.

GoPro Hero 4 camera parameters	
Size	1920×1080 [pixel]
Focal length	[1699.32, 1703.46] [pixel/mm·mm]
Principal point	[964.62, 604.94] [pixel]
Distortion Coefficient	[-0.2803, 0.1200, 0.0020, 0.0001, 0]
Large FOV camera parameters	
Size	640x480 [pixel]
Focal length	[274.13, 274.43] [pixel/mm·mm]
Principal point	[315.29, 235.52] [pixel]
Distortion Coefficient	[-0.2440, 0.0558, 0, 0, -0.0054]

plot, see Figure 4.9. Then the Vicon data is then interpolated on camera time stamp. We have now two set of 3D points that have the same dimensions, and that represent the trajectory in the two different frame of reference. To estimate the rotation, translation and scale difference between the two point set we have used the Horn's method [109]. The residual error is given by:

$$\mathbf{e}_i = \mathbf{r}_{W,i} - s_{C/W}\mathbf{R}_{C/W}(\mathbf{r}_{C,i}) - \mathbf{t}_{C/W} \tag{4.11}$$

where $\mathbf{r}_{W,i}$ and $\mathbf{r}_{C,i}$ are the camera position expressed respectively in the Vicon frame of reference and the GoPro camera frame of reference (ORB-SLAM2 frame of reference), $\mathbf{R}_{C/W}$ is the rotation matrix between the two frames, $\mathbf{t}_{C/W}$ the translation and $s_{C/W}$ the scale. We will minimize the sum of squares of these errors, if n is the number of sampled poses:

$$\sum_{i=1}^{n} \|\mathbf{e}_i\|^2 \tag{4.12}$$

The time shift has been estimated by an exhaustive search, the candidate value is the one that gives the minimal error, see Figure 4.10. In Figure 4.11 it is depicted the overlapping between the two point set after the estimation of the roto-translation. It is possible to see that the two trajectories are not perfectly overlapped. This is because the camera optical center and the camera support tracked in the Vicon system are not coincident. A nonlinear optimization is performed to obtain a more accurate estimation of the transformation, the cost function considers also the translation between the camera optical center and the camera rig:

$$\mathbf{e}_i = \mathbf{r}_{W,i} - s_{C/W}\mathbf{R}_{C/W}(\mathbf{r}_{C,i}) - \mathbf{t}_{C/W} + \mathbf{R}_W\mathbf{t}_{C/R} \tag{4.13}$$

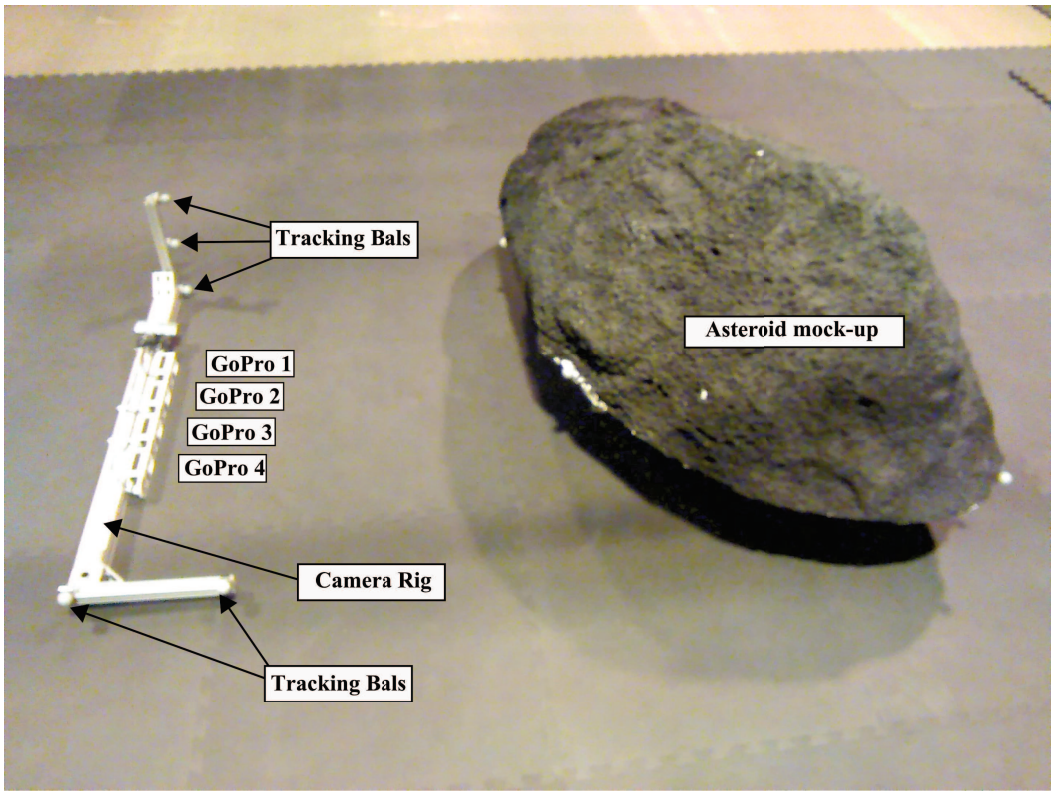


Fig. 4.8 Camera rig and experimental set-up.

where $\mathbf{t}_{C/R}$ is the translation between the camera optical centre and the camera rig and $\mathbf{R}_W$ is the pose of the camera in the Vicon frame of reference. Figure 4.12 shows the Cartesian components of the error between the camera position as measured by the Vicon and the estimated trajectory registered to the Vicon frame of reference, after the non-linear optimization.

For each mapping sequence, a .yml file with the correspondent Vicon poses has been created to allow the prior map generation. The RMS error is about 0.44 cm, the accuracy of the ground truth system (Vicon) is around 0.2 cm. The RMS angular error is equal to 0.2 deg along the three reference frame axis.

Vicon/Large FOV camera alignment A similar pipeline has been used for the large FOV camera alignment, but the time-shift estimation has been calculated using a different cost function. The large FOV camera is localized on a map that is scaled, rotated and translated to the Vicon frame of reference, so its frame of reference it is already up-to-scale, we need to estimate the translation between the camera optical center and the camera support tracked by the Vicon system. Considering that the Vicon clock was not perfectly aligned to the large FOV camera clock, we have performed an exhaustive search to retrieve the time shift and

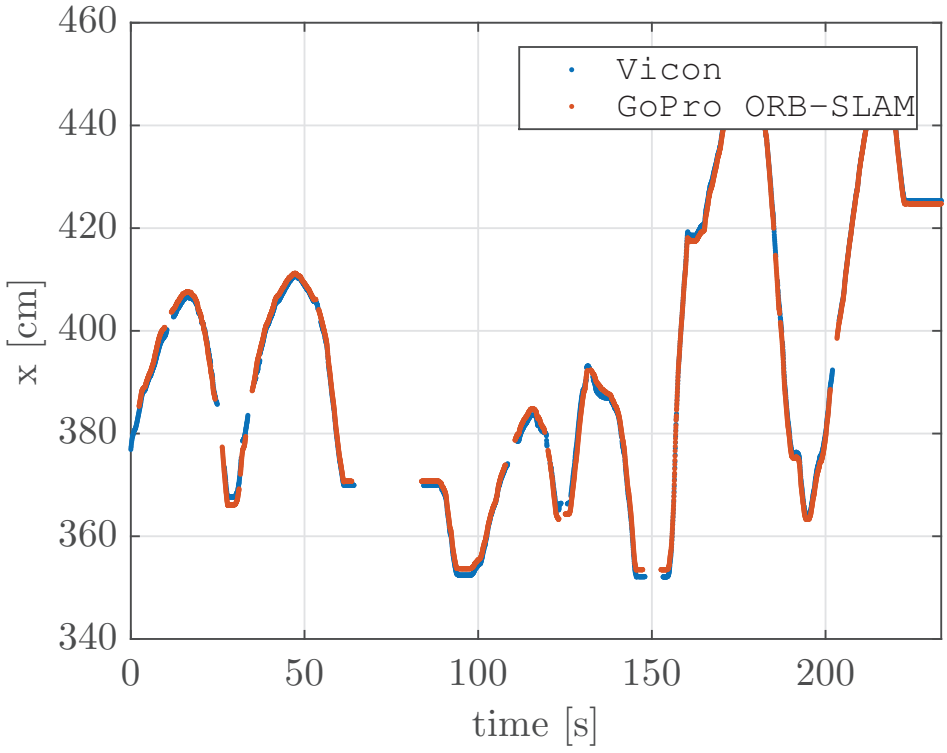


Fig. 4.9 Interpolation of the Vicon data on the GoPro camera timestamp.

the correct timestamp before performing the alignment. The cost function considers the translation between the camera optical center and the camera rig ($\mathbf{t}_{C/R}$):

$$\mathbf{e}_i = \mathbf{r}_{W,i} - \mathbf{r}_{C,i} + \mathbf{R}_W \mathbf{t}_{C/R} \tag{4.14}$$

Figure 4.14a shows the overlapping of the large FOV camera trajectory as detected by the Vicon system and estimated by ORB-SLAM, after the non-linear optimization and Figure 4.14b shows the difference between the two trajectories.

4.5.2 Mapping with priors

We have realized two maps, the first one with a sequence taken at a distance of about 160 cm from the asteroid mock up (*far mapping*), the second one is the result of three consecutive sequences closer and closer to the mock up, respectively 160 cm, 120 cm and 80 cm (*three distance mapping*). Figure 4.15 shows the position of the keyframes that have realized the map and the corresponding mappoints, for the two considered mapping sequences. During the mapping phase, we have changed the prior insertion rate and the information matrix values. A prior insertion rate of 30 means that we add a prior every 30 frames.

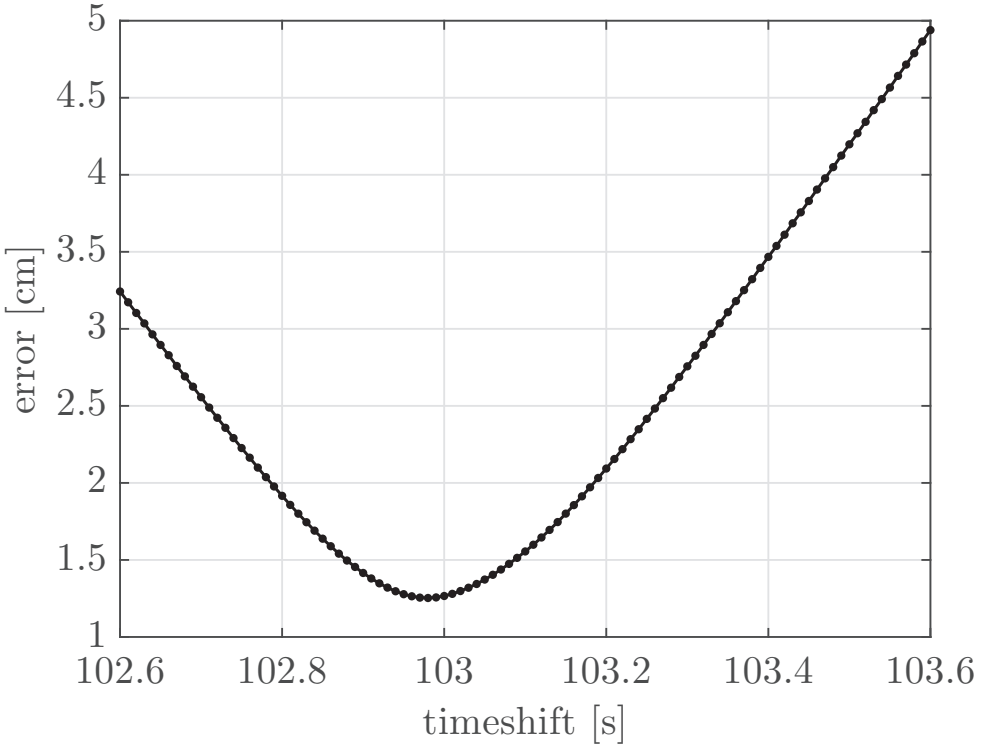


Fig. 4.10 The time shift used in the non-linear is the one that gives the minimal error.

Add prior information to the map is needed in order to scale the map to a meaningful length, and to have a common reference frame for accuracy analysis. Moreover, as we can see from Figure 4.16, the trajectory estimation error decreases by adding priors. At the same time the number of priors should not be to high, because there is the risk to over-constrain the optimization algorithm. We attempt to give an higher weight to the translational components of the prior pose (see information matrix Eq.4.10) because using the Vicon system the rotation estimation is dependent by the tracking balls baseline. Considering a tracking accuracy of the Vicon system of 0.2 cm and a baseline of 100 cm, the rotation accuracy is 0.3 deg.

In the *three distance mapping* sequence the median value of the trajectory estimation error is 0.14 cm, with a prior addition rate of 15 and $\sigma_r = \sigma_s = 10^{-9}$, $\sigma_t = 10^4$. With a prior insertion rate of 10, and $\sigma_r = \sigma_s = 10^{-9}$, $\sigma_t = 10^4$ the media error of the *far mapping* sequence is 0.12 cm. As we can see from Figure 4.16 the map optimization algorithm led the map to "break" due to numerical instabilities in two cases: in the *far mapping* sequence when the prior insertion rate $FPS = 10$, $\sigma_r = \sigma_s = 1$, $\sigma_t = 1$, and in the *three distance mapping* sequence when $FPS = 15$, $\sigma_r = \sigma_s = 10^{-3}$, $\sigma_t = 10^4$. We were not able to produce an accurate map with the *three distance mapping* when the prior addition rate $FPS = 10$.

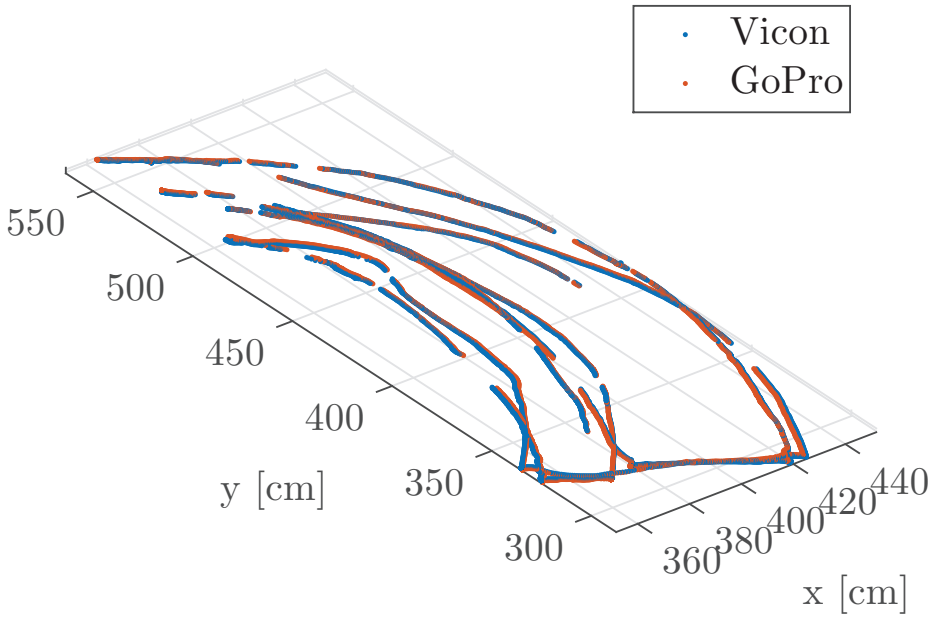


Fig. 4.11 Overlapping of the two trajectories using Horn's method.

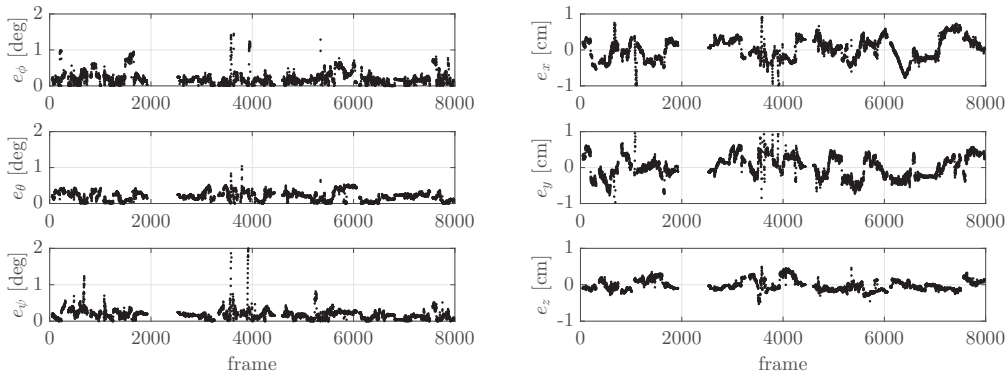


Fig. 4.12 (Right) Absolute value of the Cartesian components of the error between the camera position as measured by the Vicon and the estimated trajectory registered to the Vicon frame of reference, after the non-linear optimization. (Left) Angular error of the three Euler angles between the camera position as measured by the Vicon and the estimated ORB-SLAM trajectory

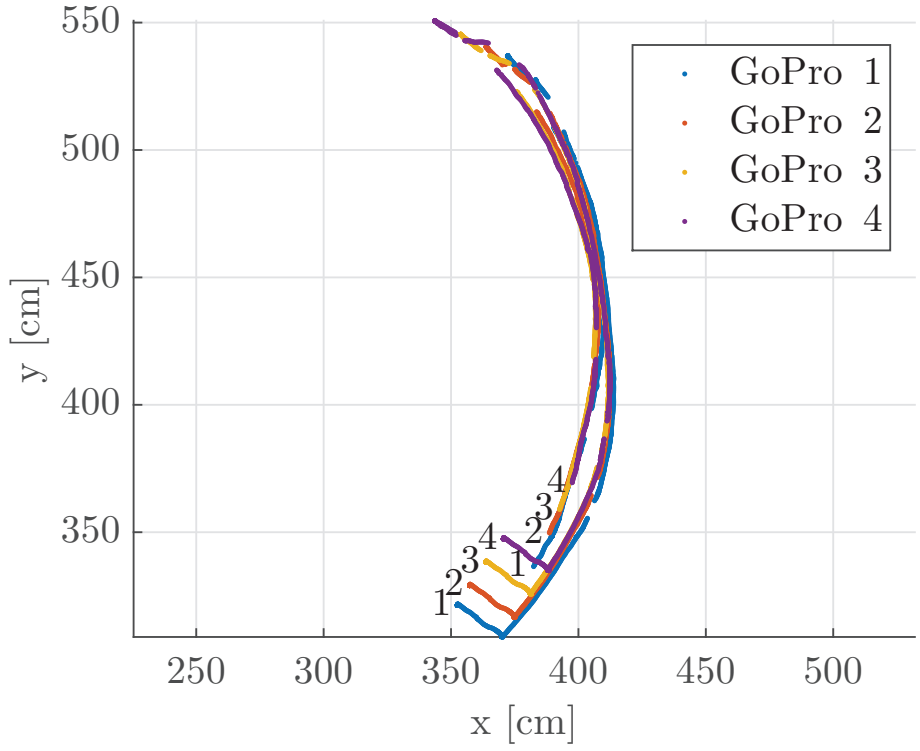
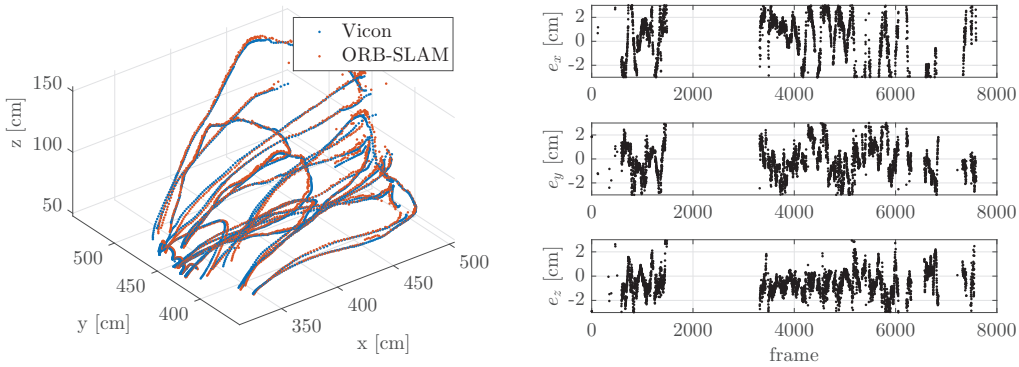


Fig. 4.13 Trajectory reconstruction of the four camera rig cameras.

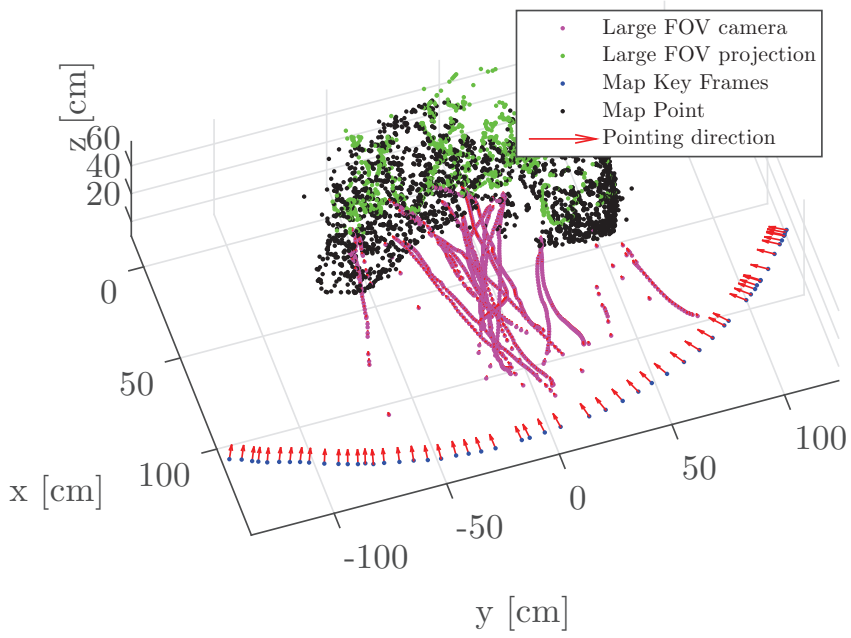


(a) (b)

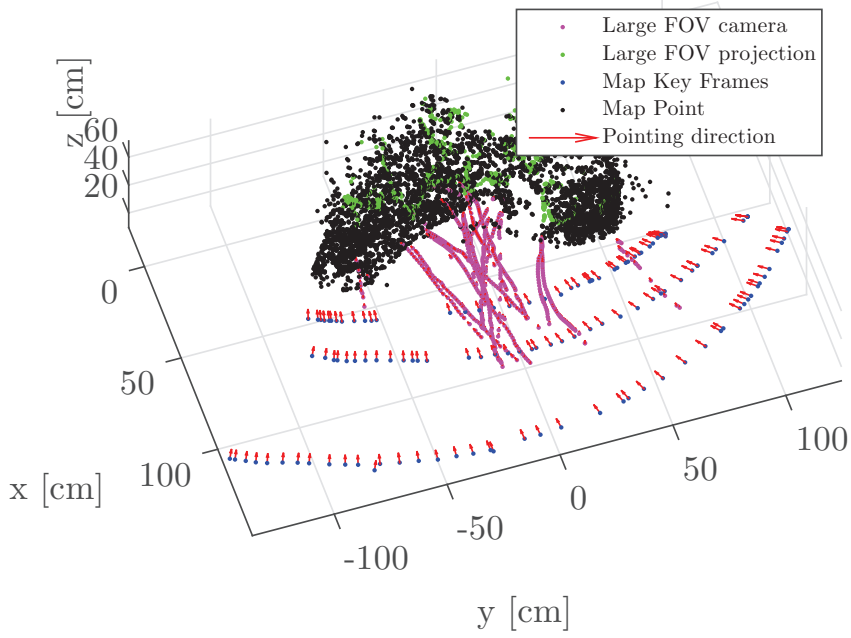
Fig. 4.14 (Left) Overlapping of the large FOV camera trajectory as detected by the Vicon system and estimated by ORB-SLAM, after the non-linear optimization. (Right) Cartesian components of the error between the large FOV camera position as measured by the Vicon system and the estimated trajectory registered to the Vicon frame of reference, after the non-linear optimization.

4.5.3 Relocalization accuracy

The relocalization accuracy of the large FOV camera on the saved map has been evaluated as a function of the distance from the asteroid, and of the off-pointing direction. The distance

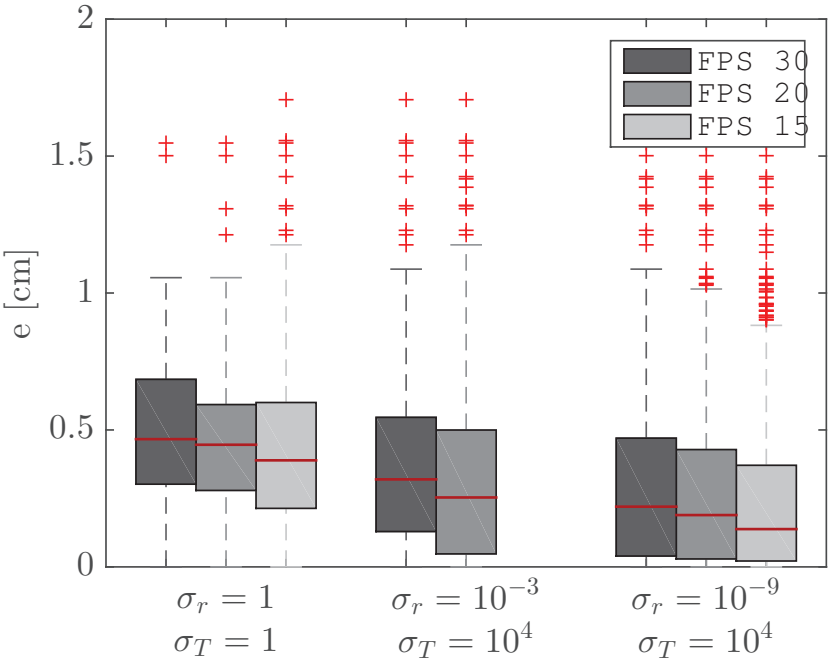


(a)

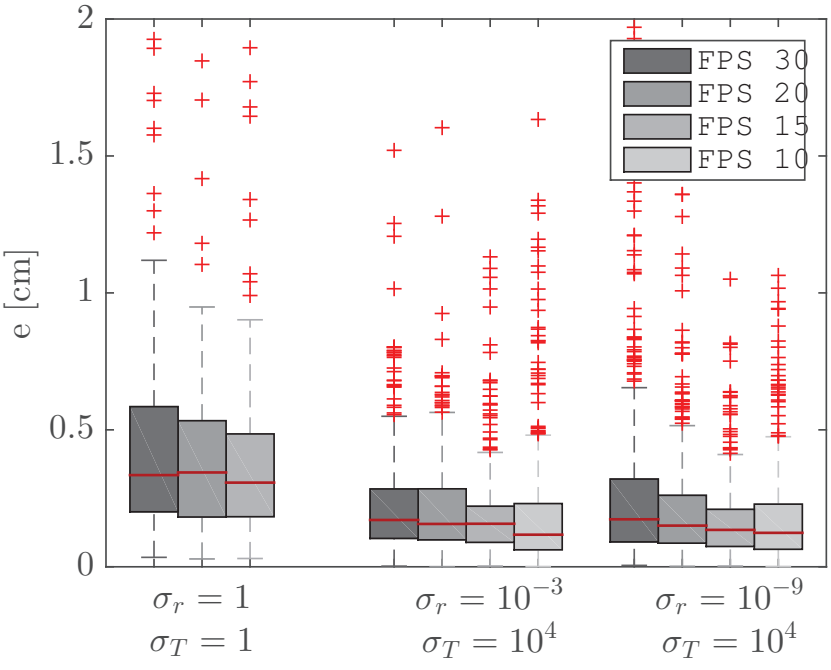


(b)

Fig. 4.15 (Top) Map created by processing the *far mapping* sequence. The position of the keyframes used to create the map are blue colored and the re-localized frames are colored in magenta. The optical axis direction is highlighted by a red vector. The priors are inserted every 15 frames and the information matrix values are $\sigma_r = \sigma_s = 10^{-3}$, $\sigma_t = 10^4$. (Bottom) Map created by processing the *three distance mapping* sequence. The position of the keyframes used to create the map are blue colored and the re-localized frames are colored in magenta. The optical axis direction is highlighted by a red vector. The priors are inserted every 20 frames and the information matrix values are $\sigma_r = \sigma_s = 10^{-9}$, $\sigma_t = 10^4$.



(a)



(b)

Fig. 4.16 Box diagram of the frames error during the mapping sequence after prior keyframes insertion. Comparison between different rate insertion and information matrix values. The median value is highlighted in red, distribution outliers are represented by a red cross. (Top) *far mapping* sequence. (Bottom) *three distance mapping*.

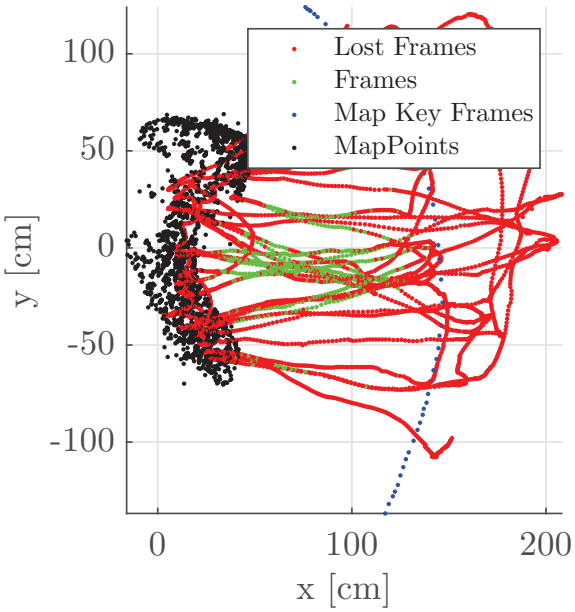
from the asteroid is the Euclidean distance from the frame position and his projection in the asteroid point cloud along the camera optical axis, as shown in Figure 4.15. The off-pointing direction is the angular deviation between the localized frame optical axes and the reference keyframe optical axis.

Figure 4.18 shows the percentage of localized frames as a function of the normalized distance from the asteroid. The 5% of frames that are located at a distance corresponding to the 0.13 of the asteroid diameter for all the tested sequences, which correspond to a height above the surface of 100 m for an asteroid with a diameter of 769 m. The localized frames percentage increases for the frames located closer to the comet surface if we use the *three distance* map, shown in Figure 4.15. By using the *far distance* map we arrive to localize the 32% of the frames which are located at a distance equal to 0.47 of the asteroid diameter.

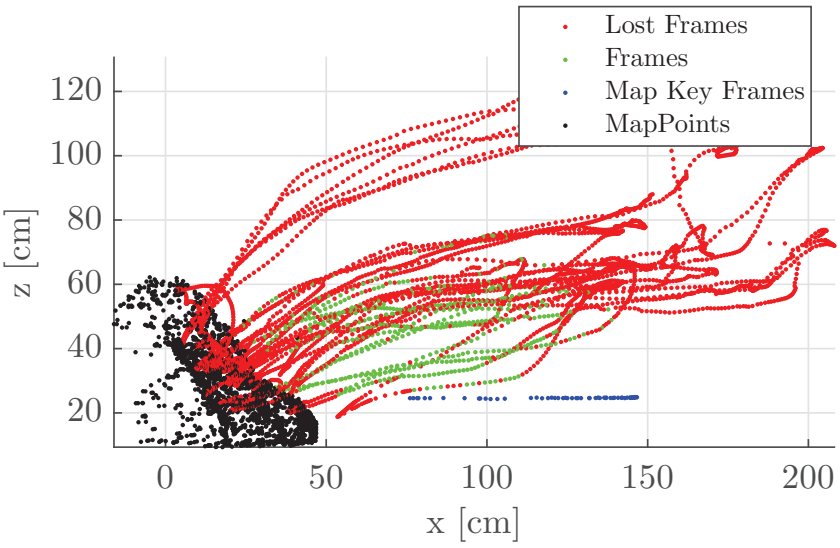
Localization performances have been evaluated by changing the number of map candidates keyframes returned by the BoW library. Keyframes which do not share enough words are discarded, in order to increase the keyframe candidates we decrease the percentage of minimum common words required. Figure 4.19a and 4.19b show the number of candidate map keyframes returned using the Bag of Word approach as function of the frames distance form the asteroid. The candidate keyframe number is filtered using two parameters k_{words} and k_{score}. k_{words} is used in order to compare the candidate keyframe only against those keyframes that share enough words. The minimum number of words that two frames should share is given by $\min_{words} = k_{words}\max_{words}$. Then the similarity score is computed and the remaining candidate keyframes are filtered again by retain only the keyframes that has the highest score, $\min_{score} = k_{score}\text{best}_{score}$. The computational efficiency has to be weighted against the possibility to localize a frame.

Frame localization on the map is performed by checking the geometric consistency of a list of candidate keyframes chosen using the frequency of repeated words. Geometric consistency is performed by using a PnP scheme embedded in a RANSAC scheme. In order to show the bag of word effects on the map localisation, and to estimate the "upper bound" of the number of frames that could be potentially localised on the map only with the geometric consistency test, we have done a series of tests by returning all the keyframes stored in the orbiter map. Results are shown in Figure 4.20.

By considering an average mapping distance of 160 cm from the comet, as for the *far map* sequence, and the camera intrinsic parameters of Table 4.7, the overlap between the two camera footprints takes place when the large FOV camera is located at 18.3 cm from the surface of the mock-up, which is equal to a normalized distance of 0.23. If the footprint of the camera to localize is smaller than the mapping camera footprint a part of the possible feature matches is throw out, so the localization becomes more difficult. We can see from



(a)



(b)

Fig. 4.17 Localization on 'far map', prior insertion rate = 15, $\sigma_r = \sigma_s = 10^{-3}$, $\sigma_t = 10^4$, and RANSAC iterations = 200. The localized frames are highlighted in green.

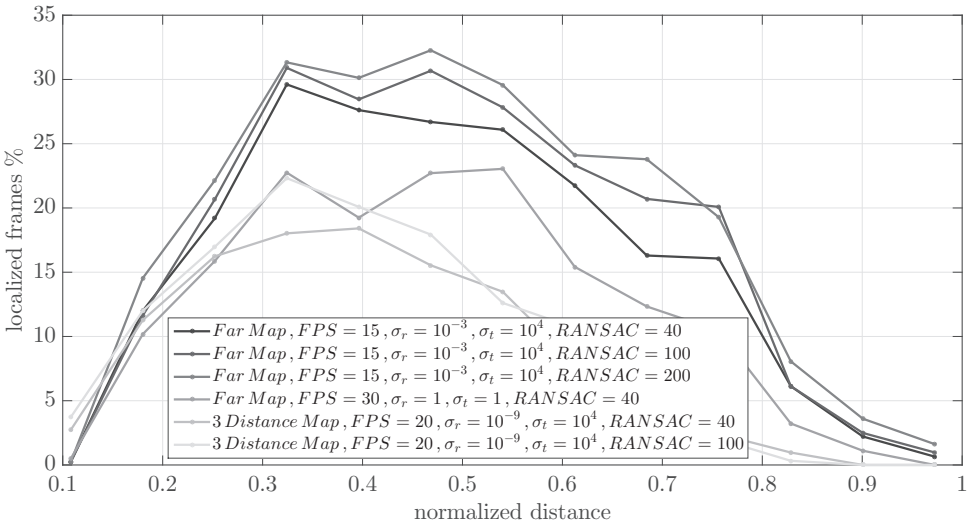
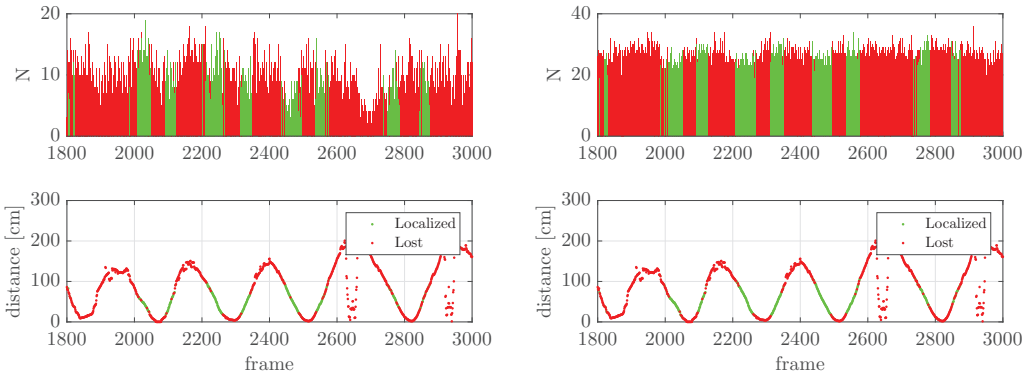


Fig. 4.18 Percentage of localized frames as a function of the normalized distance from the asteroid. The tests have been performed using different maps. The parameters of the keyframes selection are unchanged.

Figure 4.18 that the percentage of localized frames grows when the normalised distance value is between 0.3-0.6, which is correspondent to a large FOV camera footprint greater than the mapping images. When the normalized distance greater than 0.6 map tracked features cover only a small portion of the large FOV image, in this case the numerical errors are to high to pass the geometric consistency test.

Figure 4.22 shows how accurate is the localization algorithm performances as a function of the distance from the asteroid mock-up. The median value of the error is between 2 and 3 cm, which correspond to a normalized error between 0.014 and 0.022, the normalization is calculated again over the asteroid diameter. If the asteroid diameter is 535 m, like asteroid Itokawa, the localization accuracy will be between 7.5 and 11.8 m. We can notice that the accuracy slightly decreases with the distance from the asteroid. Figure 4.23 shows the normalized versions of Figure 4.22. As off-pointing angle we have considered the angle difference between the optical axis of the localized frame and the reference map keyframe optical axis. Figure 4.24 shows the localization accuracy and number of localized frames as a function of the off-pointing direction, for different map types and RANSAC iterations. Test conditions have allowed a limited number configuration which has the two optical axes aligned, see Figure 4.17. In all the tested conditions we were able to localize frames up to an off-pointing angle of 46 deg. The accuracy and the number of localized frames decrease with the off-pointing angle.



(a) (b)

Fig. 4.19 Number of candidate map keyframes returned using the Bag of Word approach, compared to the frames distance form the asteroid. The geometric consistency test is performed between the current frame and the candidates keyframes. The localization is performed over the *three sequence* map. (a) $k_{words} = 0.8$, $k_{score} = 0.75$, RANSAC=100 and (b) $k_{words} = 0.01$, $k_{score} = 0.1$, RANSAC=100. By reducing the minimum number of common words that a map keyframe have to share with the current keyframe the number of localized frame increases.

Figure 4.25 shows the box plot of the error for the two configuration which gave the highest number of localized frames. We can see that the error median value is slightly increasing as function of the distance from the asteroid.

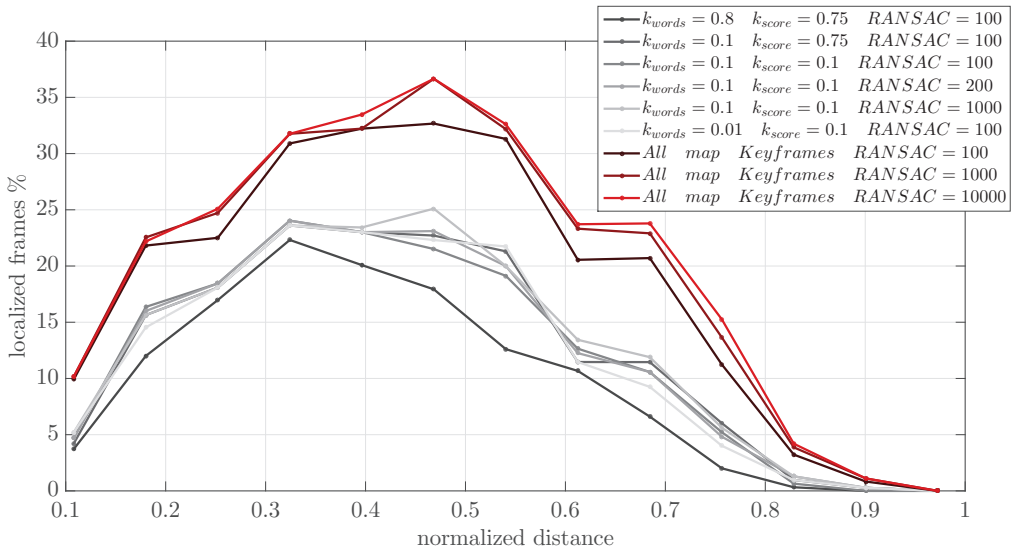


Fig. 4.20 Percentage of localized frames as a function of the normalized distance from the asteroid. The tests have been performed by changing the parameters of the candidate for localization selection. The used map is the *three distance map*. Red curves shows the localization performances by using all the keyframes of the map.

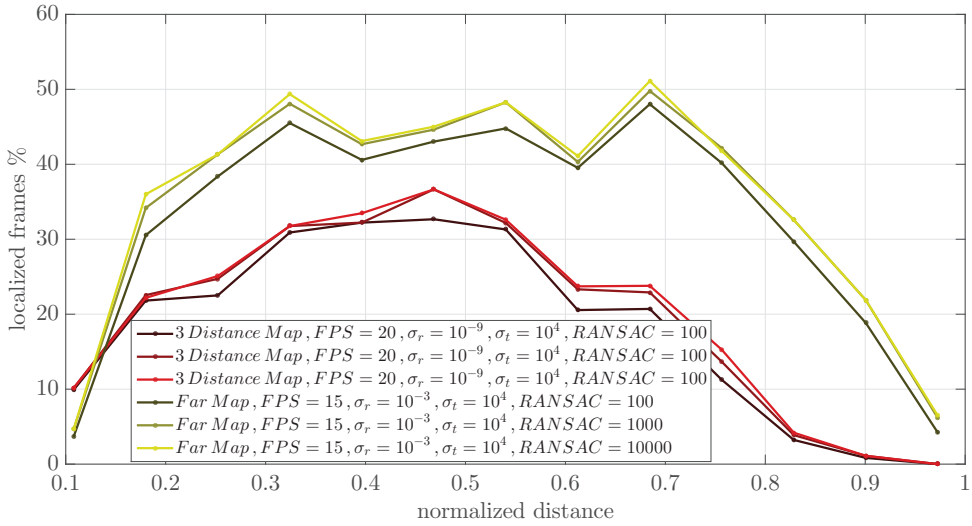


Fig. 4.21 Percentage of localized frames as a function of the normalized distance from the asteroid. "Upper bound" of the number of frames that could be relocalised by changing the map. Comparison between *three distance map* and *far map*.

Table 4.8 Localized frames lower bound distance from the asteroid mock-up and maximal off-pointing angle.

Map	**Lower bound distance**	**Lower bound distance normalized**	**Max off-pointing angle**
Far Map, $FPS = 15$, $\sigma_r = 10^{-3}$, $\sigma_t = 10^4$ RANSAC = 40	17.47 [cm]	0.13	57.03 [deg]
Far Map, $FPS = 15$, $\sigma_r = 10^{-3}$, $\sigma_t = 10^4$ RANSAC = 100	17.47 [cm]	0.13	53.25 [deg]
Far Map, $FPS = 15$, $\sigma_r = 10^{-3}$, $\sigma_t = 10^4$ RANSAC = 200	17.47 [cm]	0.13	56.45 [deg]
Far Map, $FPS = 30$, $\sigma_r = 1$, $\sigma_t = 1$ RANSAC = 40	18.55 [cm]	0.13	49.83 [deg]
3 Dist Map, $FPS = 15$, $\sigma_r = 10^{-9}$, $\sigma_t = 10^4$, RANSAC = 40	14.65 [cm]	0.11	46.50 [deg]
3 Dist Map, $FPS = 15$, $\sigma_r = 10^{-9}$, $\sigma_t = 10^4$, RANSAC = 100	14.43 [cm]	0.10	49.29 [deg]

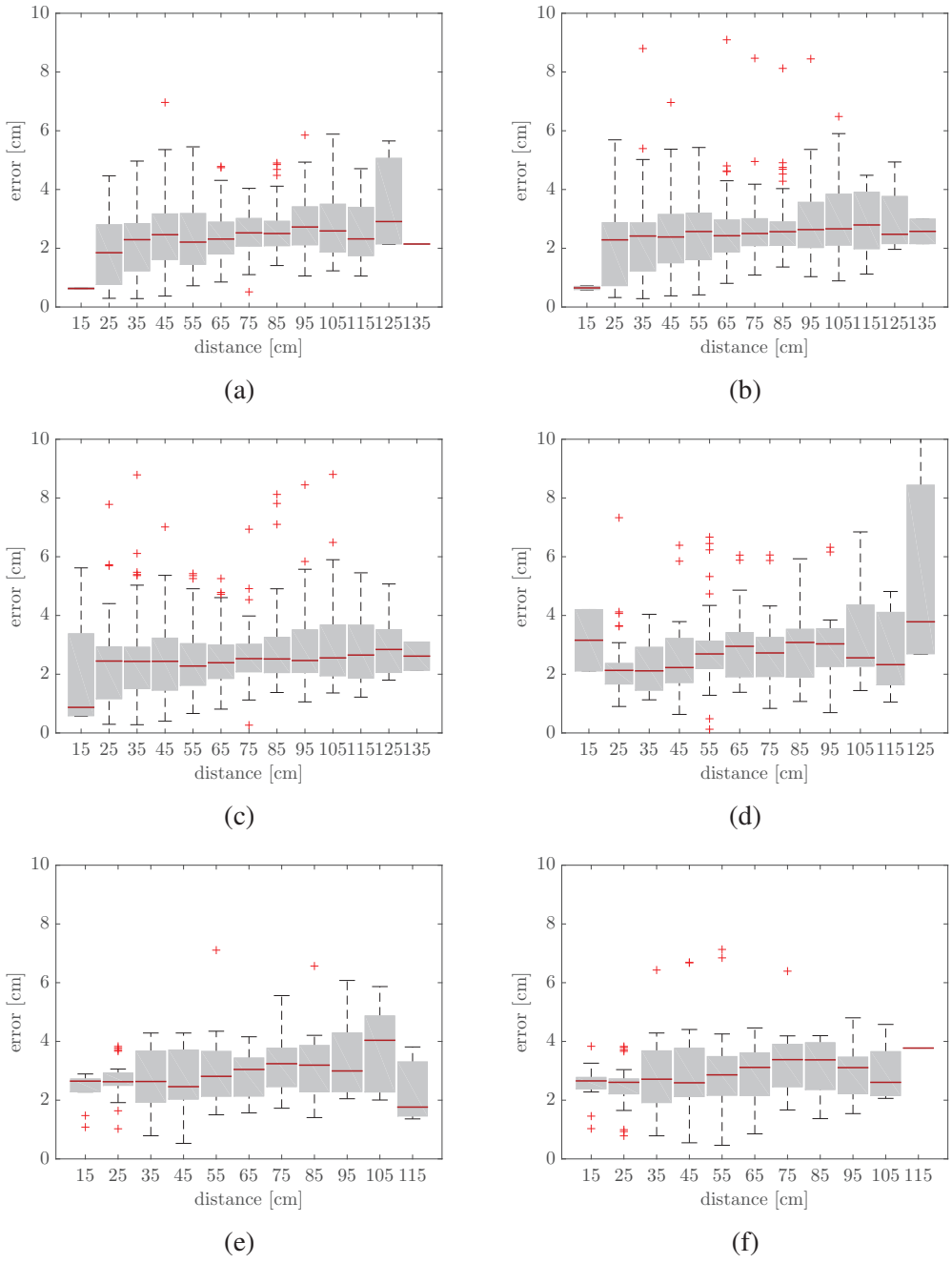


Fig. 4.22 Relocalization accuracy as a function of the distance from the asteroid mock-up surface, for different map types and RANSAC iterations. (a) (b) and (c) *far map* with $FPS = 15$, $\sigma_r = 10^{-3}$, $\sigma_t = 10^4$ and respectively 40, 100 and 200 RANSAC iterations. (d) *far map* with $FPS = 15$, $\sigma_r = 1$, $\sigma_t = 1$ and 40 RANSAC iterations. (e) and (f) *three sequence map* with $FPS = 15$, $\sigma_r = 10^{-9}$, $\sigma_t = 10^4$.

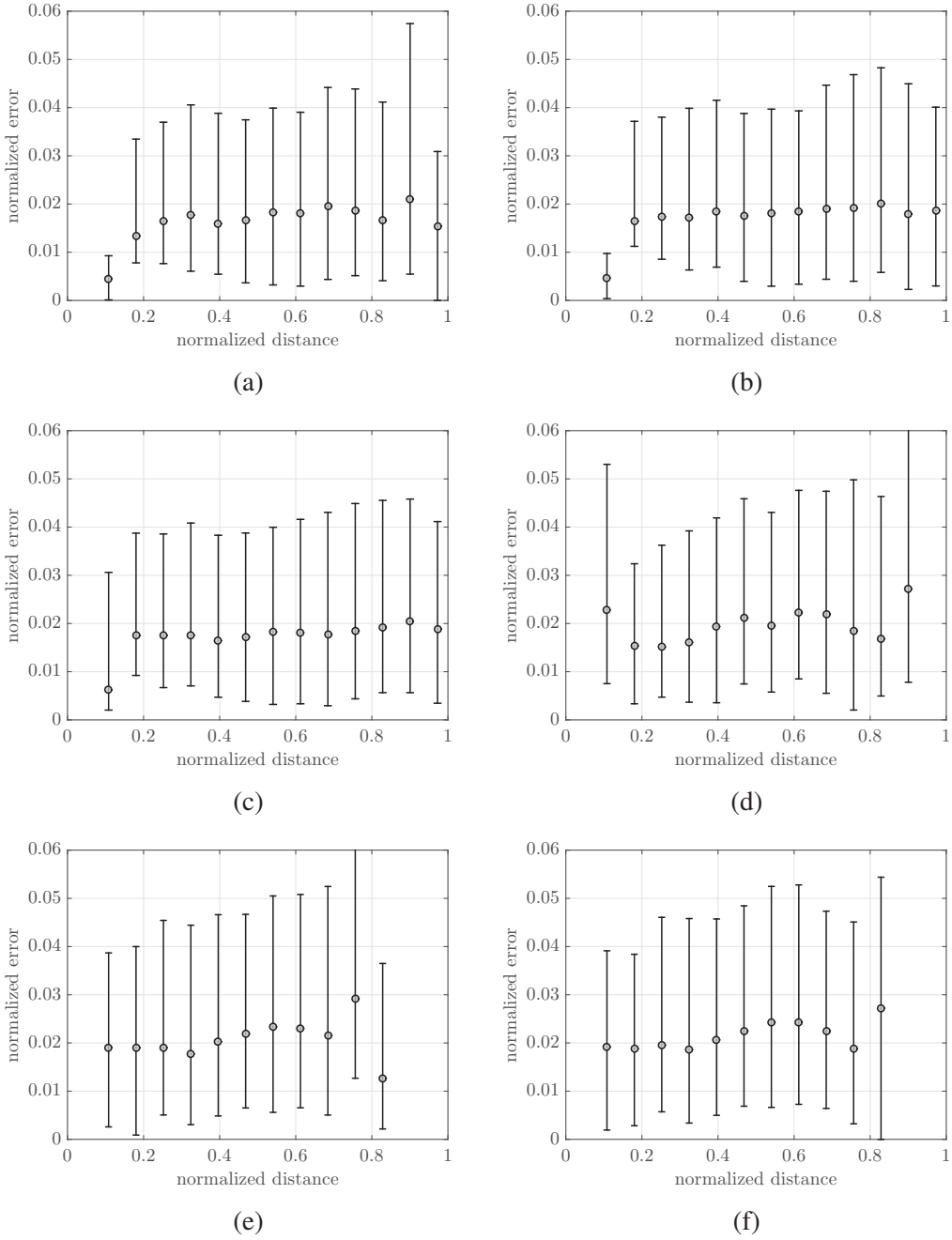


Fig. 4.23 Normalized relocalization accuracy as a function of the distance from the asteroid mock-up surface, for different map types and RANSAC iterations. (a) (b) and (c) *far map* with $FPS = 15$, $\sigma_r = 10^{-3}$, $\sigma_t = 10^4$ and respectively 40, 100 and 200 RANSAC iterations. (d) *far map* with $FPS = 15$, $\sigma_r = 1$, $\sigma_t = 1$ and 40 RANSAC iterations. (e) and (f) *three sequence map* with $FPS = 15$, $\sigma_r = 10^{-9}$, $\sigma_t = 10^4$.

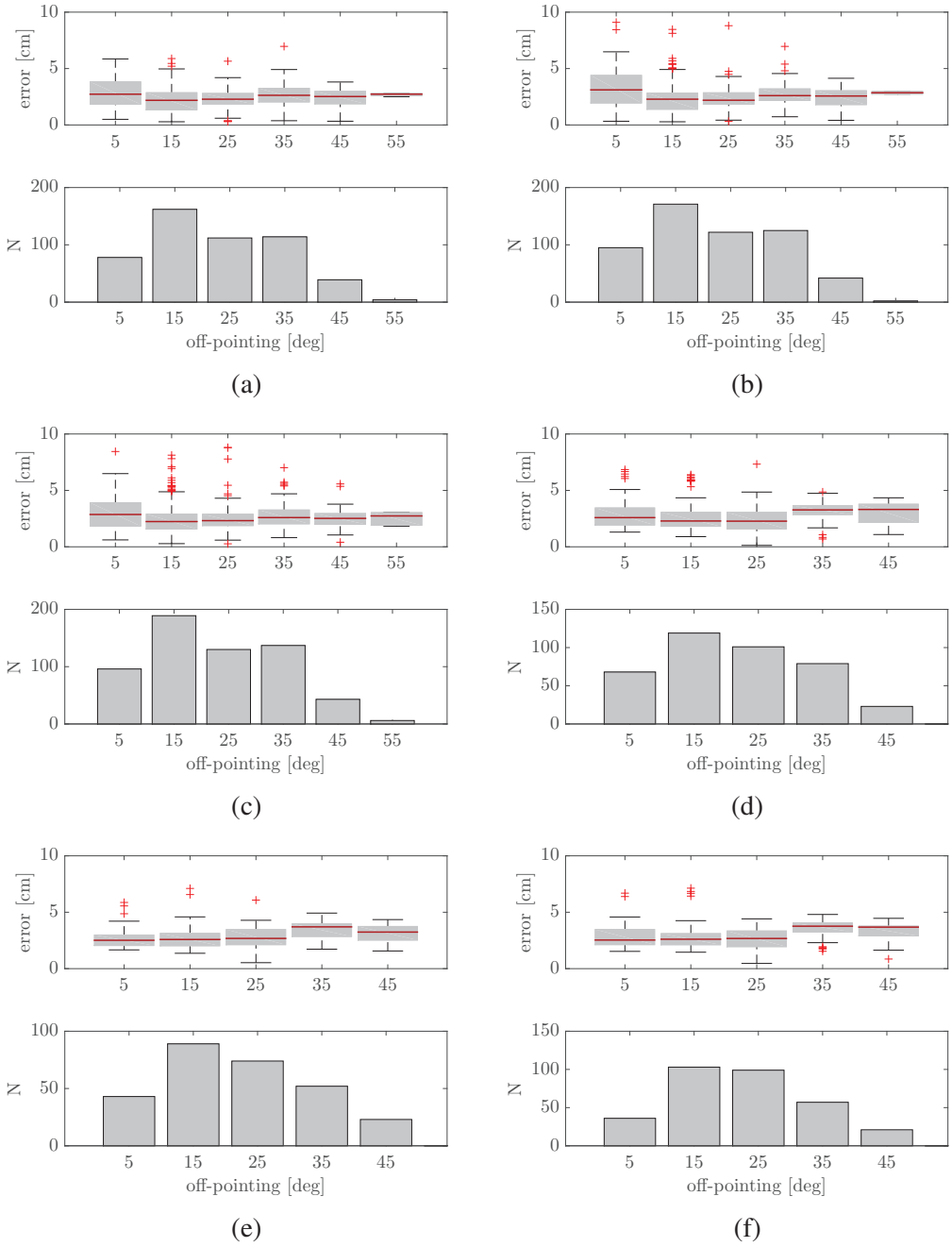
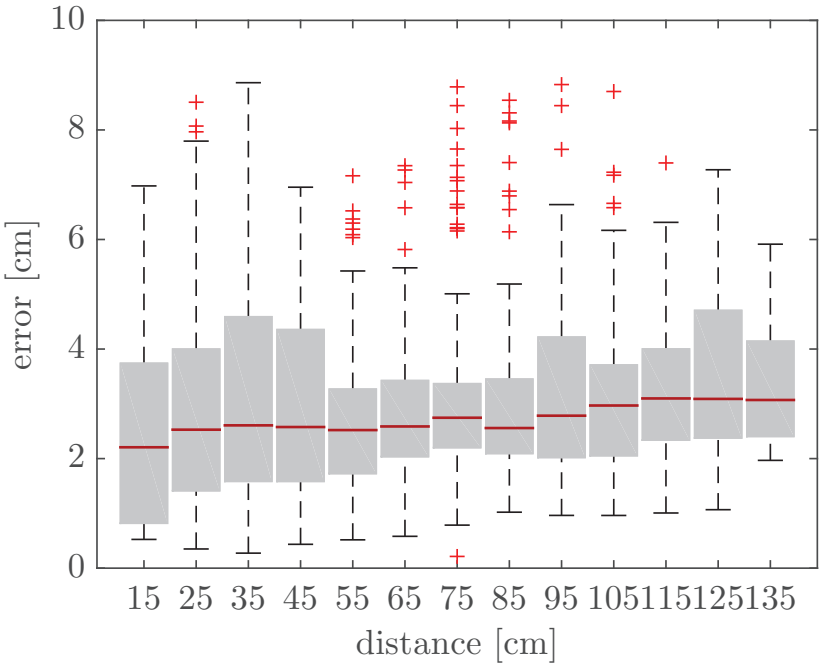
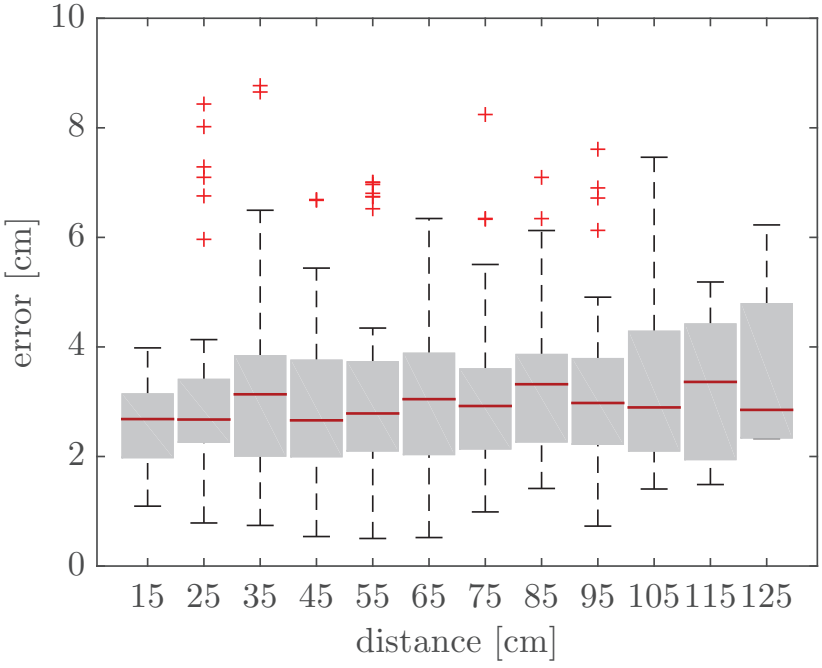


Fig. 4.24 Localization accuracy and number of localized frames as a function of the off-pointing direction, for different map types and RANSAC iterations. (a) (b) and (c) *far map* with $FPS = 15$, $\sigma_r = 10^{-3}$, $\sigma_t = 10^4$ and respectively 40, 100 and 200 RANSAC iterations. (d) *far map* with $FPS = 15$, $\sigma_r = 1$, $\sigma_t = 1$ and 40 RANSAC iterations. (e) and (f) *three sequence map* with $FPS = 15$, $\sigma_r = 10^{-9}$, $\sigma_t = 10^4$.



(a)



(b)

Fig. 4.25 Relocalization accuracy as a function of the distance from the asteroid mock-up surface, for different map types and RANSAC iterations = 10000. (a) *far map* with $FPS = 15$, $\sigma_r = 10^{-3}$, $\sigma_t = 10^4$ and (b) *three sequence map* with $FPS = 20$, $\sigma_r = 10^{-9}$, $\sigma_t = 10^4$.

Chapter 5

Conclusions

In this dissertation, different visual localization methods for rover global positioning on extraterrestrial planets, comets asteroid, and small celestial bodies have been analyzed and improved. The investigated methods are based on a synergistic approach between rover and orbiter. The first part has been developed in collaboration with ALTEC for the localization operations that will be performed by Rover Operation Control Center (ROCC) for ExoMars 2020 rover. The second part has been carried out at NASA Jet Propulsion Laboratory and presents a collaborative SLAM algorithm, based on ORB-SLAM, for tumbling rover localization on asteroid comets and small bodies.

Map-based localization methods are fundamental to plan rover operations. For this purpose two methods have been investigated, in the first one the map is an orbiter image and in the second one the map is a Digital Elevation Model. Localization on orbiter image is performed by correlation of hills and craters between a NAVCAM or PANCAM panorama and an orbiter orthorectified image (like HiRISE image). This localization method is a refinement of rover position estimation performed by a human operator. Spirit landing site localization has been performed on an HiRISE image using a panoramic image.

Global localization with reference to a DEM model has been realized thanks to an exhaustive research, the measured skyline is compared to a series of skylines rendered from DEM on a grid of template positions. The localizations of Spirit landing site, and of Opportunity inside Victoria crater, have been performed. Algorithm performances have been tested with three different type of dataset: hills in the distance, rover outside a crater and rover inside a crater. We have shown that the tested global localization algorithm is able to localize the rover for the following dataset: far away hills and rover inside crater case. The algorithm is not able to localize the rover on the DTM when the panoramic image is taken from outside the crater and the skyline is almost flat. To perform this localization we have supposed to know pitch and roll angles measures. The HiRISE products which have

been used are already processed and Mars referenced, the PANCAM cylindrical mosaic has been assumed to be pre-processed. The following localization errors have been retrieved: Spirit landing site localization error is 51 m, this error seems to be limited by the quality of the DTM. Error in azimuth measurement is 1 deg. Analyzing Opportunity image taken on Victoria crater slopes, the localization error is 27m, and 5 deg in azimuth measurement.

Stereo Visual Odometry has been already used onboard MER rovers for relative localization during the traverse. In this dissertation, we have investigated the effects of 3D detected landmarks distance from stereo-camera center on motion reconstruction, An experimental comparison between seven different positions in a laboratory environment has been performed for both rotational and translational tests. Experimental results show that the landmarks average distance affects the rotation and translation reconstruction in two different ways: for translations, the uncertainty of camera position grows with the landmarks distance; on the other end this distance does not affect rotation uncertainty if the spatial distribution of landmarks is uniform. Despite, the covariance of the 3D landmarks has been used for weighting the 3D error in the cost function, the effect of high uncertainty far points is still visible in the motion computation and its uncertainty.

The accuracy of this visual odometry framework could be improved by using the tracked points in two different ways: points that have a low uncertainty in the ray direction could be useful to estimate the translation, instead points that have a large uncertainty in the ray direction could be very informative for the rotation estimation of the camera.

A synergistic localization and mapping approach between the mother-craft and the deployed daughters that exploit the visual SLAM approach has been investigated. Between the state of the art visual SLAM algorithm, ORB-SLAM2 has been chosen and modified for this application.

The capabilities to build a map and relocalize two different cameras with a modified version of ORB-SLAM2 has been shown. It is now possible to localize a large FOV camera on a map created with a narrow FOV camera, this condition is representative of the problem of localizing the tumbling/hopping rover on a map created from mother-spacecraft images. Moreover, we have pointed out how changes in illumination conditions limit the tracking capabilities on the previous map. With ORB-SLAM2 is possible to track the daughter spacecraft on the previous map with illumination changes up to 20 deg.

In order to test the algorithm performances, a series of realistic images of an asteroid mockup have been realized. A Vicon system has been used in order to give the trajectory ground truth. Tests show that the RMS trajectory estimation error is 0.44 cm and the RMS angular error is 0.2 deg, using as dataset a video with an asteroid mock up diameter of 138 cm. Vicon measurements have been used in order to build the maps to evaluate the

localization algorithm performances. The prior map accuracy is estimated to be 0.13 cm. The generated maps have been used to estimate the localization error of a large FOV camera, which represent the tumbling platform.

With the current configuration, it is possible to localize large FOV frames from a normalized distance corresponding to 0.1 of the target asteroid diameter up to a distance corresponding to the asteroid diameter, the normalized median error is between 0.014 and 0.022. If the asteroid diameter is 535 m, like asteroid Itokawa, the localization accuracy will be between 7.5 and 11.8 m. Moreover, also the off-pointing condition has been taken into account, the outlined framework is able to localize frames which have an off-pointing angle up to 57 deg.

References

[1] John P. Grotzinger, Joy Crisp, Ashwin R. Vasavada, Robert C. Anderson, Charles J. Baker, Robert Barry, David F. Blake, Pamela Conrad, Kenneth S. Edgett, Bobak Ferdowski, Ralf Gellert, John B. Gilbert, Matt Golombek, Javier Gómez-Elvira, Donald M. Hassler, Louise Jandura, Maxim Litvak, Paul Mahaffy, Justin Maki, Michael Meyer, Michael C. Malin, Igor Mitrofanov, John J. Simmonds, David Vaniman, Richard V. Welch, and Roger C. Wiens. Mars Science Laboratory Mission and Science Investigation. *Space Science Reviews*, 170(1):5–56, 2012.

[2] J. Vago, O. Witasse, H. Svedhem, P. Baglioni, A. Haldemann, G. Gianfiglio, T. Blancquaert, D. McCoy, and R. de Groot. ESA ExoMars program: The next step in exploring Mars. *Solar System Research*, 49(7):518–528, 2015.

[3] L. Pratt, D. Beaty, and A. Allwood. The mars astrobiology explorer-cacher (MAX-C): A potential rover mission for 2018. *Astrobiology*, 10(2):127–163, 2010.

[4] National Research Council. *Vision and Voyages for Planetary Science in the Decade 2013-2022*. The National Academies Press, Washington, DC, 2011.

[5] Joseph A. Starek, Behçet Açıkmeşe, Issa A. Nesnas, and Marco Pavone. *Spacecraft Autonomy Challenges for Next-Generation Space Missions*, pages 1–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[6] David W. Dunham, Robert W. Farquhar, James V. McAdams, Mark Holdridge, Robert Nelson, Karl Whittenburg, Peter Antreasian, Steven Chesley, Clifford Helfrich, William M. Owen, Bobby Williams, Joseph Veverka, and Ann Harch. Implementation of the First Asteroid Landing. *Icarus*, 159(2):433 – 438, 2002.

[7] Hajime Yano, T. Kubota, H. Miyamoto, T. Okada, D. Scheeres, Y. Takagi, K. Yoshida, M. Abe, S. Abe, O. Barnouin-Jha, A. Fujiwara, S. Hasegawa, T. Hashimoto, M. Ishiguro, M. Kato, J. Kawaguchi, T. Mukai, J. Saito, S. Sasaki, and M. Yoshikawa. Touchdown of the Hayabusa Spacecraft at the Muses Sea on Itokawa. *Science*, 312(5778):1350–1353, 2006.

[8] Jens Biele, Stephan Ulamec, Michael Maibaum, Reinhard Roll, Lars Witte, Eric Jurado, Pablo Muñoz, Walter Arnold, Hans-Ulrich Auster, Carlos Casas, Claudia Faber, Cinzia Fantinati, Felix Finke, Hans-Herbert Fischer, Koen Geurts, Carsten Güttler, Philip Heinisch, Alain Herique, Stubbe Hviid, Günter Kargl, Martin Knapmeyer, Jörg Knollenberg, Wlodek Kofman, Norbert Kömle, Ekkehard Kührt, Valentina Lommatsch, Stefano Mottola, Ramon Pardo de Santayana, Emile Remetean, Frank Scholten, Klaus J. Seidensticker, Holger Sierks, and Tilman Spohn. The landing(s) of

Philae and inferences about comet surface mechanical properties. *Science*, 349(6247), 2015.

[9] Mission complete Rosetta's journey ends in daring descent to comet. Accessed: 13/10/2016.

[10] R.Z. Sagdeev and A.V. Zakharov. Brief history of the Phobos mission. *Nature*, 341(6243):581–585, 1989.

[11] Yuichi Tsuda, Makoto Yoshikawa, Masanao Abe, Hiroyuki Minamino, and Satoru Nakazawa. System design of the Hayabusa 2—asteroid sample return mission to 1999 JU3. *Acta Astronautica*, 91:356 – 362, 2013.

[12] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007.

[13] The Rover Team. The pathfinder microrover. *Journal of Geophysical Research: Planets*, 102(E2):3989–4001, 1997.

[14] E. Gat, R. Desai, R. Ivlev, J. Loch, and D. P. Miller. Behavior control for robotic exploration of planetary surfaces. *IEEE Transactions on Robotics and Automation*, 10(4):490–503, Aug 1994.

[15] R. Volpe. Mars rover navigation results using sun sensor heading determination. In *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 1, pages 460–467 vol.1, 1999.

[16] Rongxing Li, Kaichang Di, Larry H Matthies, William M Folkner, Raymond E Arvidson, and Brent A Archinal. Rover localization and landing-site mapping technology for the 2003 mars exploration rover mission. *Photogrammetric Engineering & Remote Sensing*, 70(1):77–90, 2004.

[17] Rongxing Li, Steven W Squyres, Raymond E Arvidson, Brent A Archinal, Jim Bell, Yang Cheng, Larry Crumpler, David J Des Marais, Kaichang Di, Todd A Ely, et al. Initial results of rover localization and topographic mapping for the 2003 mars exploration rover mission. *Photogrammetric Engineering & Remote Sensing*, 71(10):1129–1142, 2005.

[18] Rongxing Li, Brent A. Archinal, Raymond E. Arvidson, Jim Bell, Philip Christensen, Larry Crumpler, David J. Des Marais, Kaichang Di, Tom Duxbury, Matt Golombek, John Grant, Ronald Greeley, Joe Guinn, Andrew Johnson, Randolph L. Kirk, Mark Maimone, Larry H. Matthies, Mike Malin, Tim Parker, Mike Sims, Shane Thompson, Steven W. Squyres, and Larry A. Soderblom. Spirit rover localization and topographic mapping at the landing site of gusev crater, mars. *Journal of Geophysical Research: Planets*, 111(E2):n/a–n/a, 2006. E02S06.

[19] Rongxing Li, Raymond E. Arvidson, Kaichang Di, Matt Golombek, Joe Guinn, Andrew Johnson, Mark Maimone, Larry H. Matthies, Mike Malin, Tim Parker, Steven W. Squyres, and Wesley A. Watters. Opportunity rover localization and topographic mapping at the landing site of Meridiani Planum, Mars. *Journal of Geophysical Research: Planets*, 112(E2):n/a–n/a, 2007. E02S90.

[20] T Parker, M Malin, M Golombek, T Duxbury, A Johnson, J Guinn, T McElrath, R Kirk, B Archinal, L Soderblom, et al. Localization, localization, localization. In *Lunar and Planetary Science Conference*, volume 35, 2004.

[21] Jeffrey J Biesiadecki, P Chris Leger, and Mark W Maimone. Tradeoffs between directed and autonomous driving on the Mars Exploration Rovers. *The International Journal of Robotics Research*, 26(1):91–104, 2007.

[22] Mark W Maimone, P Chris Leger, and Jeffrey J Biesiadecki. Overview of the Mars Exploration Rovers' autonomous mobility and vision capabilities. In *IEEE international conference on robotics and automation (ICRA) space robotics workshop*, 2007.

[23] J Maki, D Thiessen, A Pourangi, P Kobzeff, T Litwin, L Scherr, S Elliott, A Dingizian, and M Maimone. The Mars Science Laboratory engineering cameras. *Space science reviews*, 170(1-4):77–93, 2012.

[24] Raymond E. Arvidson, Karl D. Iagnemma, Mark Maimone, Abigail A. Fraeman, Feng Zhou, Matthew C. Heverly, Paolo Bellutta, David Rubin, Nathan T. Stein, John P. Grotzinger, and Ashwin R. Vasavada. Mars Science Laboratory Curiosity Rover megaripple crossings up to sol 710 in Gale Crater. *Journal of Field Robotics*, pages n/a–n/a, 2016.

[25] Mark Maimone. Curiouser and curiouser: Surface robotic technology driving Mars rover Curiosity's exploration of Gale crater. In *Robotics and Automation Workshop: on Planetary Rovers (ICRA Workshop), 2013 IEEE International Conference on. IEEE*, 2013.

[26] Andrew E Johnson, Steven B Goldberg, Yang Cheng, and Larry H Matthies. Robust and efficient stereo feature tracking for visual odometry. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 39–46. IEEE, 2008.

[27] Tetsuo Yoshimitsu, Takashi Kubota, Ichiro Nakatani, Tadashi Adachi, and Hiroaki Saito. Micro-hopping robot for asteroid exploration. *Acta Astronautica*, 52(2–6):441 – 446, 2003. Selected Proceedings of the 4th {IAA} International conference on L ow Cost Planetary Missions.

[28] Tetsuo Yoshimitsu, Takashi Kubota, and Ichiro Nakatani. MINERVA rover which became a small artificial solar satellite. 2006.

[29] R Pardo de Santayana, M Lauer, P Muñoz, and F Castellini. Surface Characterization and Optical Navigation at the Rosetta Flyby of Asteroid Lutetia. In *Proceedings of the 24th International Symposium on Space Flight Dynamics (ISSFD), Laurel, MD*, 2014.

[30] R Pardo de Santayana and M Lauer. Optical measurements for Rosetta navigation near the comet. In *Proceedings of the 25 th International Symposium on Space Flight Dynamics (ISSFD), Munich, Germany*, 2015.

[31] Mathias Lauer, Ulrich Herfort, Dave Hocken, and Sabine Kielbassa. Optical measurements for the flyby navigation of Rosetta at asteroid Steins. In *Proceedings 21st International Symposium on Space Flight Dynamics–21st ISSFD. Toulouse, France*, 2009.

[32] Robert G Reid, Loris Roveda, Issa AD Nesnas, and Marco Pavone. Contact dynamics of internally-actuated platforms for the exploration of small solar system bodies. In *Proceedings of the 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS2014), Saint-Hubert, Canada*, page 9, 2014.

[33] B Hockman, Andreas Frick, Issa AD Nesnas, and Marco Pavone. Design, control, and experimentation of internally-actuated rovers for the exploration of low-gravity planetary bodies. In *Field and Service Robotics*, pages 283–298. Springer, 2016.

[34] Marco Pavone, Julie Castillo, Jeffrey A. Hoffman, and Issa Nesnas. Spacecraft/Rover Hybrids for the Exploration of Small Solar System Bodies. Technical report, Final report for NASA NIAC Program, 2012.

[35] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[36] Laurent Kneip. *Real-Time Scalable Structure from Motion: From Fundamental Geometric Vision to Collaborative Mapping*. PhD thesis, University of Zurich, 2012.

[37] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building rome in a day. In *2009 IEEE 12th international conference on computer vision*, pages 72–79. IEEE, 2009.

[38] Stefano Debei, Alessio Aboudan, Giacomo Colombatti, and Marco Pertile. Lutetia surface reconstruction and uncertainty analysis. *Planetary and Space Science*, 71(1):64–72, 2012.

[39] Susie Green, Andrew Bevan, and Michael Shapland. A comparative assessment of structure from motion methods for archaeological research. *Journal of Archaeological Science*, 46:173 – 181, 2014.

[40] D. Scaramuzza and F. Fraundorfer. Visual Odometry [Tutorial]. *IEEE Robotics Automation Magazine*, 18(4):80–92, Dec 2011.

[41] F. Fraundorfer and D. Scaramuzza. Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications. *Robotics Automation Magazine, IEEE*, 19(2):78–90, June 2012.

[42] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[43] OpenCV library, http://opencv.org/.

[44] Frédéric Devernay and Olivier Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24, 2001.

[45] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000.

[46] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[47] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[48] Edward Rosten and Tom Drummond. *Machine Learning for High-Speed Corner Detection*, pages 430–443. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[49] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[50] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[51] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. *CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching*, pages 102–115. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[52] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. *BRIEF: Binary Robust Independent Elementary Features*, pages 778–792. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[53] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, Nov 2011.

[54] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, Oct 2015.

[55] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*, pages 2564–2571. IEEE, 2011.

[56] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980.

[57] L. Matthies and S.A. Shafer. Error modeling in stereo navigation. *Robotics and Automation, IEEE Journal of*, 3(3):239–248, June 1987.

[58] Clark F. Olson, Larry H. Matthies, Marcel Schoppers, and Mark W. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215 – 229, 2003.

[59] Yang Cheng, M.W. Maimone, and L. Matthies. Visual odometry on the Mars exploration rovers - a tool to ensure accurate driving and science imaging. *Robotics Automation Magazine, IEEE*, 13(2):54–62, June 2006.

[60] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652–I–659 Vol.1, June 2004.

[61] L. Kneip and H. Li. Efficient computation of relative pose for multi-camera systems. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 446–453, June 2014.

[62] Andrew I Comport, Ezio Malis, and Patrick Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 40–45. IEEE, 2007.

[63] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-D point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(5):698–700, Sept 1987.

[64] Berthold K. P. Horn, H.M. Hilden, and Shariar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOURNAL OF THE OPTICAL SOCIETY AMERICA*, 5(7):1127–1135, 1988.

[65] Yi Ma, Stefano Soatto, Jana Kosecka, and Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Verlag, 2004.

[66] D. Nister. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):756–770, June 2004.

[67] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976, June 2011.

[68] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155, 2008.

[69] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[70] Joint Committee for Guides in Metrology. JCGM 100: Evaluation of measurement data - guide to the expression of uncertainty in measurement. Technical report, JCGM, 2008.

[71] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613, May 2011.

[72] Manolis IA Lourakis and Antonis A Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)*, 36(1):2, 2009.

[73] Jose-Luis Blanco. A tutorial on SE(3) transformation parameterizations and on-manifold optimization. *University of Malaga, Tech. Rep*, 2010.

[74] Sameer Agarwal, Keir Mierle, and Others. Ceres Solver. http://ceres-solver.org.

[75] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics Automation Magazine*, 13(3):108–117, Sept 2006.

[76] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1309–1332, Dec 2016.

[77] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2016.

[78] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Aaai/iaai*, pages 593–598, 2002.

[79] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234, Nov 2007.

[80] L. Riazuelo, Javier Civera, and J.M.M. Montiel. C2TAM: A cloud framework for cooperative tracking and mapping. *Robotics and Autonomous Systems*, 62(4):401 – 413, 2014.

[81] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. REMODE: Probabilistic, monocular dense reconstruction in real time. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2609–2616. IEEE, 2014.

[82] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *2011 International Conference on Computer Vision*, pages 2320–2327, Nov 2011.

[83] Jakob Engel, Thomas Schöps, and Daniel Cremers. *LSD-SLAM: Large-Scale Direct Monocular SLAM*, pages 834–849. Springer International Publishing, Cham, 2014.

[84] Alejo Concha and Javier Civera. DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 5686–5693. IEEE, 2015.

[85] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.

[86] D. Galvez-López and J. D. Tardos. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, Oct 2012.

[87] R. Prakash, P. D. Burkhart, A. Chen, K. A. Comeaux, C. S. Guernsey, D. M. Kipp, L. V. Lorenzoni, G. F. Mendeck, R. W. Powell, T. P. Rivellini, A. M. S. Martin, S. W. Sell, A. D. Steltzner, and D. W. Way. Mars Science Laboratory entry, descent, and landing system overview. In *2008 IEEE Aerospace Conference*, pages 1–18, March 2008.

[88] Rongxing Li, Shaojun He, Yunhang Chen, Min Tang, Pingbo Tang, Kaichang Di, Larry Matthies, Raymond E. Arvidson, Steven W. Squyres, Larry S. Crumpler, Tim Parker, and Michael Sims. MER Spirit rover localization: Comparison of ground image– and orbital image–based methods and science applications. *Journal of Geophysical Research: Planets*, 116(E7):n/a–n/a, 2011. E00F16.

[89] David E Smith, Maria T Zuber, Sean C Solomon, Roger J Phillips, James W Head, James B Garvin, W Bruce Banerdt, Duane O Muhleman, Gordon H Pettengill, Gregory A Neumann, et al. The global topography of mars and implications for surface evolution. *Science*, 284(5419):1495–1503, 1999.

[90] Douglass A. Alexander, Robert G. Deen, Paul M. Andres, Payam Zamani, Helen B. Mortensen, Amy C. Chen, Michael K. Cayanan, Jeffrey R. Hall, Vadim S. Klochko, Oleg Pariser, Carol L. Stanley, Charles K. Thompson, and Gary M. Yagi. Processing of Mars Exploration Rover imagery for science and operations planning. *Journal of Geophysical Research: Planets*, 111(E2):n/a–n/a, 2006. E02S02.

[91] Rongxing Li, Kaichang Di, Jue Wang, Xutong Niu, Sanchit Agarwal, Evgenia Brodyagina, Erik Oberg, and Ju Won Hwangbo. A WebGIS for spatial data processing, analysis, and distribution for the MER 2003 mission. *Photogrammetric Engineering & Remote Sensing*, 73(6):671–680, 2007.

[92] Alfred S McEwen, Eric M Eliason, James W Bergstrom, Nathan T Bridges, Candice J Hansen, W Alan Delamere, John A Grant, Virginia C Gulick, Kenneth E Herkenhoff, Laszlo Keszthelyi, et al. Mars Reconnaissance Orbiter's High Resolution Imaging Science Experiment (HiRISE). *Journal of Geophysical Research: Planets*, 112(E5), 2007.

[93] Fengliang Xu. *Mapping and localization for extraterrestrial robotic explorations*. PhD thesis, Citeseer, 2004.

[94] F. Stein and G. Medioni. Map-based localization using the panoramic horizon. *IEEE Transactions on Robotics and Automation*, 11(6):892–896, Dec 1995.

[95] Patrick J.F. Carle, Paul T. Furgale, and Timothy D. Barfoot. Long-range rover localization by matching LIDAR scans to orbital elevation maps. *Journal of Field Robotics*, 27(3):344–370, 2010.

[96] R. L. Kirk, E. Howington-Kraus, M. R. Rosiek, J. A. Anderson, B. A. Archinal, K. J. Becker, D. A. Cook, D. M. Galuszka, P. E. Geissler, T. M. Hare, I. M. Holmberg, L. P. Keszthelyi, B. L. Redding, W. A. Delamere, D. Gallagher, J. D. Chapel, E. M. Eliason, R. King, and A. S. McEwen. Ultrahigh resolution topographic mapping of Mars with MRO HiRISE stereo images: Meter-scale slopes of candidate Phoenix landing sites. *Journal of Geophysical Research: Planets (1991–2012)*, 113(E3), 3 2008.

[97] R. L. Kirk, E. Howington-Kraus, M. R. Rosiek, J. A. Anderson, B. A. Archinal, K. J. Becker, D. A. Cook, D. M. Galuszka, P. E. Geissler, T. M. Hare, I. M. Holmberg, L. P. Keszthelyi, B. L. Redding, W. A. Delamere, D. Gallagher, J. D. Chapel, E. M. Eliason, R. King, and A. S. McEwen. Ultrahigh resolution topographic mapping of mars with MRO HiRISE stereo images: Meter-scale slopes of candidate Phoenix landing sites. *Journal of Geophysical Research: Planets*, 113(E3):n/a–n/a, 2008.

[98] M Pertile, S Debei, and E Lorenzini. Uncertainty analysis of a stereo system performing ego-motion measurements in a simulated planetary environment. *Journal of Physics: Conference Series*, 459(1):012056, 2013.

[99] Paul Furgale, Pat Carle, John Enright, and Timothy D Barfoot. The Devon Island rover navigation dataset. *The International Journal of Robotics Research*, page 0278364911433135, 2012.

[100] S. Chiodini, M. Pertile, and S. Debei. Visual odometry system performance for different landmark average distances. In *2016 IEEE Metrology for Aerospace (MetroAeroSpace)*, pages 382–387, June 2016.

[101] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[102] M. Pertile, S. Chiodini, and S. Debei. Comparison of visual odometry systems suitable for planetary exploration. In *Metrology for Aerospace (MetroAeroSpace), 2014 IEEE*, pages 232–237, May 2014.

[103] Marco Pertile, Sebastiano Chiodini, Stefano Debei, and Enrico Lorenzini. Uncertainty comparison of three visual odometry systems in different operative conditions. *Measurement*, 78:388 – 396, 2016.

[104] V. Peretroukhin, J. Kelly, and T.D. Barfoot. Optimizing camera perspective for stereo visual odometry. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 1–7, May 2014.

[105] G. Dubbelman and F.C.A. Groen. Bias reduction for stereo based motion estimation with applications to large scale visual odometry. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2222–2229, June 2009.

[106] Joint Committee for Guides in Metrology. JCGM 101: Evaluation of measurement data - supplement 1 to the "guide to the expression of uncertainty in measurement" - propagation of distributions using a monte carlo method. Technical report, JCGM, 2008.

[107] J. Delaune, G. Le Besnerais, T. Voirin, J.L. Farges, and C. Bourdarias. Visual–inertial navigation for pinpoint planetary landing using scale-based landmark matching. *Robotics and Autonomous Systems*, 78:63 – 82, 2016.

[108] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Scale drift-aware large scale monocular SLAM. *Robotics: Science and Systems VI*, 2010.

[109] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.