

ADVANCEMENTS OF VISION BASED AUTONOMOUS PLANETARY LANDING SYSTEMS AND PRELIMINARY TESTING BY MEANS OF A NEW DEDICATED EXPERIMENTAL FACILITY

Marco Ciarambino, Paolo Lunghi, Luca Losi, and Michèle Lavagna

Politecnico di Milano, Aerospace Science and Technology Department, via La Masa, 34, 20156 Milano, Italy

ABSTRACT

This paper presents the design and testing activities of a new experimental facility currently under development at Politecnico di Milano, at the premises of the Aerospace Science and Technology Department (DAER). The purpose of the facility is the testing of novel Guidance, Navigation, and Control algorithms for autonomous landing on planets and small bodies, especially for vision-based relative navigation and hazard detection and avoidance. The design, integration and testing activities of the facility are presented, as well as the suite of GNC algorithms, currently under development at PoliMi-DAER, that are going to be tested. The ultimate goal is to create and bring up to TRL 5 a whole Autonomous Guidance, Navigation and Control chain for autonomous landings.

Key words: Planetary Landing; Hazard Detection & Avoidance; Vision-based navigation; Trajectory optimization; Experimental facility .

1. INTRODUCTION

In the next years, autonomous navigation and landing capabilities are going to play a crucial role for the next space system generation. A renewed interest in space exploration had brought to the development of numerous missions in which spacecraft autonomy and complex landing maneuvers are involved. ESA is working together with ROSCOSMOS on a cooperative programme for Moon, Mars, and Phobos exploration: the Luna-Resurs Lander mission (Luna 27) is planned for 2020 and is focused on the exploration of the lunar south pole. Part of the European contribution is PILOT (Precise and Intelligent Landing using Onboard Technologies), a subsystem devoted to the validation of pinpoint landing and Hazard Detection and Avoidance (HDA) technologies [1]. The two agencies collaborate also in the ExoMars programme, devoted to the exploration of Mars: in October 2016, despite the failure encountered in the landing phase, the lander Schiaparelli was able to collect precious data, that will be exploited to refine the design of the next 2018 mission, which includes the release of a rover on the Martian ground [2]. Also a Phobos sample return mission, called Phootprint, has been proposed to

be launched in 2024. The autonomous landing problem has been studied by NASA since 2006 in the frame of the Autonomous Landing Hazard Avoidance Technology (ALHAT) programme [3]. Part of the developed technologies were tested on the Morpheus lander demonstrator, and they will be integrated in future exploration missions. Also CNSA has recently performed its first lunar landing with a lander/rover system in the Chang'e 3 mission, while ISRO is planning to put a lander carrying a rover on the lunar surface by the early 2018 in the Chandrayaan 2 mission.

Major improvements in Guidance, Navigation and Control are still required to obtain a fully autonomous system with HDA capabilities: vision-based systems are one of the most promising technologies to achieve the required level of precision and accuracy. Extensive test and validation activities are required to ensure the necessary level of robustness: affordable and repeatable datasets from real mission are scarcely available, often lacking the additional data necessary for the computation of a ground-truth solution to compare with the obtained results. Artificially generated images can be a good substitute, but the required level of realism implies the use of computationally intensive, high fidelity rendering algorithms that make difficult closed-loop testing (and often impossible real-time hardware-in-the-loop). The use of analog facilities, capable to simulate a landing in a scaled environment, can supply repeatable and controllable data. This paper presents the design and testing activities of a novel CNG chain for autonomous landing, together with the setup of a new experimental facility currently under development at Politecnico di Milano, at the premises of the Aerospace Science and Technology Department (DAER).

2. HAZARD DETECTION SYSTEM

The hazard detection system under development at PoliMi-DAER is based on Artificial Neural Networks (ANN). If properly trained, ANNs are capable to autonomously find underlying rules that correlate one input space to the correspondent output, without any previous knowledge of the actual link between the twos. The capability to operate also in conditions not explicitly considered during the projects phase makes this kind of system very attractive for HDA tasks. Information extracted from a single-channel image provided by a monocular

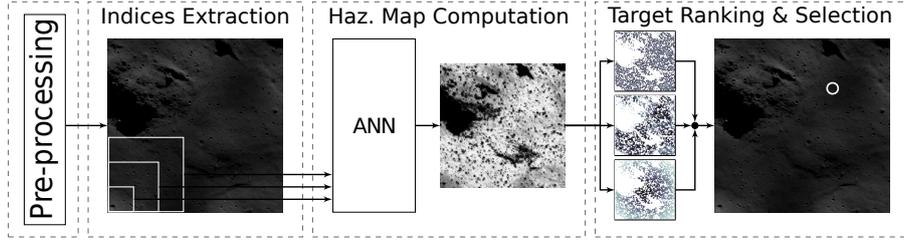


Figure 1: Hazard detection and target selection system architecture.

navigation camera are processed by a ANN to generate a hazard map of the landing area, processed by the site selection routine to compute the new target. Figure 1 shows a logical scheme of the algorithm. Following, the hazard detection method is briefly summarized. A detailed dissertation is available in [4].

2.1. System Architecture

The retargeting process is divided in 4 different sub-phases:

Input and preprocessing. A single channel 8-bit 1024×1024 px image is assumed as input. During the hazard detection maneuver the spacecraft is assumed in a near vertical attitude, small deviations from nadir pointing are corrected by applying a perspective transformation.

Image processing and input assembly. The input image is analyzed and different indexes are extracted. This process computes low level information from the image in order to reduce the data space in which the network should detect the morphological features of the terrain. The same image processing techniques are computed at different scales to allow the ANN to grasp relative distances and depths [5]. Considered indexes extracted from the image are local mean μ and standard deviation σ , together with higher order information such as image gradient $Grad$ (1st order) and Laplacian of Gaussian LoG (2nd order). LoG combines a Gaussian filter with the Laplacian operator, and it is often used as an edge detector in images. It is defined analytically as:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}, \quad (1)$$

where x, y are the image coordinates and σ the standard deviation. Indexes can be computed efficiently with linear filters, to reduce the computational burden. Also the Sun elevation angle above the horizon is considered, allowing the network to understand features at different illumination conditions.

A total of 13 indexes are fed into the network to compute each hazard map pixel: $\mu, \sigma, grad, LoG$ for three different image scales, plus the value of Sun elevation angle.

Hazard map computation and Target selection. extracted indexes are processed by a cascade neural network [6], that computes a hazard map. Danger is estimated by a

hazard index whose value spans from 0 (completely safe), to 1 (completely unsafe). The map is exploited to discard unsafe targets and to select the best landing site. All the sites that do not respect minimum requirements on safety or dimension (taking into account the lander footprint, margined with the expected dispersion at touchdown) are immediately discarded. The remaining candidates are ranked according to 3 criteria: minimum hazard index, maximum landing area, minimum distance from the nominal landing site (to maximize the probability to compute a feasible landing trajectory). The weight of these parameters can be tuned to maximize the performance.

2.2. Ground truth dataset

High resolution lunar DEMs (5-2 m/point) from LROC¹ have been used as the base to craft an artificial images dataset. Resolution has been increased up to 0.3 m/point with the addition of random small craters, boulders and fractal noise following real statistical distributions [7, 8, 4]. Images are then rendered with ray tracing techniques² with realistic illumination conditions assuming a pinhole camera with 60° angle of view. Ground truth is computed directly from DEM data: the local mean plane of the DEM is adopted to estimate the local slope angle, while the maximum and minimum deviations from the plane express the local roughness. Slopes and roughness maps are projected in camera coordinates, to match the navigation images. Then, the ground truth hazard map is assembled assigning a hazard index 1 for pixels in shadow, 0.33 to those pixels which fail the roughness or the slope criteria, 0.66 to those which fail both. The remaining pixels are classified as safe and set to 0. Finally, the ground truth solution is downsampled through Gaussian pyramids to the final hazard map resolution.

2.3. Performances

The system was tested on a set of 8 images (not used in the ANN training) of four lunar regions rendered at two different Sun inclination, 15° and 80° . A lander of 3 meters of diameter in footprint and a navigation error of 15 meters at 3σ has been considered. Terrain roughness 0.5 m and slopes over 15° are considered unsafe.

¹Courtesy of NASA and Arizona State University. URL: http://wms.lroc.asu.edu/lroc/rdr_product_select, last visit on: June 1, 2017.

²Persistence of Vision Raytracer (Version 3.7). Retrieved from <http://www.povray.org/download/>

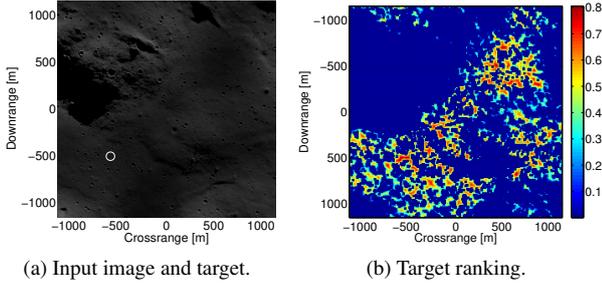


Figure 2: Target landing site ranking and selection.

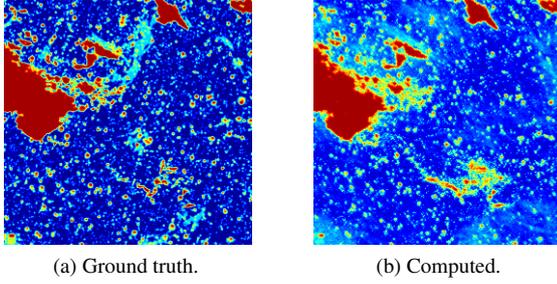


Figure 3: Comparison between the computed hazard map and the ground truth solution.

A comparison between the ground truth and the computed hazard map are shown in Fig. 3: terrain features are correctly interpreted by the network. While the ranking of the candidate landing sites is shown in In Fig. 2. A landing site can be classified as *True Positive* (TP), meaning a correctly identified target, *False Positive* (FP), an unsafe site classified erroneously as safe, and the corresponding *False Negative* (FN), and *True Negative* (TN). Defining the *Safety Ratio* r_S as the fraction of true positives with respect to the total number of landing sites found:

$$r_S = \frac{TP}{TP + FP}, \quad (2)$$

and defining the *Correctness Ratio* r_C as the fraction of correctly identified sites (TP) with respect to the true safe landing site in the image:

$$r_C = \frac{TP}{TP + FN} \quad (3)$$

the probability to select an unsafe site is minimized maximizing r_S , whereas as r_C increases, the available landing area increases. The whole performance can be assessed with a unique index J expressed as:

$$J = r_S^5 R_C^{1/5} \quad (4)$$

where the two exponents privilege landing sites safety during J maximization. A True Positive as Target Landing site: in the worst case, the first False Positive position in the ranking is position 39, (average 695). The obtained mean Safety Ratio is 0.9649, meaning that over 96% of the landing sites found are actually safe. The routine was coded in C++ and run on a AMD A10-7700K APU,

with a 64 bit Ubuntu 14.04 GNU/Linux operative system. Computation time is well below 440 ms, with indexes extraction stage almost 50% or the total runtime. Recent improvements in space qualified hardware for computationally intensive tasks, will make possible to massively execute the heaviest tasks in parallel, with expected dramatic improvements [9].

3. ADAPTIVE GUIDANCE

Once a safe landing site is selected, the system must compute a new feasible trajectory toward the new target. A fuel-optimum criterion is followed, in order to maximize the attainable landing area in subsequent target updates, that may be required as soon as smaller terrain features become observable. Here the structure of the algorithm is summarized, and some numerical results are presented. For a detailed description, see [10, 11].

The expected time of flight of the landing phase in which HDA tasks take place is in the order of 1 min, and the mass is supposed to significantly change during the maneuver. Distances, for both downrange and altitude, are small compared to the planet's radius; thus, the assumption of a constant gravity field with flat ground is appropriate. Aerodynamic forces are neglected: the effects of the possible presence of atmosphere (especially for low densities, as in the case of Mars) could be omitted due to the relative low velocity (on the order of 100 m s^{-1}), and the associated forces can be treated as disturbances. The translational dynamics of the spacecraft are expressed in a ground reference system as:

$$\dot{\mathbf{r}} = \mathbf{v} \quad \dot{\mathbf{v}} = \frac{\mathbf{T}}{m} + \mathbf{g} \quad \dot{m} = -\frac{T}{I_{sp}g_0} \quad (5)$$

where $\mathbf{r} = [x, y, z]^T$, x is the altitude, y is the downrange direction and z is the cross-range; \mathbf{g} is the constant acceleration of gravity vector of the planet, I_{sp} the specific impulse of the main engine, and g_0 the standard gravity acceleration on Earth. The thrust net magnitude is indicated with $T = \|\mathbf{T}\|$.

The thrust vector acts as the control variable. The mass equation is linked to the control acceleration by the thrust-to-mass ratio \mathbf{P} :

$$\mathbf{P} = \mathbf{T}/m = \dot{\mathbf{v}} - \mathbf{g} \quad (6)$$

Then, the mass equation in system (5) can be rewritten as a first order linear ordinary differential equation:

$$\dot{m} = -\frac{P}{I_{sp}g_0}m \quad (7)$$

where $P = \|\mathbf{P}\|$.

The states \mathbf{r}_0 , \mathbf{v}_0 and m_0 at the initial time t_0 are supposed to be known. At the end of the maneuver, at time t_f , the final states \mathbf{r}_f and \mathbf{v}_f are constrained. Then, the optimal guidance problem is to find a control profile $\mathbf{T}(t)$ to bring the system from the initial to the target final states,

compatibly with all the constraints imposed by the actual system architecture. For sake of simplicity is considered $t_0 = 0$. The main thruster is assumed to be tightly connected to the spacecraft body. Then, the thrust vector depends only on the attitude of the spacecraft, expressed by the vector of Euler's angle \mathbf{e} and on the thrust magnitude T . Then, the initial acceleration is function only of the initial thrust magnitude. Moreover, at the end of the maneuver, the lander is required to be aligned with the local vertical on the Target Landing Site. In case of flat surface, this condition reduces to impose null horizontal acceleration. A total of 17 boundary constraints are then available for position, velocity and acceleration components: 6 on initial states, 3 on initial acceleration (function of initial thrust magnitude), 6 on target final states and 2 on the final acceleration due to the final attitude requirements

$$\begin{aligned} \mathbf{r}(0) &= \mathbf{r}_0 & \mathbf{r}(t_f) &= \mathbf{r}_f \\ \mathbf{v}(0) &= \mathbf{v}_0 & \mathbf{v}(t_f) &= \mathbf{v}_f \\ \dot{\mathbf{v}}(0) &= \mathbf{f}(T_0) & \dot{\mathbf{v}}(t_f) &= [\text{free}, 0, 0]^T \end{aligned} \quad (8)$$

The acceleration is expressed in a polynomial form of minimum order needed to satisfy the boundary constraints:

$$\dot{\mathbf{v}}(t) = \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} \dot{v}_{0x} + c_{1x}t + c_{2x}t^2 \\ \dot{v}_{0y} + c_{1y}t + c_{2y}t^2 + c_{3y}t^3 \\ \dot{v}_{0z} + c_{1z}t + c_{2z}t^2 + c_{3z}t^3 \end{bmatrix} \quad (9)$$

By integrating the acceleration twice and applying the boundary conditions, the trajectory is parametrized in terms of time-of-flight t_f and initial thrust magnitude T_0 , that are considered as optimization parameters. The thrust-to-mass ratio can be obtained from Eq. (6) and the thrust profile is:

$$\mathbf{T} = m\mathbf{P} \quad (10)$$

where the mass profile is obtained by solving Eq. (7). From the thrust vector a complete guidance profile, in terms of Euler angles and thrust magnitude, is easily obtained.

In addition to boundaries constraints, the system is subject also to path and box constraints. The initial thrust magnitude is bounded to the thrust actually available on-board, while the time-of-flight must lie between its lower and upper limit:

$$0 < T_{\min} \leq T_0 \leq T_{\max} \quad (11)$$

$$0 < t_{\min} \leq t_f \leq t_{\max} \quad (12)$$

The theoretical t_{\max} is determined by the amount of fuel on board m_{fuel} , whereas the adopted t_{\min} corresponds to the time required by the lander to reach the ground with maximum thrust pointing downward. The angular velocity of the spacecraft is limited by the actual control torques $M_{C_{\max}}$ given by the Attitude Control System (ACS). Torques are approximated by the decoupled term due to the angular acceleration, which is a sufficiently accurate approximation in case of low angular speed.

Exploiting this approximation leads to the following inequality:

$$I_{\max} \|\dot{\boldsymbol{\omega}}(t)\| \leq M_{C_{\max}} \quad (13)$$

in which $\dot{\boldsymbol{\omega}}$ is the derivative of the rotational velocity vector, and I_{\max} is the maximum moment of inertia at initial time t_0 . The spacecraft is required to remain in a cone pointed at the target and defined by the maximum slope angle δ_{\max} . This constraint has the dual purpose to assure that the lander does not penetrate the ground, even in presence of bulky terrain features near the landing site, and to limit the angle of view on the target, as could be required by HDA systems [12]. The constraint is expressed by the inequality:

$$r_y^2(t) + r_z^2(t) \leq r_x^2(t) \tan^2(\delta_{\max}) \quad (14)$$

Finally, the final mass must be included between the initial value and the spacecraft dry mass. Since the mass trend is strictly monotone by problem construction, the only constraint with respect the minimum mass is verified:

$$m(t_f) \leq m_{\text{dry}} \quad (15)$$

3.1. Optimization Problem and Numerical Results

The optimization problem takes the form:

Find T_0 and t_f , in the domain defined by the inequalities (11), that minimize the fuel consumption computed by the Eq. (7), subject to constraints (13), (14), and (15).

The optimization could be solved with any non-linear programming (NLP) solver: the choice of this solver has a huge impact over the final convergence properties and computational time. A dedicated optimization algorithm based on Taylor Differential Algebra (DA) was developed. Instead to be modeled as simple real numbers, quantities are represented as their Taylor expansion around a nominal point. In this way, DA variables carry more information rather than their mere punctual values. The computation of the objective function as a DA variable includes its sensitivity to the variation of the optimization variables, leading to find the optimal solution in a reduced number of iterations, using only simple algebraic computation between Taylor coefficients. A detailed discussion about the developed optimization algorithm is included in [11].

A series of different Monte Carlo (MC) simulations regarding a realistic case of lunar landing were carried out to estimate the system's performances. The parameters considered in the initial dispersion include initial position, velocity, attitude, amount of fuel on board, specific impulse, spacecraft moment of inertia, available thrust and gravity acceleration. The case here reported as example represents a large scale diversion maneuver from a nominal altitude of 2000 m, in which a random large horizontal diversion (600 m 1σ) is ordered. Obtained trajectories are shown in Fig. 4a: in all the cases, the ordered diversion was found feasible by the guidance algorithm. A MC simulation is exploited also to assess the algorithm performances in terms of attainable landing area and fuel consumption. A series of 1×10^5 random diversions between ± 4000 m along both the horizontal axes is ordered

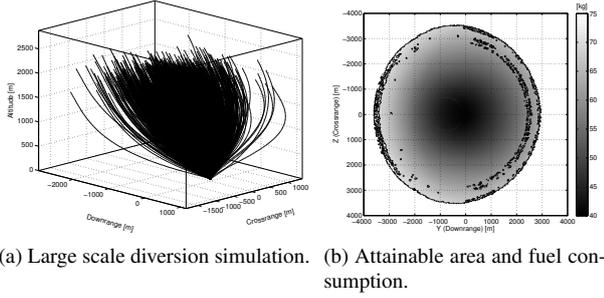


Figure 4: Adaptive Guidance Monte Carlo Simulation (from Ref. [11]).

from the same nominal conditions of Fig. 4a. The attainable landing area can be obtained by correlating optimization results together with the coordinates of the target, as shown by Fig. 4b, in which only the feasible points (satisfying all the constraints) are shown. The attainable area is approximately circular, with a radius larger than 2300 m centered at the nominal landing site, a performance better than what is required for similar scenarios [13]. The same simulation was exploited also to obtain an estimation of the computation time. Running on a Intel® Core™ i7-2630QM CPU at 2 GHz of frequency, the mean computation time is 25.23 ms with a standard deviation (STD) of 7.16 ms, a performance compatible with on-board computation.

4. VISION-BASED NAVIGATION

The Navigation Algorithm in development at PoliMi-DAER relies on a single channel camera working in the visible spectrum and is based on Visual Odometry and Simultaneous Localization and Mapping (SLAM) [14, 15]. Salient features are extracted and tracked from the incoming images, while a local sparse 3D map of the landing area is built and used for navigation: relative position and attitude of the spacecraft are optimized with Bundle Adjustment (BA). Here a summary of the structure of the system is presented, along with results obtained from a test campaign made on synthetic images from the Lunar dataset. The proposed architecture for the Navigation System is shown in Fig. 5.

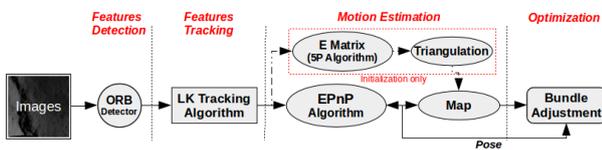


Figure 5: Navigation System architecture.

Features detection. The Oriented FAST and Rotated BRIEF (ORB) [16] is adopted as features detector: features are extracted from the first frame to initialize the navigation. The computational burden of ORB is very light, with good invariance to viewpoint and scale, and resiliency to different light conditions. To improve the

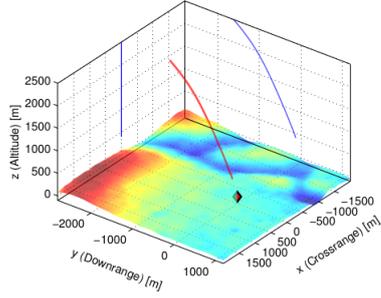
features distribution over the frame, the image is segmented into 64 sectors, in which features are extracted independently. An upper bound of 300 features is set to be extracted to maintain a low computational cost.

Features tracking. ORB features extracted from the first frame are tracked on subsequent images with the pyramidal Lucas-Kanade algorithm [17]. This tool works projecting known features on subsequent frames searching for correspondence with the new ones in a bounded region. The pyramidal approach makes the algorithm more robust to track large motions, searching for features starting from the highest level of an image pyramid. A stringent culling keypoint procedure is applied to reject wrongly tracked features or ones tracked with low accuracy, with the objective to constantly retain a modest number of features but tracked with high precision. In case of tracking failure or each time the number of features drops below a threshold, a new detection is triggered and tracking restarted. Still tracked features are merged with the new ones to constantly keep tracking the highest number possible.

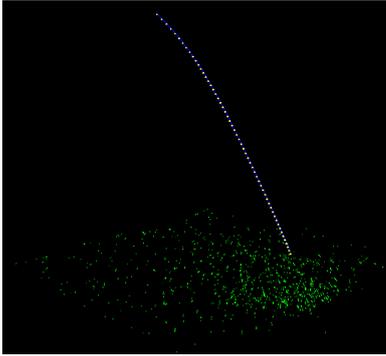
Motion estimation. Motion estimation relies on correspondences between 3D features of a map and 2D tracked features. The map is built in parallel by the algorithm itself, and it is initialized by the first 2 frames. 2D to 2D correspondences are exploited to implement the 5-Point algorithm [18] and retrieve the essential matrix E of the first image pair. The motion is obtained up to a scale factor applying Singular Value Decomposition (SVD) and cheirality check. Outliers are rejected by RANSAC iterations. From the obtained motion 2D features are triangulated implementing the optimal method proposed in [19]. Reprojection error check is made at every step and a stringent culling policy is applied to discard wrongly triangulated points along with correspondent features. Full BA is finally adopted to refine the map. Whenever the tracking fails or is re-triggered, a new map is triangulated and merged with the existent. Once the 3D sparse map is computed and features are tracked, the camera pose is retrieved by solving the "Perspective-n-Point problem" (PnP). EPnP algorithm [20], which gives a non-iterative solution both for planar and non planar scenes. The trajectory is reconstructed except for an absolute scale factor. The scene scale can be computed through data fusion, filtering measurements coming from other sensors (IMU, Laser Altimeter): a proper navigation filter is currently under development.

Optimization. Features extracted and tracked in images are affected by noise: the epipolar constraint is never exactly satisfied, and uncertainties affect the triangulated points, causing an increasing drift of the trajectory. Bundle Adjustment is implemented on the navigation algorithm to counteract these effects. Levenberg Marquadt algorithm is exploited to solve the optimization problem, which is implemented in two ways: **Full BA.** At the end of map initialization a full BA run is made. Optimizing both triangulated points location and estimated pose between the two frames used for initialization.

Motion only BA. At the end of each motion estimation step, retrieved camera pose is optimized keeping fixed the observed map points.



(a) Reference true Approach trajectory.



(b) Reconstructed map and trajectory of the Approach dataset with downward pointing camera.

Figure 6: Ground truth and reconstructed approach trajectory of the Lunar dataset.

4.1. Tests on Lunar Dataset

The system was tested on sequences of synthetic images simulating Lunar landing trajectories. The method described in Section 2 has been adopted to generate the images. Two landing trajectories were considered, *Main Brake* and *Approach*. *Main Brake* is a constant thrust horizontal brake maneuver over the Moon Planck crater, while *Approach* is a variable thrust approach phase maneuver representing the last part of the landing. Hereafter results are presented for the *Approach* trajectory with fixed downward pointing camera running at 1 Hz. Fig. 6a shows the reference trajectory, while Fig. 6b shows the reconstructed one along with the sparse 3D map built. In these results, the absolute scale factor is recovered in postprocessing.

Fig. 7 shows the trend of the reconstructed pose compared to the ground truth. Error is shown in terms of norm of the difference, with RMSE below 2 m on each of the reference axis along the whole trajectory. Results obtained at this stage are quite satisfactory and show the whole potential of the Vision-Based approach, errors of the order of 2 meters are obtained on a really large scale trajectory with a difficult motion and a non-optimal

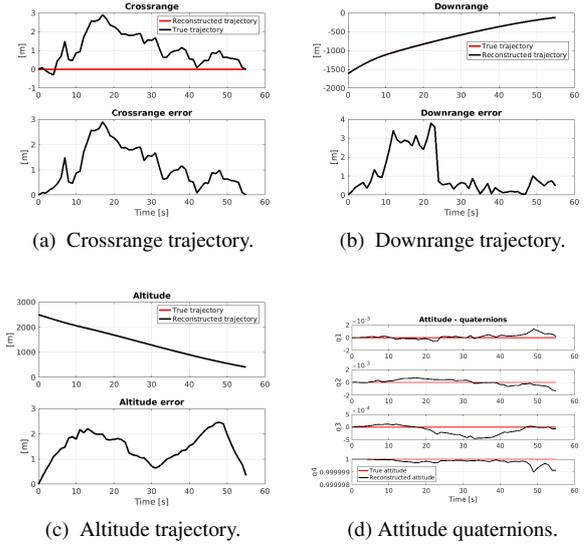


Figure 7: Reconstructed trajectory in time compared against ground truth and error module.

features distribution. Future development will focus on the improvement of the current BA implementation along with data fusion and scale factor direct estimation.

5. EXPERIMENTAL FACILITY

To further increase the TRL of the aforementioned algorithms, an hardware-in-the-loop experimental facility is under setup at PoliMi-DAER premises. Since the scarce availability of complete real landing imagery datasets, vision-based algorithms development relies widely on synthetic images. To validate such approach, experiments are necessary. Moreover, the whole navigation system performance can be assessed only connecting the composing parts together, to verify mutual influences. The facility is composed by a robotic arm carrying a suite of sensors to simulate lander dynamics, a 3D planetary mock-up, an illumination system and control and test computers. The setup is shown in Fig. 8a. The goal is to reproduce the landing maneuver over a scaled but realistic environment. The system is designed to verify either hardware and software breadboards up to TRL 4, with the possibility to update the system in the future to carry out also real-time hardware-in-the-loop simulations to qualify GNC technologies up to TRL 5.

Planetary mock-up. The selection of a proper scale factor is needed to properly determine the facility manufacturing requirements. It is assumed that the hazard detection starts at a maximum altitude of 2000 m. For reasons of cost containment, the choice of the robotic arm was restricted to the use of an already available hardware, with an operative envelope of 1 m. Then, a maximum scale factor of 2000:1 has been considered. The target accuracy for a navigation algorithm at touchdown is in the order of 10 m, which corresponds to 5 mm in the scaled environment. To have a resolution of the terrain at least one order



(a) The facility running the test.



(b) Landing image taken by the landing camera.

Figure 8: Experimental facility running a preliminary functional test.

of magnitude greater than the landing accuracy, a resolution of at least 0.5 mm is required. The scale factor can be adapted to simulate closer range maneuvers with higher details, due to the fractal structure of the Moon surface. Urethane foam has been chosen as material for the lunar diorama due to its surface finish that yields the correct optical properties and because of its great workability. The overall dimension of the diorama (2400×2000 mm) was selected to exploit the full envelope of the robotic arm and a maximum field of view of 60° for the landing camera. The model is divided in 8 tiles, each measuring 1200×500 mm. The Digital Elevation Model (DTM) has been selected from the GLD-100 NASA LROC dataset³, in order to have mixed terrain features, from plains to rough slopes. The model was enriched with low scale detail, non visible due to the limited DEM resolution, by the addition of small craters, boulders and fractal noise [4].

Robotic Arm. A Mitsubishi PA10-7C robotic arm is available at DAER. It features 7 DoF and it is capable to handle a 10 kg payload in a operative maximum spatial range of 1.03 m. At its end effector, the robotic arm carries the sensor suite, that is made up by a navigation camera and a range sensor simulating a laser altimeter.

Illumination system. It must be able to guarantee the realistic light environment of the planetary surface during the simulation. In particular, for planets without atmosphere like the Moon, diffuse light must be avoided. It is composed by an array of LED spotlights with narrow beam angle, and a non reflective black structure to prevent external sunlight and internal reflections to jeopardize the simulation accuracy.

Diorama Calibration. The actual milled diorama is generally different from the numerical model, due to imperfections during the production process. The camera trajectory relative to the terrain must be reconstructed with high accuracy, at least one order of magnitude better than

the navigation algorithms expected to be tested on the facility. Being the desired accuracy in the development of navigation algorithms up to 10 m (3σ , which corresponds to 5 mm at the target scale factor 1:2000), the target accuracy in diorama calibration is better than 0.5 mm. Dense matching method was selected to perform the shape reconstruction: several photos from different angles are taken for each tile; then, structure from motion algorithm are used to obtain the camera pose for each image. Finally, dense cloud point models are obtained by triangulation of optical features between different frames. The tiles models are eventually assembled by means of Iterative Closest Point (ICP) algorithms, generating the final model for the whole diorama. The final accuracy is estimated by the features reprojection error, resulted well below the 0.5 mm threshold. An example of reconstructed tile model is presented in Fig. 9.

5.1. Test Campaign

Currently, facility functional test are ongoing, and GNC testing activities are going to start during the summer 2017. GNC algorithms will be tested through a "bottom-up" procedure: first, single subsystems are checked for their standalone performance; then, they are progressively joined together to check if relative couplings degrade the performances; eventually, software-in-the-loop simulations tests to validate the whole system in flight-like conditions.

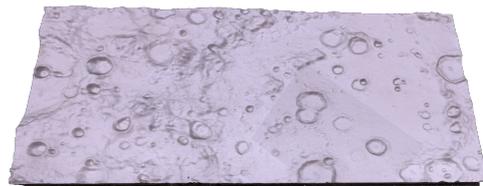


Figure 9: Tile digital reconstruction by means of structure from motion technique.

³Courtesy of NASA and Arizona State University. URL: http://wms.lroc.asu.edu/lroc/rdr_product_select, last visit on: June 1, 2017.

6. CONCLUSION

A suite of tools and algorithms for vision based navigation for autonomous landings in development at PoliMi-DAER has been presented. An hazard detector based on a single camera and artificial neural networks and an efficient semi-analytical adaptive guidance algorithm are in an advanced state, while an efficient vision-based navigation system is under development. An experimental facility based on a robotic arm to simulate lander dynamics and a planetary mock up is being built for their validation and test. The aim is the development and validation of an entire Adaptive GNC chain for autonomous landing: the position of the target selected on the hazard map is transformed in a 3D point in the physical world by the mapping process. The lander states are then utilized by the guidance algorithm to compute a new trajectory and perform a divert maneuver. Furthermore, possible future developments of the testing facility will include also a system update to host real-time hardware-in-the-loop simulation, to test algorithms on flight representative hardware.

REFERENCES

1. J. D. Carpenter, B. Houdou, R. Fisackerly, D. D. Rosa, B. Patti, J. Schiemann, B. Hufenbach, and B. Foing, "Lunar exploration in ESA," in *Annual Meeting of the Lunar Exploration Analysis Group*, Laurel, MD, 2014.
2. T. Tolker-Nielsen, "EXOMARS 2016 - Schiaparelli anomaly inquiry," European Space Agency, Tech. Rep. DG-I/2017/546/TTN, 2017.
3. C. Epp and T. Smith, "Autonomous precision landing and hazard detection and avoidance technology (alhat)," in *Aerospace Conference, 2007 IEEE*. Institute of Electrical and Electronics Engineers, March 2007, pp. 1–7.
4. P. Lunghi, M. Ciarambino, and M. Lavagna, "A multilayer perceptron hazard detector for vision-based autonomous planetary landing," *Advances in Space Research*, vol. 58, no. 1, pp. 131–144, 2016.
5. A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in Neural Information Processing Systems*, 2005, pp. 1161–1168.
6. S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems 2*, D. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 524–532. [Online]. Available: <http://papers.nips.cc/paper/207-the-cascade-correlation-learning-architecture.pdf>
7. F. Hörz, R. Grieve, G. Heiken, P. Spudis, and A. Binder, *Lunar Surface Processes*. Cambridge University Press, 1991.
8. U. Shankar, W.-J. S., T. Criss, and D. Adams, "Lunar terrain surface modeling for the alhat program," in *IEEE Aerospace Conference*, March 2008, pp. 1–10.
9. G. Capuano, M. Severi, E. D. Sala, R. Ascolese, C. Facchinetti, and F. Longo, "Compact and high-performance equipment for vision-based navigation," in *63rd International Astronautical Congress (IAC)*, 2012.
10. P. Lunghi, M. Lavagna, and R. Armellin, "A semi-analytical guidance algorithm for autonomous landing," *Advances in Space Research*, vol. 55, no. 11, pp. 2719–2738, 2015.
11. P. Lunghi, R. Armellin, P. Di Lizia, and M. Lavagna, "Semi-analytical adaptive guidance computation based on differential algebra for autonomous planetary landing," ser. 26th AAS/AIAA Space Flight Mechanics Meeting, Napa, CA, Aug. 2016.
12. G. Flandin, B. Polle, N. Despré, J. Lheritier, N. Perrimon, and P. Blanc-Paques, "Maturing vision based navigation solutions to space exploration," ser. AIAAGNC, Toronto, Ontario Canada, Aug. 2010, AIAA Paper 2010-7601.
13. J. Delaune, D. De Rosa, and S. Hobbs, "Guidance and control system design for lunar descent and landing," ser. AIAAGNC, Toronto, Ontario Canada, 2010.
14. D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011.
15. F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 2, pp. 78–90, 2012.
16. E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.
17. J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001.
18. D. Nistér, "An efficient solution to the five-point relative pose problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 756–770, 2004.
19. R. I. Hartley and P. Sturm, "Triangulation," *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.
20. V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnnp: An accurate o(n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009.