VISION-BASED HAZARD DETECTION WITH ARTIFICIAL NEURAL NETWORKS FOR AUTONOMOUS PLANETARY LANDING

Paolo Lunghi, Marco Ciarambino, and Michèle Lavagna

Politecnico di Milano, Aerospace Science & Technology Department (DAER), via La Masa 34, 20156, Milano, Italy

ABSTRACT

In this paper a hazard detection and landing site selection algorithm, based on a single, visible light, camera acquisition, processed by Artificial Neural Networks (ANNs), is presented. The system is sufficiently light to run onboard a spacecraft during the landing phase of a planetary exploration mission. Unsafe terrain items are detected and arranged in a hazard map, exploited to select the best place to land, in terms of safety, guidance constraints and scientific interest. A set of statistical indexes is extracted from the raw frame, progressively at different scales in order to characterize features of different size and depth. Then, a set of feed-forward ANNs interprets these parameters to produce a hazard map, exploited to select a new target landing site. Validation is carried out by the application of the algorithm to images not considered during the training phase. Landing sites maps are compared to ground-truth solution, and performances are assessed in terms of false positives ratio, false negatives ratio and final selected target safety. Results for different scenarios are shown and discussed, in order to highlight the effectiveness of the proposed system.

Key words: Hazard Detection; Planetary Landing; Vision-based Navigation; Spacecraft Autonomy; Artificial Neural Networks.

1. INTRODUCTION

In recent years, a renewed interest in space exploration had brought to the development of several missions in which the Entry Descent and Landing phase fulfils a critical role. The high level of precision and safety required by the next space missions generation makes the presence of a Hazard Detection and Avoidance (HDA) system mandatory. An autonomous landing system should be able to scan the area around the landing site, to verify if the nominal target can be reached with the required level of safety and, if not, to seek for an alternative safe one. Then, a new landing path toward the updated target should be computed, followed by the execution of the divert maneuver.

Four main criteria concur to determine if a landing site can be classified as safe: visibility by sensors, surface roughness, slopes, and available area. Areas that cannot be analyzed by the sensors system should be classified a priori as unsafe. At the same time, the actual architecture of the lander touchdown system (legs, airbags), determines which are the maximum allowed dimensions of local obstacles and slopes that maintain the probability to avoid damages over tolerable values. Finally, the landing site dimension must be compatible with the lander footprint plus expected uncertainties due to Guidance, Navigation and Control (GNC) system. Poor computational capabilities imposed severe restrictions over early studies about HDA systems. Local variance over an intensity image has been considered as criterion to estimate surface roughness, together with surface major irregularities detection performed by a scanning ranging laser [1]. Later, the development of more powerful systems and specialized hardware paved the way to the development of more complex and accurate hazard detection methods. In the frame of the Autonomous Landing and Hazard Avoidance Technology (ALHAT) project, carried out by NASA since 2006, extensive studies have been conducted on the hazard estimation based on a Digital Elevation Map (DEM) obtained by active ranging sensors, such as Doppler LIDAR and flash LIDAR [2]. Other methods to reconstruct a DEM of the landing area through image processing techniques, such as shape from shading [3], stereo-vision [4] and shadow analysis [5] have been widely investigated.

Recently our research group proved the feasibility of HDA system based on Artificial Neural Networks (ANNs) [6]. ANNs appear particularly attractive for their generalization properties: in fact, once trained with proper data, this kind of systems is able to autonomously determine "fading" rules that describes the phenomenon under investigation. This property is very relevant for hazard detection. In fact, during algorithms development, it is impossible to consider in advance all the types of terrain morphological structures that a landing spacecraft could potentially deal with during operations. At the same time, ANNs working principle relies on a long series of elementary mathematical operations (sums and multiplications), giving them a high computational efficiency, compatible with real-time systems.
In this paper, a further step in the development of an

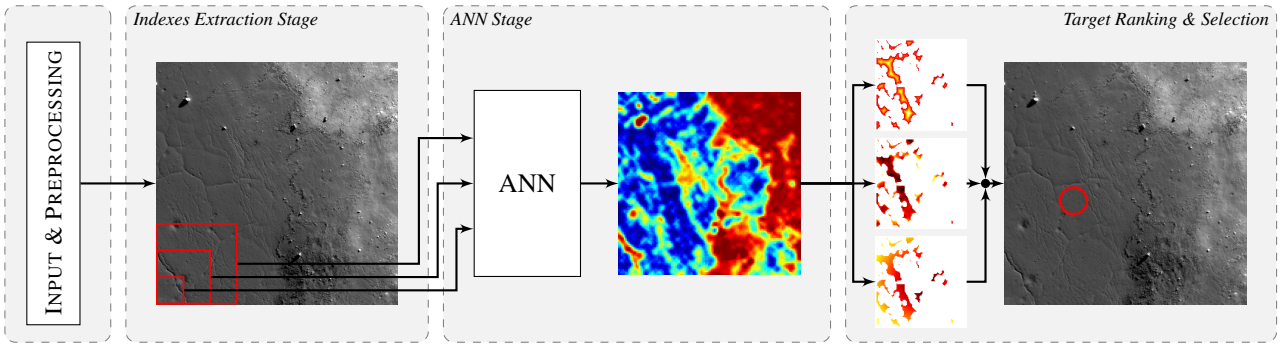


Figure 1: Hazard Detection system neural networks logical scheme. A back propagation feed-forward network estimates hazard index value from elementary information extracted from the image at different scales.

affordable ANN-based hazard detection system is presented. The paper is structured as following: in Section 2 the system architecture is described. The generation of ground truth models for system training and validation are explained in Section 3. In Section 4 obtained results and performances are assessed, while conclusions and suggestions on future developments are expounded in Section 5.

2. SYSTEM ARCHITECTURE

In Fig. 1 the logical scheme of the hazard detection system is shown. For each position of the landing area, this system assigns a hazard index, giving a measure of the safety of that position if chosen as landing site. Hazard index can assume any value between 0 (completely safe) and 1 (absolutely hazardous). The hazard map computation consists in 4 stages:

1. Preprocessing: the raw image is acquired. Image perspective correction is applied if needed.
2. Indexes extraction: image is segmented at different scales and low level information is extracted.
3. Image elementary indexes are processed by a feed-forward back-propagation neural network and arranged in a hazard map.
4. Target Landing Site search. Computed hazard map is exploited to select the most attractive landing site.

In the following sections each stage is expounded in details.

2.1. Input and Preprocessing

Grayscale (8 bit, single channel) images have been considered as system input. An image size of 1024×1024 pixels (compatible with most of the present camera devices for use in space) has been adopted.

The system considers images taken from a near vertical descent. Small deviations from the vertical attitude can be corrected by the application of a perspective transformation just before the hazard map computation. This type of transformation assumes that the scene can be considered as flat. The presence of orographic reliefs introduces an approximation acceptable only for a limited angle of view. Anyway, further corrections can be applied if additional information are available, e.g. by a vision-based navigation system. Plus, in order to perform hazard detection and avoidance tasks, the landing site area is required to remain in the lander cameras field of view for a certain amount of time during the approach phase, and maneuverability requirements on the landing trajectory are imposed to maximize the lander divert capabilities. These requirements exclude tightest trajectories [7], like the ones exploited during the Apollo missions [8], limiting in this way the maximum view angle during hazard detection phase.

2.2. Indexes Extraction and ANN Input Building

Indexes extraction is a key feature for the correct hazard detection with neural networks. They have to be selected in order to detect correctly slopes, craters, scraps and any kind of obstacle that make a site non-suitable to land on. To achieve such an accomplishment with a single camera, it has been considered to use both zeroth and higher differential orders of the gray-scale image. Zeroth order allows to have a reference value from which the network can understand the general brightness of the area analyzed in the acquired image. First and second derivatives are obviously dedicated to detect the various features present on planetary surface detecting variations of pixel intensity. Moreover, the exploitation of various image scales (downsampling the original one) helps the neural network understanding depth and relative distances inside the image[9]. Size of these image windows side has been selected trading off computational time and final hazard map resolution, that is given by the original image size divided by small window size. It has been opted for a small, medium and large window sides respectively of $s_S = 4$, $s_M = 8$, $s_L = 16$ px, therefore a final

hazard map resolution of 256×256 px. In the current architecture, indexes provided to the neural network can be split in two categories:

- *Window-based* indexes: mean μ and standard deviation σ of the current window pixels are computed. This process is performed over each of the three window sizes. The two statistical indexes are defined as:

$$\mu = \frac{\sum_{i=1}^{N} I_i}{N}, \tag{1}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (I_i - \mu)^2}{N-1}}, \tag{2}$$

  where I_i corresponds to the intensity of the i-th pixel, N is the number of pixels inside the considered image window.

- *Global* indexes: image gradient (grad) and Laplacian of Gaussian (LoG) are computed across the whole original acquired image through custom kernels convolution. Then, both resulting matrices are downsampled to 256×256, 128×128 and 64×64, in order to be able to assemble all the indexes in a single matrix that represents the input for the neural network. Grad is approximated through an expanded 5×5 Prewitt filter for both horizontal and vertical directions[10]. Then, the square root of the sum of the square of every element of directional gradients yields the total image gradient. Laplacian of Gaussian is a second order operator widely used as edge detector[11]. It combines a Gaussian smoothing with the Laplacian operator and its general formulation in continuous space is:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3}$$

  where x, y represents image coordinates, while standard deviation σ determines the characteristic length at which the filter tends to reject noise. Here it has been implemented through an approximated 5×5 kernel.

In addiction to these indexes, also *Sun inclination angle* is assembled in the input matrix. This is necessary to make the neural net correctly compute hazard maps with both sharp and blunt shadows. Summarizing, indexes are 13 in total: μ, σ, Grad and LoG for each of the 3 considered scales, plus the Sun inclination angle. Every kind of input is normalized to unit value. Eventually, assembling of the whole input is concluded expanding indexes relative to bigger image windows and higher downsample levels because of their intrinsic smaller size.

2.3. Artificial Neural Network and Hazard Map Assembly

After assembling, input is processed by a Feed-Forward Back Propagation Neural Network. It consists in a very simple *single hidden-layer* unit of *15 neurons* with hyperbolic tangent function. Output is a scalar value and represents the intensity of a single pixel of the hazard map. Thus for each input sample of 13 values, a single double number depicting the hazard value is computed. Numerical values are bounded between (0,1) through a logarithmic sigmoid function, where one represents a totally unsafe hazard map pixel, zero a completely safe one.
To relate nearby pixels, a light blur filter is applied directly on the resulting hazard map. The size of the neural

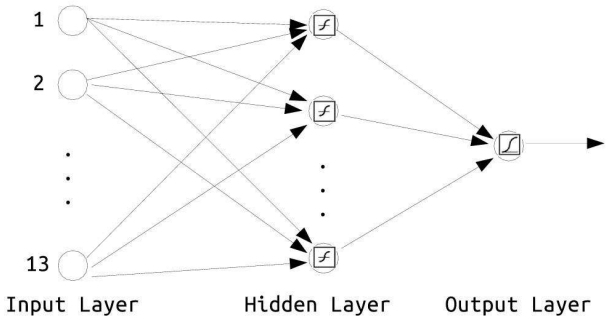


Figure 2: Sketch of the used Neural Network.

network came out to be so conveniently small after comparisons with much bigger counterparts: due to the fact that almost no performance increase was registered up to nets made of three hidden layers with fifteen neurons each, it has been opted to privilege the simplest and lightest architecture. Therefore, the training time was beneficially reduced too.

2.4. Target Selection

Once the hazard map is computed, the system seeks for a safe landing site. Possible sites are classified and ranked according to the following drivers:

- Minimum hazard index
- Maximum landing area
- Minimum distance from nominal landing site (required to maximize the probability to find a landing site actually reachable by the lander divert capabilities)

In order to do that, for each of these principles, a specific index is assigned to each landing site candidate. Then, the three indexes are fused in a unique score exploited to create a global landing site ranking. The image reference frame is considered. This frame is centered in the pixel at the upper-left image corner, aligned with the image borders. Distances in image reference frame can be expressed in pixel units (the position of each pixel corresponds to the number of its column and row, numerated

from 0) or in real units (meters). The transformation between them is a simple scaling conversion:

$$\mathbf{r} = d_{\text{res}} s_{\text{s}} \mathbf{x}, \tag{4}$$

where $\mathbf{r}^T$ is the image position vector expressed in meters, $\mathbf{x}^T$ is the same vector expressed in pixels units, d_{res} is the original image resolution, expressed in meters per pixel, and s_{s} is the length, in pixels, of the "small" window considered by neural networks during hazard map computation. It is remarkable that a sufficiently good estimation of the image resolution is required in order to correctly evaluate areas of possible landing sites. Then, at least information about the altitude, camera field of view, and attitude (the minimum required for image perspective correction and resolution estimation) are assumed to be available to the system.
The following procedure is adopted:

1. The hazard map is thresholded at the maximum tolerable level of hazard index for a safe landing site, denoted as z_{max}. All the pixels above the threshold are classified as unsafe. Pixels under this level are initially classified as Candidate Landing Site (CLS).

2. For each pixel at coordinates $\mathbf{x}^T = [i, j]$ with $z_{ij} \leq z_{\text{max}}$ the *Size Score* $r_{\text{CLS}ij}$ is computed as the distance from the nearest unsafe pixel.

3. A safe landing site is required to respect a minimum dimension requirement. Modeled as a circle, its radius is required to be:

$$r_{\text{CLS}ij} \geq r_{\text{min}} = \frac{d_{\text{foot}}}{2} + e_{\text{gnc}}, \tag{5}$$

   where d_{foot} is the lander footprint diameter, and e_{gnc} is the expected landing error due to navigation imprecision, with the desired level of confidence. Then, all CLSs that do not respect this constraint are discharged.

4. For each CLS the *Diversion Score* $d_{\text{CLS}ij}$ is computed as the distance from the Nominal Landing Site (NLS):

$$d_{\text{CLS}ij} = \|\mathbf{r}_{\text{CLS}ij} - \mathbf{r}_{\text{NLS}}\|, \tag{6}$$

   where $\mathbf{r}_{\text{CLS}ij}$ is the metric position of the CLS at image coordinates, and $\mathbf{r}_{\text{NLS}}$ is the metric position of the NLS.

5. The *Safety Score* $z_{\text{CLS}ij}$ of each CLS is obtained by the mean of the hazard index of the pixels contained in the circle centered at the CLS_{ij} of radius $r_{\text{CLS}ij}$.

6. The three scores are normalized to make their values to the same order of magnitude. Than, a global score $l_{\text{CLS}ij}$ is obtained as:

$$l_{\text{CLS}ij} = \mathbf{w}^T \begin{bmatrix} \tilde{r}_{\text{CLS}ij} \\ 1 - \tilde{d}_{\text{CLS}ij} \\ 1 - \tilde{z}_{\text{CLS}ij} \end{bmatrix}, \tag{7}$$

   where $\mathbf{w}^T$ is a vector of weights, introduced in order to give to the user the faculty to confer more relative importance to one index with respect to others. Symbols marked with a tilde stand for normalized values.

7. Finally, the CLS with the larger global score is selected as Target Landing Site (TLS).

3. NETWORK TRAINING

ANNs performance depend widely on the completeness and coherence of the dataset used to train the network. In the specific case of HDA systems, the type of training set must be also tailored on the celestial body target of the mission. In this paper, a lunar landing case is considered. True lunar images present several criticalities: image metadata (model, position, and attitude of the camera when the picture is taken) are scarcely available, as well as detailed model of the terrain depicted, required to obtain the correspondent ground truth solution. On the contrary artificial images make possible an objective and precise ground truth reconstruction, being all the setting and the 3D model used for image generation completely known. Despite that, an high level of photorealism is required to preserve coherence.
In this work, a dataset of 98 images has been generated and exploited for training and testing purposes. Images are taken from random altitude between $2000\,\text{m}$ (an altitude inside the interval in which HDA system is required to operate [12]), and they are divided into 3 subsets: training, validation and test. The training set (67 images) is exploited to directly optimize the network weights with a backpropagation algorithm; network overfitting is avoided when the RMS error, evaluated on the validation set (23 images), reach a minimum (early stopping method). The test set, made up of 8 images, is exploited to assess the system performances (see Sec. 4).

3.1. Training Set Building

High resolution lunar DEMs obtained from LROC data[1], with a variable resolution between 2 and 5 m/point, have been used as starting point for the creation of artificial images. First, DEMs resolution is improved up to $0.3\,\text{m}$/point, by adding fractal noise, small craters and boulders [13]. Craters deposition respects the statistical distribution observed on the real lunar surface, as well as the real craters formation process [14], while craters morphology follows empirical morphometric relations obtained from lunar imagery [15]. Finally, the camera frame is rendered in *POV-Ray* [2] with the desired settings for camera, position, and attitude with realistic

[1] Courtesy of NASA and Arizona State University. http://wms.lroc.asu.edu/lroc/rdr_product_select, last visit on: May 4, 2015.

[2] Persistence of Vision Raytracer (Version 3.7) [Computer software]. Retrieved from http://www.povray.org/download/

illumination conditions. A pinhole camera model, with a $60°$ angle of view has been adopted.

3.2. Ground-truth Solution Computation

Slopes and roughness can be extracted directly from DEM data. For each DEM point, a surrounding circular window with diameter equal to the lander footprint is considered: the slope is computed as the inclination of the mean plane obtained by a least squares approximation of the points in the window. The plane is expressed by the equation:

$$Z = aX + bY + c, \tag{8}$$

where X and Y are the coordinates of the points in the window, and Z is the altitude. Then, the plane inclination S is obtained as:

$$S = \tan^{-1}\big(\sqrt{a^2 + b^2}\big). \tag{9}$$

Roughness R is estimated as the difference between the maximum and the minimum deviation of the window points from the mean plane:

$$\begin{aligned} R = \max\big(Z_i - (aX_i + bY_i + c)\big) \\ - \min\big(Z_i - (aX_i + bY_i + c)\big), \end{aligned} \tag{10}$$

where the subscript i denotes the i-th DEM point inside the window. Once slopes and roughness maps are available, they are converted in camera image coordinates by a perspective transformation, computed through rendering software. Then, they are exploited to obtain the correspondent ground-truth hazard map. Each point is considered safe if respect the following conditions:

- $S \leq S_{\text{max}}$,
- $R \leq R_{\text{max}}$,
- the point is not in shadow.

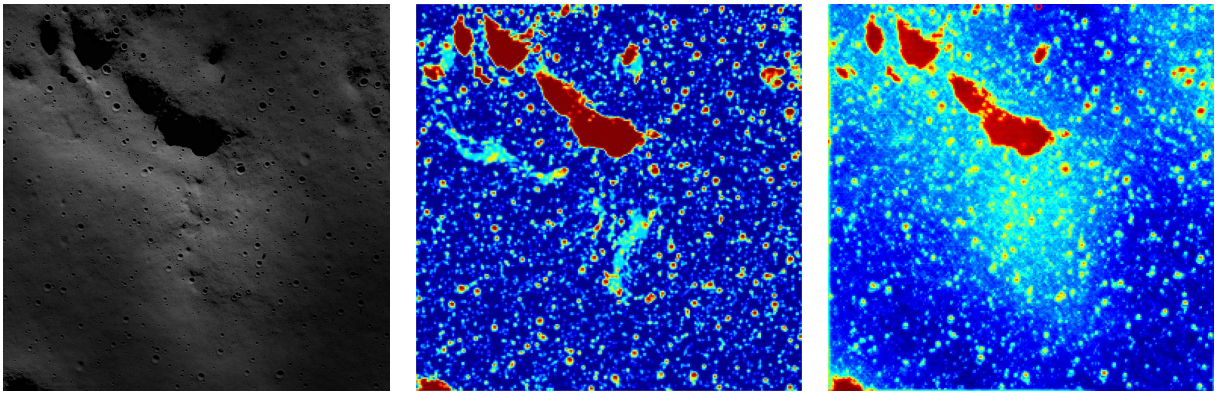
Shadow map is easily obtained by the application of a threshold on the camera image histogram. At each pixel of the camera image is assigned the hazard index 0 (perfectly safe) if respects all the conditions mentioned above. Pixels in shadow are considered as out of the sensor range, and are then considered as completely unsafe (hazard index 3). Hazard index 1 is assigned to those pixels that fail only one of the tests on slope and roughness, while the value 2 is assigned to those ones that fail both the tests. Then, the obtained hazard map is normalized to bring back the hazard index in the interval $[0, 1]$. At this stage the ground-truth hazard map has the same resolution of the camera image. In the last step, the map size is decreased up to the resolution computed by ANNs (256×256 for a 1024×1024 px frame) by applying a Gaussian pyramid. The downsampling process increases the hazard map smoothness, making easier the ANNs training process (feed-forward neural network are less effective in reproducing discontinuous functions). Figure 3a reports an example of artificial image, obtained from a real DEM of the Manilius crater floor, while the correspondent ground truth hazard map is depicted in Fig. 3b. Based on this hazard map, is possible to compute the true safety of the landing sites with the algorithm presented in Sec. 2.4.

4. PERFORMANCE ASSESSMENT

The system performance is verified by comparison with the ground truth solution of a test set, which consists of four landscapes with two different sun inclination angles ($15°$ and $80°$) for a total of 8 images. For each image, from the correspondent ground truth hazard map, the pull of the real safe landing sites is computed through the procedure described in Sec. 2.4 with a safety threshold $z_{\text{max}} = 0.30$. A footprint $d_{\text{foot}} = 3\,\text{m}$ and a navigation error $e_{\text{gnc}} = 15\,\text{m}\ (3\sigma)$ have been considered.
ANNs are not expected to exactly reproduce the original ground truth hazard map; instead an approximation of them is expected. In Fig. 3 an example of hazard map computed by the ANN system is shown. It is possible to see how all the large scale hazardous features are correctly detected; the network response tends to be conservative, with a mean hazard index higher than the ground truth solution. For these reasons, the safety threshold z_{max} used in the target search is not required to be equal to the value used in ground truth computation; plus, the actual ranking of the landing sites is affected by the choice of the weights vector $\mathbf{w}$. The 4 parameters (the safety threshold plus the 3 weights) should be then tuned in order to reproduce the desired behavior. The adopted optimization criteria are:

- The first ranked target landing site shall be always a true positive. Also if the distance from the NLS is taken into account in the sites ranking, there is always a possibility that the guidance is not able to find a feasible trajectory to the selected target. Then, this requirement is extended to the first 2 landing sites to have always a backup target.
- The total number of false positives should be minimized;
- The total number of false negatives should be minimized.

A multi-objective optimization has been performed to optimize the algorithm capabilities in terms of safety of the landing site. The optimization has been run in Matlab® environment with an elitist variant of NSGA-II algorithm [16] with 100 individuals. As preliminary attempt, 2 objective functions have been taken into account:



(a) Original frame. (b) Ground truth hazard map. (c) ANN computed hazard map and TLS.

Figure 3: Hazard map computation. Comparison between original image (a), ground truth (b) and computed solution (c). It can be seen as ANNs tends to be conservative with a more diffuse hazard index. The small red circle at the top of figure 3c indicates the TLS found.

- The sorting position of the first false positive target (to be maximized);
- The maximum number of false positives in the whole test (to be minimized).

The following optimization constraints have been considered:

- hazard index threshold between 0.20 and 0.30;
- landing site selection weights between 0 and 1;
- sum of landing site weights equal to 1.

At the end of the optimization 35 quadruplets (hazard threshold, weights) resulted optimal. Among these optimal values, it has been selected a threshold of 0.24 and weights equal to (0.0067581, 0.50467, 0.48794). With such values, performances on the 8 images of the test set are evaluated. At least 4 true positives landing sites has been identified in every test image, with a peak of 438. An average of 5634 false negatives has been recorded, mainly because of the very low hazard threshold adopted. An average of *6.5 false positives* landing sites occurred in the whole set. For 3 test images no false positives at all were recorded. In the remaining 5, the *average value of the first false positive position in landing site candidates ranking* resulted 119.4, with only one critical case in which occurred to be position 3.

4.1. True Images

The very same neural network and algorithm has been also tested on real lunar images and photos taken by Rosetta mission of the 67P/Churyumov–Gerasimenko. Being unknown the Sun inclination angle, it has been briefly hypothesized looking at the photos. Moreover, in these images there is no ground truth to quantitatively test the hazard maps with. Thus, results are to be intended just as an example of the generalization capabilities of the neural network and must be not intended as a valuable result of the hazard detection system. Anyway, the choice of the photos was dictated by the presence of relevant morphological features, that could have challenged the system.

Moon In Fig. 4, taken by LROC Narrow Angle Camera, depicting part of the Larmor Q crater floor, it is possible to spot some fractures on the surface in the lower left hand side half, while the rest of the image is characterized by diffuse roughness due to craters. In its relative hazard map (Fig. 5), the neural network seems to have qualitatively understood the terrain features, assigning a distributed high hazard value to the rough region at top right hand side and about maximum value precisely where fractures are located.

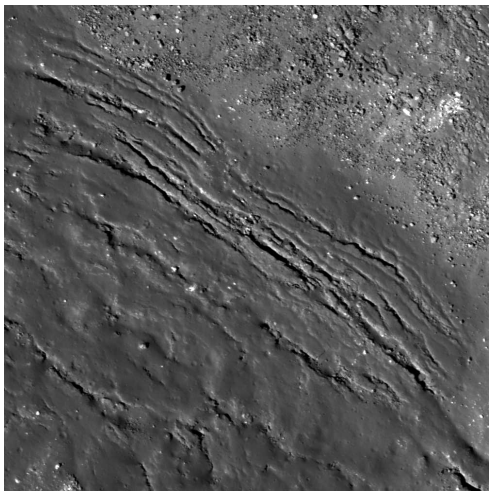


Figure 4: Lunar real surface image, Larmor Q crater floor, NAC frame M151726155R, courtesy of NASA/GSFC/ASU.

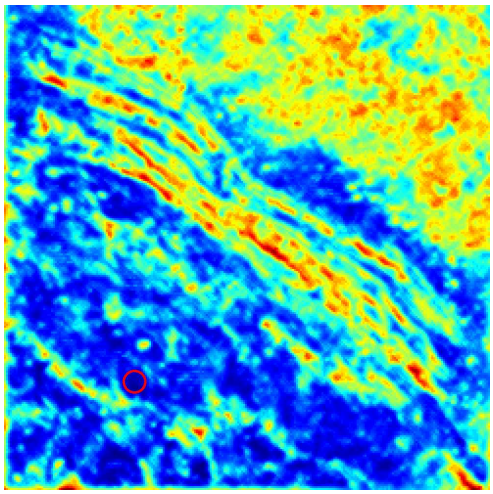


Figure 5: Computed hazard map relative to Fig. 4

67P/Churyumov–Gerasimenko The great interest of both the scientific community and companies in small celestial bodies pushed to test the same hazard system used for lunar images on 67P/C-G. Not many suitable images are available for the purpose, and even less are equipped with data the neural network should need to be as much efficient as it can. A test on the Imhotep region is presented in Fig. 6. Such an area is composed by many well distinct features: a planar plateau with sharp boulders and rifts, developing from the center to the top of the picture, surrounded by an irregular area full of craters and high sloped sides. In the relative computed hazard map (Fig. 7), the system seems to have qualitatively understood hazard trends of the various areas: deep blue (safe) for the planar area apart from the irregularities, green and red (unsafe) for the most of the rest.

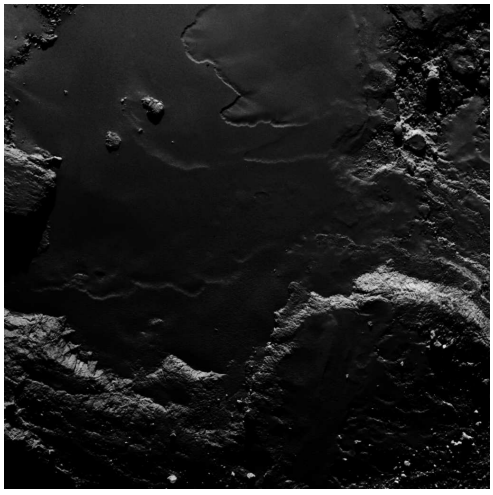


Figure 6: Comet 67P Churyumov-Gerasimenko, Imhotep region (Photo: ESA).

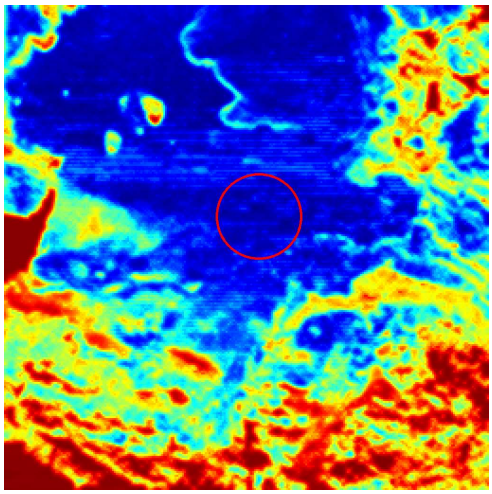


Figure 7: Imhotep region computed hazard map.

4.2. Profiling

To properly estimate the computational weight of the proposed HDA system, a profiling analysis has been carried out. *Gperftools*, a tool released by Google under BSD license, has been selected as main profiler; as verification, obtained result have been crosschecked with the C++ standard library's CPU time computation. All tests have been performed on a Intel® Core™ i7-4712HQ CPU @ 2.3 GHz running 64 bit Ubuntu 14.10 GNU/Linux operative system. In each profiling test, the hazard detector runs in a cycle for 1000 times, while the sampling frequency has been set to 250 Hz (the highest possible value) to maximize the precision in runtime estimation. In order to avoid modern processors' automatic multi-core computation, the system has been forced to run in single-thread configuration. Gperftools registered 56 666 hits at 250 Hz, for a total time of 226.66 s, while the correspondent CPU time resulted 235 s. Taking into account the possible overhead that can affect measurements differently with the two methods, the values are comparable. The mean obtained runtime is 230 ms. The principal bottleneck is identified in the indexes extraction, that requires more than the 65% or the total runtime. This result agrees with the expected: image processing algorithms, that constitute most of the task, are computationally expensive. Recent developments of dedicated space qualified hardware [17, 18] allow to expect further improvements in real applications up to real-time performances.

5. CONCLUSIONS

In this papers, a completely revised design of a previously implemented HDA algorithm, based on Artificial Neural Networks, is proposed. A deep analysis to detect which information can be extracted from the original image and exploited as network input has been carried out. The exploitation of more informative indexes from the

original image dramatically improves both classification capabilities and computational performances, allowing a drastic simplification in the neural network architecture with respect to previous configurations. A fully objective training and validation method has been developed, in order to avoid any dependency of the system performance from the operator's choices during the training phase and to have an affordable estimation of the system capabilities. Further improvements are currently under study at PoliMi DAER: alternative neural network structures, such as cascade neural network, should be investigated and compared to the feed-forward architecture. Also multi-objective optimization of internal parameters can be enhanced by the addiction of the minimization of false negative sites to assure fully compliance with the requirements stated in Sec. 4. Eventually, in order to definitely increase the TRL of this technology, algorithm hardware porting and hardware-in-the-loop testing are required. A landing simulation facility for vision based navigation systems is currently under development, scheduled to be operative at DAER premises by the Autumn 2015.

REFERENCES

[1] H. Pien, "Autonomous hazard detection and avoidance for mars exploration," in *8th AIAA Computing in Aerospace Conference*, 1991.

[2] N. Trawny, J. M. Carson, A. Huertas, M. E. Luna, V. E. Robak, A. E. Johnson, K. E. Martin, and C. Y. Villalpando, "Helicopter flight testing of a real-time hazard detection system for safe lunar landing," in *AIAA SPACE Conference and Exposition*, 2013.

[3] B. Parreira, E. Di Sotto, A. Caramagno, and J. Rebordão, "Hazard avoidance for planetary landing: Gnc design and performance assessment," in *7th International ESA Conference on Guidance, Navigation & Control Systems*, 2008.

[4] S. Woicke and E. Mooij, "Stereo-vision algorithm for hazard detection during planetary landings," in *AIAA SciTech*, 2014.

[5] L. Matthies, A. Huertas, Y. Cheng, and A. Johnson, "Landing hazard detection with stereo vision and shadow analysis," in *AIAA Infotech@Aerospace Conference and Exhibit*, 2007.

[6] P. Lunghi and M. Lavagna, "Autonomous vision-based hazard map generator for planetary landing phases," in *65th International Astronautical Congress (IAC)*, pp. 1–12, 2014.

[7] E. Wong, J. Masciarelli, and G. Singh, "Autonomous guidance and control design for hazard avoidance and safe landing on mars," in *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2002.

[8] A. R. Klumpp, "Apollo lunar descent guidance," *Automatica*, vol. 10, no. 22, pp. 133–146, 1974.

[9] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in Neural Information Processing Systems*, pp. 1161–1168, 2005.

[10] H. B. Kekre and S. M. Gharge, "Image Segmentation using Extended Edge Operator for Mammographic Images," *International Journal on Computer Science and Engineering*, vol. 2, no. 4, pp. 1086–1091, 2010.

[11] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*. Newnes, 2002.

[12] 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.

[13] U. Shankar, W.-J. S., T. Criss, and D. Adams, "Lunar terrain surface modeling for the alhat program," in *IEEE Aerospace Conference*, pp. 1–10, March 2008.

[14] C. A. Cross and D. L. Fisher, "The computer simulation of lunar craters," *Monthly Notices of the Royal Astronomical Society*, vol. 139, no. 2, pp. 261–272, 1968.

[15] F. Hörz, R. Grieve, G. Heiken, P. Spudis, and A. Binder, *Lunar Surface Processes*. Cambridge University Press, 1991.

[16] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[17] 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.

[18] M. Dunstan and K. Hornbostel, "Compact and high-performance equipment for vision-based navigation," in *9th International ESA Conference on Guidance, Navigation, and Control Systems (GNC 2014)*, 2014.