P. Lunghi, M. Ciarambino, M. Lavagna
A Multilayer Perceptron Hazard Detector for Vision-Based Autonomous Planetary Landing
Advances in Space Research, Vol. 58, N. 1, 2016, p. 131-144
doi:10.1016/j.asr.2016.04.012

A Multilayer Perceptron Hazard Detector for Vision-Based Autonomous Planetary Landing

Paolo Lunghi[1,*], Marco Ciarambino[2], Michèle Lavagna[3]

Aerospace Science & Technology Department, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy

Abstract

A hazard detection and target selection algorithm for autonomous spacecraft planetary landing, based on Artificial Neural Networks, is presented. From a single image of the landing area, acquired by a VIS camera during the descent, the system computes a hazard map, exploited to select the best target, in terms of safety, guidance constraints, and scientific interest. ANNs generalization properties allow the system to correctly operate also in conditions not explicitly considered during calibration. The net architecture design, training, verification and results are critically presented. Performances are assessed in terms of recognition accuracy and selected target safety. Results for a lunar landing scenario are discussed to highlight the effectiveness of the system.

Keywords: autonomous landing, hazard detection and avoidance, 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

*Corresponding author

Email addresses: paolo.lunghi@polimi.it (Paolo Lunghi), marco.ciarambino@mail.polimi.it (Marco Ciarambino), michelle.lavagna@polimi.it (Michèle Lavagna)

[1]Ph.D. Candidate

[2]Research Assistant

[3]Associate Professor

phase fulfills a critical role. Such phase can be often considered as a bottleneck in space missions: a failure encountered during the landing would lead with high probability to the complete loss of the spacecraft. In these cases, safety is the main driver in mission analysis and design process. Historically, high uncertainties in attainable position at touchdown had always imposed severe requirements on the landing site choice. The selection process is very complex, with the strong limitation of fitting the absolute landing site dispersion ellipse in a safe area (Masursky and Crabill, 1976; Arvidson et al., 2008; Spencer et al., 2009; Golombek et al., 2012; Ulamec et al., 2015). In spite of that, in many cases scientifically relevant places are associated with hazardous terrain features or confined in small areas; in other cases there is no possibility to completely characterize an interesting region with the required accuracy. The possibility to adapt the trajectory during the descent would reduce the landing dispersion, making possible the execution of absolute correction maneuvers. At the same time, in conjunction with the capability to distinguish hazardous from safe landing areas, the safety criteria during the mission analysis could be relaxed, leaving to the system the task of Hazard Detection and Avoidance (HDA). Such a 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 and reachable one. Then, a new relative landing path toward the updated target should be computed, followed by the execution of a divert maneuver avoiding in this way local obstacles and slopes.

Hazard detection studies were carried out in parallel with the development of affordable trajectory computation methods. One of the first systems capable of retargeting was introduced in the Apollo Program: in that case, the target selection relied on human intervention (Klumpp, 1974). In subsequent years, with the increase of the available computational power, more complex and optimal guidance algorithms were investigated. A derivative of the Apollo lunar descent guidance was later proposed by Wong et al. (2002) for the Mars Science Laboratory (MSL), while another variant of this explicit scheme based on a polynomial formulation of the acceleration – called *E-Guidance* – has been recently considered by Parreira et al. (2007) to accomplish HDA tasks. Direct numerical methods for trajectory optimization were widely investigated, not requiring the explicit consideration of the necessary conditions and with better convergence properties (Betts, 1998). These methods were used together with Chebyshev pseudospectral techniques, to allow the reduction of the number of the optimization variables (Fahroo and Ross,

2002). Açikmeşe and Ploen (2007) considered convex programming to guarantee the convergence of the optimization; this approach, coupled with direct collocation methods, has proved that the size of the region of feasible initial states, for which there exist feasible trajectories, can be increased drastically (more than twice) compared to the traditional polynomial-based guidance approaches, but at the price of a higher computational cost. This method was coupled with a minimum-landing-error approach, in order to compute a landing trajectory even in case a feasible solution for the selected landing site is not found (Blackmore et al., 2010). Lunghi et al. (2015) extended the polynomial formulation leaving a limited number of parameters free in to obtain a suboptimal solution and to cope with trajectory constraints, maintaining at the same time a light computational weight. Wibben and Furfaro (2016) augmented a zero-effort-miss/zero-effort-velocity with sliding mode control to obtain a guidance algorithm robust to uncertainties.

Investigations on hazard detection algorithms followed a similar development. Early studies on HDA systems exploited very simple principles: Pien (1991) considers local variance over an intensity image as criterion to estimate surface roughness, together with surface major irregularities detection performed by a scanning ranging laser. 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, as shown by Trawny et al. (2013). A proposal to include also scientific criteria in the selection process is done by Furfaro et al. (2012) exploiting soft computing techniques. Other methods to reconstruct a DEM of the landing area through image processing techniques, such as shape from shading (Parreira et al., 2008), stereo-vision (Woicke and Mooij, 2014) and shadow analysis (Matthies et al., 2007) have been widely investigated.

Four main criteria concur to determine if a landing site can be classified as safe: visibility by sensors, surface roughness, slopes, and size of the safe area. Areas that cannot be analyzed by the sensors system should be classified a priori as unsafe; considering systems based on visual information, areas in shadows are included in this category. 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. Plus, also if a target is found safe, it could be impossible to be reached, due to the limited control authority of the spacecraft. Then, also the probability to find a feasible trajectory to the target should be taken into account in the selection process.

Recently our research group proved the feasibility of an HDA system based on Artificial Neural Networks (ANNs) (Lunghi and Lavagna, 2014). 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 describe the phenomenon under investigation (Hornik, 1991). 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 major step in the development of an affordable ANN-based hazard detection system is presented, with an optimized architecture and the development of a full objective method to train and test the system. It is assumed that the system receives as input only images from a monocular camera and some basic telemetry, including spacecraft altitude and attitude. The aim is to demonstrate the robustness and the effectiveness of a neural networks based system with minimal available information. In a real case, ANNs can be provided with additional input from different sources (LIDARs, feature tracking systems, stereo cameras etc.), making the system even more effective.

The paper is structured as following: first, in Section 2, the system architecture is described; two different structures of ANNs are considered. Then, the generation of ground truth models for system training and validation are explained in Section 3, and obtained results and performances are assessed in Section 4. Finally, conclusions and suggestions on future developments are expounded.

1.1. Nominal HDA Maneuver

In this section, some assumptions about the operations sequence during a planetary landing maneuver are expounded. HDA operations are assumed to be executed by a landing spacecraft during the so-called Approach phase, starting at an altitude between 2500 and 1000 m, as the nominal landing site comes into sensors' field of view, depending on the landing strategy and on the specific target celestial body. The actual required performances and operative conditions of a hazard detection system largely depend on the specific operations planned to be performed in this phase.

For this work, a lunar landing is considered. The assumed operation sequence (summarized in Figure 1) is the following:

1. HDA maneuvers starts at the beginning of the Approach Phase. At this point, called HDA High Gate, the lander is assumed to have a near vertical attitude, a vertical downward velocity in the range $15-30\,\mathrm{m\,s^{-1}}$, and horizontal velocity $5-15\,\mathrm{m\,s^{-1}}$.
2. The system performs a Large Scale Hazard Avoidance Maneuver:
   (a) The system scans the landing area and builds a large-scale hazard map. A new target landing site is selected.
   (b) The landing trajectory is computed and a diversion maneuver is commanded. The final target of this maneuver is a point, called HDA Low Gate, located on the vertical to the target, at altitude $250-500\,\mathrm{m}$. The target velocities are $10-5\,\mathrm{m\,s^{-1}}$ and $2-4\,\mathrm{m\,s^{-1}}$ respectively in vertical and horizontal directions, with vertical attitude.
3. Then, the system performs a Small Scale Hazard Avoidance Maneuver:
   (a) The system scans again the landing area, building a small-scale hazard map. If required by hazard detection, the target landing site is updated.
   (b) The trajectory is updated and if needed a new diversion maneuver is commanded. The target point, called Terminal Gate, is located at the target landing site, at altitude $30-50\,\mathrm{m}$. At the Terminal Gate the lander is required to have a vertical attitude, with a null horizontal velocity ($\leq 1\,\mathrm{m\,s^{-1}}$) and a small vertical speed $\leq 3\,\mathrm{m\,s^{-1}}$.
4. At the Terminal Gate the Approach phase ends and the Terminal Descent phase starts. In this phase, the lander follows a vertical trajectory at constant speed until touch down.

2. System Architecture

In Fig. 2 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.

Due to the kind of the considered Approach maneuver, it is assumed that HDA systems operate in a near vertical attitude. Small deviations from nadir pointing are corrected by the application of a perspective transformation just before the hazard map computation.

The pixel size is not uniform across the image for an inclined view: more distant objects appear smaller. In the perspective correction process the frame is rectified and re-sampled to allow all the subsequent computation, but no additional information is actually recovered. Then, the image resolution *before* resampling should be high enough to allow the system to correctly detect hazardous features, in every part of the frame. The resolution requirement is different at different altitudes: for a large scale maneuver, the pixel size should not exceed the order of magnitude of the lander footprint, while for lower altitudes, corresponding to a low scale hazard detection, it should not be greater than the largest obstacle dimension handleable by the landing system. Assuming a 60° field of view camera and a 1024×1024 pixels sensor, the pixel size for a perfect down-looking attitude is equal to 2.25 m

uniformly on the whole image, taken from 2000 m altitude. For an inclined view, this value increases (at worst) up to 2.88 m for a deviation from nadir of 10°. After this value, the pixel size becomes quickly very large, increasing proportionally to the tangent of the view angle. Considering also that, with the same inclination, the pixel size increases linearly with the altitude, angles up to 10° are considered acceptable.

This type of transformation assumes that attitude information is known by the system, and that the scene can be considered as flat. Errors in attitude and altitude estimation, and presence of orographic reliefs introduce approximations whose impact over the system is discussed more deeply in Section 4.3. Anyway, further and more complex distortion corrections can be applied if additional information are available, e.g. terrain shape model by a vision-based navigation system.

Plus, in order to perform hazard detection and avoidance tasks, the landing site region is required to come into the sensor's field of view with sufficient time and control margin to maximize the lander divert capabilities. This requirement excludes too inclined trajectories (Wong et al., 2002), like the ones exploited during the Apollo missions (Klumpp, 1974), limiting in this way the maximum view angle during hazard detection phase.

2.2. Information Extraction and ANN Input Assembly

Indexes extraction is a key feature for a correct hazard detection with neural networks. It consists in the extraction of low level information from the raw image, with the aim to reduce the space of the data in which neural network can detect morphological features of the real world. The smaller the space, the more effective the network training process, together with a reduction of the system's complexity. At the same time, an inappropriate choice of the indexes could lead to an excessive loss of information, with consequent loss of accuracy in the results.

A single-channel image can be considered as a discrete two-dimensional function where the intensity on the ith pixel I_i depends on the two spatial coordinates:

$$I_i = f(x_i, y_i). \tag{1}$$

Then, it is possible to obtain derivative information with the appropriate discrete filters. Zeroth order information 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 instead

indicators of the presence of specific features on planetary surface detecting variations of pixel intensity. Four different quantities were selected: mean, standard deviation, gradient (1st derivative) and Laplacian of Gaussian (estimation of the 2nd derivative). All the selected entities can be computed by the application of different linear filters to the image. In this work, only the most basic quantities have been considered, trying to maintain the computational complexity at a minimum. More sophisticated techniques to extract information from images actually exist, such as independent component analysis (Hyvärinen et al., 2001), principal component analysis (Schölkopf et al., 1998), and Gabor filters (Daugman, 1988). Despite their high efficiency in carrying information, they are also more demanding in terms of computational weight: their study is left for future developments (see Section 5 for a more complete discussion).

Each quantity can be computed in two different ways: by segmentation or globally. In the first case, the input image is segmented in sub-windows: the index is computed for the whole bounce of pixels included in each of them. The resulting matrix of indexes has a dimension equal to the size of the input divided by the size of the adopted window. In the global case, the index is instead computed on the whole image, by applying a linear filter with a sliding kernel of predefined dimension. Then, the result is a matrix of the same size of the input: the desired final size is obtained by downsampling the image trough Gaussian pyramid in order to match the size of the desired scale.

The knowledge of the same kind of information at different scales allows neural networks to better understand depth and relative distances as described by Saxena et al. (2006). The scale can be varied by varying the segmentation window size or the number of subsequent downsamplings. Five scales, 2, 4, 8, 16, and 32 pixels (corresponding to a downsampling pyramid levels number from 1 to 5 for global indexes) have been initially selected: after some preliminary evaluations, scales 2 and 32 have been discarded. The former because of the greater computation required, the latter for too excessive loss of information.

Combining 4 quantities, 2 extraction methods and the 3 remaining scales, a total of 24 indexes are available. In a try and error process, they were tested in 12 different configurations. In this work, only the most effective is presented: the minimum RMS error obtained at the end of ANNs training phase, measured on the validation set (see Section 3), was adopted as effectiveness criterion. In the actual implementation all the 3 different scales,

called simply *small*, *medium* and *large* from here onward, respectively of $s_S = 4$, $s_M = 8$, $s_L = 16$ pixel, are included, leading to a final hazard map of 256×256 pixel in size. For a down-looking camera with a 60° angle of view, this corresponds to a resolution of 9.02 meters per pixel, for images taken from an altitude of 2000 m (representative of a HDA High Gate altitude), decreasing to a value of 1.80 meters per pixel for images taken from 400 m. This value is of the same order of magnitude of a realistic lander footprint, making the final hazard map resolution appropriate for an efficient landing site selection.

Mean μ and standard deviation σ are computed by segmentation at each of the considered scales. The two statistical indexes are defined as:

$$\mu = \frac{\sum_{i=1}^{N} I_i}{N}, \tag{2}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (I_i - \mu)^2}{N-1}}, \tag{3}$$

where I_i corresponds to the intensity of the i-th pixel, N is the number of pixels inside the considered image window. Image gradient (*Grad*) and Laplacian of Gaussian (*LoG*) are computed by global filtering and downsampling. *Grad* is approximated through an expanded 5×5 Prewitt kernel for both horizontal and vertical directions (Kekre and Gharge, 2010). 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 (Nixon and Aguado, 2008, Ch. 4.2). It combines a Gaussian smoothing with the Laplacian operator and its general formulation in continuous space is:

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

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

In addiction to these indexes, also *Sun inclination angle*, defined as the angle of the sun above the local horizon, is assembled in the input matrix.

This is necessary to make the neural network able to correctly distinguish between sharp and blunt shadows. Summarizing, the value of each hazard map pixel is computed from a vector of 13 component: μ, σ, Grad and LoG for each of the 3 considered scales, plus the Sun inclination angle. Each component is normalized before it is passed to the next stage. 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 Computation

After assembling, input is processed by an Artificial Neural Network. In this work, two alternative architectures are considered. A *Feedforward Multilayer Neural Network*, and a *Cascade Neural Network*.

Feedforward multilayer (shown in Figure 3) is the most simple type of ANN, widely used in function regression and pattern recognition (Hornik et al., 1990; Leshno et al., 1993). It consists in an input layer, one or more hidden layers, and an output layer. The input is processed by each layer sequentially; each layer is fully connected with the subsequent one. The network architecture (number of hidden layers and neurons for each layer) is a priori determined; the training process consists in an optimization (by error backpropagation) of the weights of each neuron. The main criticality of feedforward multilayer networks is the choice of a suitable architecture: too complex schemes make the system prone to overfitting phenomena and slow down the optimization process. In this work, in order to determine the minimal effective size of the net, progressively more complex architectures have been tested: the final scheme consists in a single hidden layer made up by 15 neurons. Addition of further neurons produces no significant performance improvement.

In cascade networks, each layer is made up by a single neuron: the input for each layer is the original input of the network, plus the output of each previous hidden layer (see Figure 4). In this structure, hidden neurons are part of the optimization: at the beginning of the training the net has only input and output layers, and hidden units are progressively added, leading to a near-optimal architecture (Fahlman and Lebiere, 1990).

The two architectures shares some characteristics. Hyperbolic tangent is used as hidden layer activation function, making the networks able to handle nonlinearities (Yonaba et al., 2010). Since the desired output is limited in the range $[0, 1]$, the logarithmic sigmoid is adopted as output function. At

the end of the computation, a very light blur filter is applied on the hazard map to relate nearby pixels.

2.4. Target Selection

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

- Minimum hazard level
- 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)

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:

$$\mathbf{r} = d_{\text{res}} s_{\text{s}} \mathbf{x}, \tag{5}$$

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 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 $z_{ij} \leq z_{\max}$ the *Size Score* $r_{\mathrm{CLS}ij}$ 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:

$$r_{\mathrm{CLS}ij} \geq r_{\min} = \frac{d_{\mathrm{foot}}}{2} + e_{\mathrm{gnc}}, \tag{6}$$

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 remaining CLS the *Diversion Score* $d_{\mathrm{CLS}ij}$ is computed as the distance from the Nominal Landing Site (NLS):

$$d_{\mathrm{CLS}ij} = \|\mathbf{r}_{\mathrm{CLS}ij} - \mathbf{r}_{\mathrm{NLS}}\|, \tag{7}$$

where $\mathbf{r}_{\mathrm{CLS}ij}$ is the metric position of the CLS at image coordinates, and $\mathbf{r}_{\mathrm{NLS}}$ is the metric position of the NLS.
5. The *Safety Score* $z_{\mathrm{CLS}ij}$ 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 $r_{\mathrm{CLS}ij}$.
6. The three scores are normalized to make their values to the same order of magnitude. Than, a global "landability" score $l_{\mathrm{CLS}ij}$ is obtained as:

$$l_{\mathrm{CLS}ij} = \mathbf{w}^T \begin{bmatrix} \tilde{r}_{\mathrm{CLS}ij} \\ 1 - \tilde{d}_{\mathrm{CLS}ij} \\ 1 - \tilde{z}_{\mathrm{CLS}ij} \end{bmatrix}, \tag{8}$$

where $\mathbf{w}^T$ is a vector of weights, introduced 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: additional data required to obtain the corresponding ground truth solution (in particular an affordable 3D model that can be associated with the images) are only available for a very limited number of sites. 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.
This latter approach has been selected for this work: a dataset of 98 artificial images has been generated and exploited for training and testing purposes. Images are taken from an altitude of 2000 m, a value inside the interval in which HDA system is required to operate (Lunghi et al., 2015), and they are divided into 3 subsets: training, validation and test. The training set (67 images) is exploited to directly optimize the networks weights with a back-propagation algorithm; network overfitting is avoided using an early stopping method: the training is stopped when the RMS error, evaluated on the validation set (23 images), reaches a minimum. The test set, made up of 8 images, is exploited to assess the system performances.

3.1. Artificial Images Generation

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 m/point, by adding fractal noise, small craters and boulders (Shankar et al., 2008). Craters deposition respects the statistical distribution observed on the real lunar surface, as well as the real craters formation process (Cross and Fisher, 1968), while craters morphology follows empirical morphometric relations obtained from lunar imagery (Hörz et al., 1991). Finally, the camera frame is rendered in *POV-Ray* [2] with the desired settings for camera, position, and attitude with realistic illumination conditions. A pinhole camera model, with a 60° angle of view has been adopted. Table 1 summarizes the assumed camera parameters.

[1]Courtesy of NASA and Arizona State University. http://wms.lroc.asu.edu/lroc/rdr_product_select, last visit on: April 7, 2016.

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

Table 1: Camera model parameters for artificial images renderings.

Model	Pinhole
Resolution	1024×1024 pixel
Angle	60°
Color	8 bit grayscale

3.2. Ground Truth Solution

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:

$$Z = aX + bY + c, \tag{9}$$

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:

$$S = \tan^{-1}\big(\sqrt{a^2 + b^2}\big). \tag{10}$$

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

$$R = \max\big(Z_i - (aX_i + bY_i + c)\big) - \min\big(Z_i - (aX_i + bY_i + c)\big), \tag{11}$$

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:

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

Shadow map can be obtained through the same rendering software adopted for the image generation: a rendering of the scene with the terrain model

textured in uniform white, with no reflections and no environmental light, produces a boolean map in which sunlit regions are perfectly white and shadows are perfectly black. Practically, the process has been exploited only for the first images to automatically identify an intensity threshold, in order to obtain shadow maps by simply thresholding the camera image histogram (with a consistent speedup of the training set generation process). 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 pixel frame) by applying a Gaussian pyramid. The downsampling process increases the hazard map smoothness, making easier the ANNs training process (ANNs are less effective in reproducing discontinuous functions). Figure 5a 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 Figure 5b. Based on this hazard map, is possible to compute the true safety and ranking of landing sites with the algorithm presented in the previous section, with $z_{\max} = 0.3$.

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. A footprint $d_{\text{foot}} = 3\,\text{m}$ and a navigation error $e_{\text{gnc}} = 15\,\text{m}$ (3σ) have been considered. The training ended for the two network with a very similar value of Mean Square Error (MSE) over the test set, equal respectively to 0.01940 (multylayer) and 0.02039 (cascade).

ANNs are not expected to exactly reproduce the original ground truth hazard map; instead an approximation of them is expected. In Figure 6 an example of hazard map computed by the two architectures, relative to the same input image of Figure 5, is shown. It is possible to see how in both cases 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. It can be seen how the cascade architecture tends to give sharper results, while multilayer tends to smooth the hazard index. For these reasons, the safety threshold $z_{\max}$ used in the actual system is not required to be equal to the value used in ground truth computation: on the contrary, its value should be tailored to the specific architecture. Once compared with the ground truth, a landing site can be classified as:

- *True Positive* (TP): a site correctly classified as safe;
- *False Positive* (FP): an unsafe site, erroneously considered safe;
- *False Negative* (FN): a safe site, seen as unsafe;
- *True Negative* (TN): a correctly recognized unsafe site.

It this then possible to define the *Safety Ratio* r_S as the ratio between the number of TP landing sites and the total number of sites classified as safe:

$$r_S = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{12}$$

while the *Correctness Ratio* r_C is the ratio between the number of the correctly found sites and the total number of true safe landing sites in the image:

$$r_C = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{13}$$

The probability to select an unsafe target is minimized by maximizing r_S, while the maximization of r_C increase the available landing area. The system performances can be expressed in a unique index J defined as:

$$J = r_S^5 \, r_C^{1/5}, \tag{14}$$

where the exponents 5 and 1/5 are introduced to give more relative importance to landing safety, which is the main driver in the performance assessment.

Values of $z_{\max}$ from 0.04 to 0.30 have been tested on the hazard maps computed by the architectures under exams on the test set images: obtained r_S and r_C are shown in Figure 7. A null ratio value mean that no landing site

has been found for the specific threshold value. Looking at the compound performance index J in Figure 8, it is easy to identify the best threshold values of 0.23 and 0.17 respectively for the multilayer and the cascade networks.

Once the threshold is selected, it is possible to compare the performances of the two architectures. In sites ranking, a weight vector $\mathbf{w}^T = [0.6\ 0.3\ 0.1]$ is adopted. The heavier weight (0.6) has been assigned to the size score r_{CLS}: a larger available area increase the robustness of the system with respect to navigation uncertainties. The intermediate weight corresponds to the diversion score d_{CLS}, in order to maximize the probability that a feasible trajectory is found by the guidance system, minimizing the requested diversion; the lightest weight is linked to the mean hazard index, being all the candidate targets already under the threshold $z_{\max}$, and then considered as safe. Following the obtained results are summarized:

- Always a True Positive is selected as first Target Landing site;
- The mean ranking of the first FP is 695 (cascade) and 460 (multilayer), allowing the system to always find a backup landing site if required by the guidance system (worst case on single image: 39 and 38, respectively).

Globally, the cascade architecture obtain a higher score principally due to a higher safety ratio (0.9649 with respect to 0.9430 obtained by multilayer architecture). Figures 9 and 10 show the landing site ranking, and the final selected target, for the original image of Figure 5, computed by the two architectures under test. It is possible to see how the sharper hazard map obtained by cascade networks allows to find a larger number of landing sites also with the application of a smaller threshold value.

4.1. Real Images

The system 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 ANNs generalization capabilities, 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. 11a, 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 maps (Figures 11b and 11c), the neural networks seem 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. The higher sharpness of cascade networks w.r.t. the smoother maps computed by multilayer architecture is clearly visible.

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. 12a. This 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 (Figures 12b and 12c), the system seems to have qualitatively understood hazard trends of the various areas: bright colors (safe) for the planar area apart from the irregularities, gray and black (unsafe) for the most of the rest. The very low albedo of this particular image is sometimes interpreted as shadow by the cascade system, which give more conservative results.

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 method. Results have been cross-checked in two independent ways: each subroutine execution has been measured with the high resolution clock of the standard C++ *chrono* library, while with the *GNU/time* command has been exploited to verify the overall computation. All tests have been performed on a AMD A10-7700K APU, running 64 bit Ubuntu 14.04 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 108 148 hits at 250 Hz, for a total time of 432.59 s, while the correspondent CPU time resulted 432.67 s. Taking into account the possible overhead that can affect measurements differently with the two methods, the values are comparable. Figure 13 shows the breakdown of the computation time over the different algorithm stages: the principal bottleneck is identified in indexes extraction, that requires more than the 49% or the total runtime. This result agrees with the expected: image processing algorithms, that constitute most of the task, are computationally expensive.

The sensitivity of the hazard map computation time from the actual network architecture is weak. ANNs are computationally efficient and the number of operations to be performed in this stage has the same order of magnitude in both feedforward multilayer and cascade neural network.

The computation time demanded by preprocessing, indexes extraction, and hazard map computation is practically constant, operating these subroutines on the whole image indifferently. It is not the same for target ranking and selection, since the number of sites classified as safe is determined by both the input image and the adopted threshold $z_{\max}$. To estimate the theoretical maximum required time, a profiling test with $z_{\max} = 1$ has been performed. In this condition, all the image is considered as a potential target and performances decrease drastically (this case is represented with a dashed line in Figure 13). Anyway this can be considered an unrealistic case, and it is used only to identify a computational weight upper boundary: as can be seen in the graph, the actual mean time measured during profiling test is much lower, and target selection resulted the fastest of the four algorithm's stages. All the algorithm bottlenecks are located in high parallel tasks: recent developments of dedicated space qualified hardware, based on high performance parallel units (such as Field Programmable Gate Arrays) allow to expect further improvements in real applications up to real-time performances, with a speedup greater than 100 times with respect a full software implementation (Capuano et al., 2012; Dunstan and Hornbostel, 2014).

4.3. Sensitivity to Uncertainties

The proposed system requires additional information about the altitude and the attitude of the spacecraft: in this section it is briefly discussed how uncertainties in this information could affect the proposed hazard detection system. The attitude measure is involved in the perspective correction during the preprocessing phase: in this case, the deformation of the scene is not

uniform on the image, but increasing with the inclination at which a certain region is seen (proportional to the tangent of the angle). Assuming a maximum allowed inclination from the vertical equal to 15° and a 60° FoV for the camera, the maximum inclination at which an area can appear in the input image is 45°. The expected attitude estimation accuracy is far better than 1°, relying on star trackers and inertial measurement units (Riedel et al., 2010). Taking a 1 degree error as a worst case scenario, the corresponding maximum relative error in lengths estimation would be 3.55 %.

The altitude value is required for the computation of the image resolution d_{res}: uncertainties in the altitude estimation introduces a scaling error of the entire scene, with a linear relationship. In real systems the expected accuracy in position (and then altitude) estimation is better than 1 %, using pure vision-based navigation. Conjugated with laser range measures, an accuracy in position better than 0.2 m is expected (Amzajerdian et al., 2013; Trawny et al., 2013): this is equivalent to an error less than 0.2 % from an altitude of 1000 m. The errors introduced by altitude measurement are then at least one order of magnitude lower with respect to other sources of uncertainty.

Also unknown orographic reliefs introduces errors in distances estimation: a certain level of distortion is introduced by the perspective correction, applied with the flat scene approximation, especially near the edges of the scene. A certain amount of non uniform scaling error is also introduced, due to the deviation from the scene mean plane that alters the actual distance of the terrain with respect to the camera. ANNs proved to be effective in correctly identifying the slopes, while including the size of the landing area in the target ranking criteria copes with scaling errors, privileging wider target. During the tests, all the points selected as target resulted several times larger than the minimum required. Terrain models with deviations from the mean plane up to ±10 % (with respect to the altitude) have been included in the training and test set.

5. Conclusion

In this paper, a new Hazard Detection and Avoidance 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 extraction of the most informative indexes from the original image reduces the dimension of the neural network input space, allowing a precise classification even with a simple network architec-

ture, maintaining in this way a light computational weight. 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. Two possible ANNs types have been tested; also if a safe landing site was selected in the 100% of the test cases, the cascade networks ability to autonomously get a suboptimal structure makes them more effective in false positives rejection than standard multilayer neural networks. More complex ANNs architectures are still possible, with additional margins of improvement.

A primary role in detection accuracy is played by the information extraction stage: a proper selection of the input makes the neural networks training process faster and more effective. In this field, several improvements are still possible, and some options are under study. Discrete wavelet transform is a promising technique, allowing an easy distinction between high an low frequency content of the image recursively at multiple scales; also other representations, such as independent component analysis and principal component analysis, are more efficient as information carriers with respect the quantities here considered. Anyway, most of these methods requires more computation: a trade-off between index complexity and needed number of indexes should be carried out to maximize the system performance, minimizing at the same time the overall computational cost of the system. Nevertheless, the selection of the indexes remains dependent on the user choice, and the verification of the effectiveness of a specific combination is a very slow process, requiring a complete training of the system to be properly estimated. In this context, the application of machine learning techniques appears attractive, to automatically obtain an optimal minimal representation suitable for hazard detection in place of the indexes vector. Autoencoders, convolutional neural network and self-organized maps appears the most promising options and are currently under study.

Finally, the next step in system validation will require a deeper testing. A wider test set will be considered, with more terrain types and variations in environmental condition. Plus, the robustness of the system with respect possible image quality degradation will be assessed.

References

Açikmeşe, B., Ploen, S. R., 2007. Convex programming approach to powered descent guidance for Mars landing. Journal of Guidance, Control, and Dynamics 30 (5), 1353–1366.

Amzajerdian, F., Pierrottet, D. F., Hines, G. D., Petway, L. B., Barnes, B. W., 2013. Doppler lidar sensor for precision navigation in GPS-deprived environment. Proc. SPIE 8731, 87310G–87310G–6.

Arvidson, R., Adams, D., Bonfiglio, G., Christensen, P., Cull, S., Golombek, M., Guinn, J., Guinness, E., Heet, T., Kirk, R., Knudson, A., Malin, M., Mellon, M., McEwen, A., Mushkin, A., Parker, T., Seelos, F., Seelos, K., Smith, P., Spencer, D., Stein, T., Tamppari, L., 2008. Mars Exploration Program 2007 Phoenix landing site selection and characteristics. Journal of Geophysical Research: Planets 113 (E3).

Betts, J. T., 1998. Survey of numerical methods for trajectory optimization. Journal of Guidance, Control, and Dynamics 21 (2), 193–207.

Blackmore, L., Açikmeşe, B., Scharf, D. P., 2010. Minimum-landing-error powered-descent guidance for Mars landing using convex optimization. Journal of Guidance, Control, and Dynamics 33 (4), 1161–1171.

Capuano, G., Severi, M., Della Sala, E., Ascolese, R., Facchinetti, C., Longo, F., Oct. 2012. Compact and high-performance equipment for vision-based navigation. In: 63rd International Astronautical Congress (IAC). Napoli, Italy.

Cross, C. A., Fisher, D. L., 1968. The computer simulation of lunar craters. Monthly Notices of the Royal Astronomical Society 139 (2), 261–272.

Daugman, J. G., Jul 1988. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. IEEE Trans. Acoust., Speech, Signal Process. 36 (7), 1169–1179.

Dunstan, M., Hornbostel, K., Jun. 2014. Image processing chip for relative navigation for lunar landing. In: 9th International ESA Conference on Guidance, Navigation, and Control Systems. Porto, Portugal.

Fahlman, S. E., Lebiere, C., 1990. The cascade-correlation learning architecture. In: Touretzky, D. S. (Ed.), Advances in Neural Information Processing Systems 2. Morgan-Kaufmann, San Francisco, CA, pp. 524–532.

Fahroo, F., Ross, I. M., 2002. Direct trajectory optimization by a Chebyshev pseudospectral method. Journal of Guidance, Control, and Dynamics 25 (1), 160–166.

Furfaro, R., Fink, W., Kargel, J. S., 2012. Autonomous real-time landing site selection for venus and titan using evolutionary fuzzy cognitive maps. Applied Soft Computing 12 (12), 3825–3839.

Golombek, M., Grant, J., Kipp, D., Vasavada, A., Kirk, R., Fergason, R., Bellutta, P., Calef, F., Larsen, K., Katayama, Y., Huertas, A., Beyer, R., Chen, A., Parker, T., Pollard, B., Lee, S., Sun, Y., Hoover, R., Sladek, H., Grotzinger, J., Welch, R., Noe Dobrea, E., Michalski, J., Watkins, M., 2012. Selection of the Mars Science Laboratory landing site. Space Science Reviews 170 (1), 641–737.

Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Networks 4 (2), 251–257.

Hornik, K., Stinchcombe, M. B., White, H. L., 1990. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural Networks 3 (5), 551–560.

Hörz, F., Grieve, R., Heiken, G., Spudis, P., Binder, A., 1991. Lunar Surface Processes. Cambridge University Press, New York, Ch. 4, pp. 61–120.

Hyvärinen, A., Karhunen, J., Oja, E., 2001. Independent Component Analysis. John Wiley & Sons, Inc., New York.

Kekre, H. B., Gharge, S. M., 2010. Image segmentation using extended edge operator for mammographic images. International Journal on Computer Science and Engineering 2 (4), 1086–1091.

Klumpp, A. R., 1974. Apollo lunar descent guidance. Automatica 10 (2), 133–146.

Leshno, M. M., Lin, V., Pinkus, A., Schocken, S., 1993. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Networks 6 (6), 861–867.

Lunghi, P., Lavagna, M., Oct. 2014. Autonomous vision-based hazard map generator for planetary landing phases. In: 65th International Astronautical Congress (IAC). Toronto, Ontario Canada.

Lunghi, P., Lavagna, M., Armellin, R., 2015. A semi-analytical guidance algorithm for autonomous landing. Advances in Space Research 55 (15), 2719–2738.

Masursky, H., Crabill, N. L., 1976. The Viking landing sites: Selection and certification. Science 193 (4255), 809–812.

Matthies, L., Huertas, A., Cheng, Y., Johnson, A., May 2007. Landing hazard detection with stereo vision and shadow analysis. In: AIAA Infotech@Aerospace Conference. Rohnert Park, CA, AIAA Paper 2007-2835.

Nixon, M. S., Aguado, A. S., 2008. Feature Extraction and Image Processing, 2nd Edition. Academic Press, London.

Parreira, B., Di Sotto, E., Caramagno, A., Rebordão, J., Jun. 2008. Hazard avoidance for planetary landing: GNC design and performance assessment. In: 7th International ESA Conference on Guidance, Navigation & Control Systems. Tralee, Ireland.

Parreira, B., Rogata, P., Di Sotto, E., Caramagno, A., Rebordao, J. M., Motrena, P., Mancuso, S., Aug. 2007. Consolidated performance assessment of hazard avoidance techniques for vision based landing. In: AIAA Guidance, Navigation, and Control Conference. Hilton Head, SC, AIAA Paper 2007-6854.

Pien, H., 1991. Autonomous hazard detection and avoidance for Mars exploration. In: 8th AIAA Computing in Aerospace Conference. Baltimora, MD, AIAA Paper 1991-3731.

Riedel, J., Vaughan, A., Werner, R. A., Tseng-Chan, W., Nolet, S., Myers, D., Mastrodemos, N., Lee, A., Grasso, C., Ely, T., Bayard, D., Aug. 2010. Optical navigation plan and strategy for the lunar lander Altair; OpNav for lunar and other crewed and robotic exploration applications. In: AIAA Guidance, Navigation, and Control Conference. Toronto, Ontario Canada, AIAA Paper 2010-7719.

Saxena, A., Chung, S. H., Ng, A. Y., Dec. 2006. Learning depth from single monocular images. In: Advances in Neural Information Processing Systems 18. MIT Press, Cambridge, MA, pp. 1161–1168.

Schölkopf, B., Smola, A., Müller, K.-R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10 (5), 1299–1319.

Shankar, U. J., S., W.-J., Criss, T. B., Adams, D., Mar. 2008. Lunar terrain surface modeling for the ALHAT program. In: IEEE Aerospace Conference. Big Sky, MT, pp. 1–10.

Spencer, D. A., Adams, D. S., Bonfiglio, E., Golombek, M., Arvidson, R., Seelos, K., 2009. Phoenix landing site hazard assessment and selection. Journal of Spacecraft and Rockets 46 (6), 1196–1201.

Trawny, N., Carson, J. M., Huertas, A., Luna, M. E., Robak, V. E., Johnson, A. E., Martin, K. E., Villalpando, C. Y., 2013. Helicopter flight testing of a real-time hazard detection system for safe lunar landing. In: AIAA SPACE Conference and Exposition. San Diego, CA, AIAA Paper 2013-5313.

Ulamec, S., Biele, J., Blazquez, A., Cozzoni, B., Delmas, C., Fantinati, C., Gaudon, P., Geurts, K., Jurado, E., Küchemann, O., Lommatsch, V., Maibaum, M., Sierks, H., Witte, L., 2015. Rosetta lander – Philae: Landing preparations. Acta Astronautica 107, 79–86.

Wibben, D. R., Furfaro, R., 2016. Optimal sliding guidance algorithm for Mars powered descent phase. Advances in Space Research 57 (4), 948–961.

Woicke, S., Mooij, E., Jan. 2014. Stereo-vision algorithm for hazard detection during planetary landings. In: AIAA Guidance, Navigation, and Control Conference. National Harbour, MD, AIAA Paper 2014-0272.

Wong, E., Masciarelli, J., Singh, G., Aug. 2002. Autonomous guidance and control design for hazard avoidance and safe landing on Mars. In: AIAA Atmospheric Flight Mechanics Conference. Monterey, CA, AIAA Paper 2002-4619.

Yonaba, H., Anctil, F., Fortin, V., 2010. Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting. Journal of Hydrologic Engineering 15 (4), 275–283.

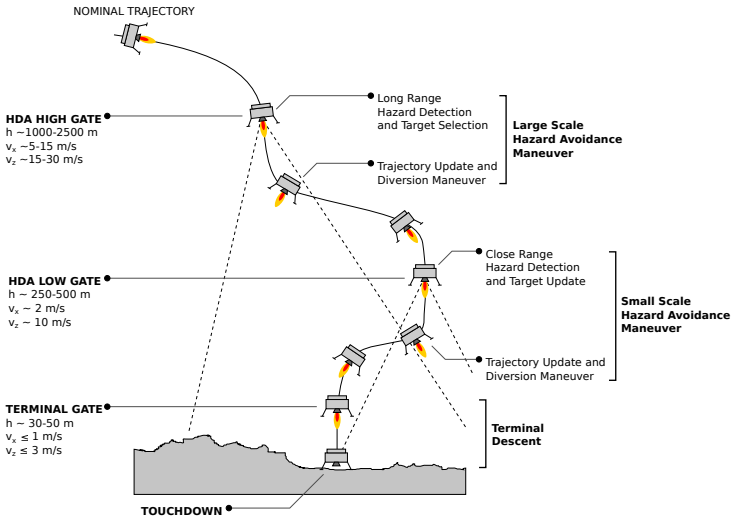


Figure 1: Nominal Hazard Detection and Avoidance maneuver.

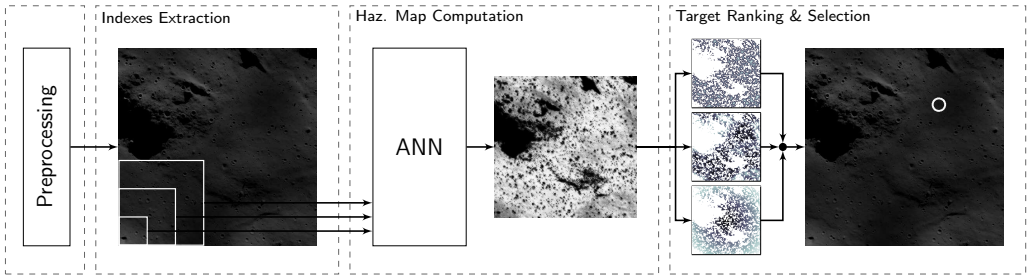


Figure 2: Hazard Detection system logical scheme. An artificial neural network estimates hazard index value from elementary information extracted from the image at different scales. Different ranking criteria are fused together to select the best target.

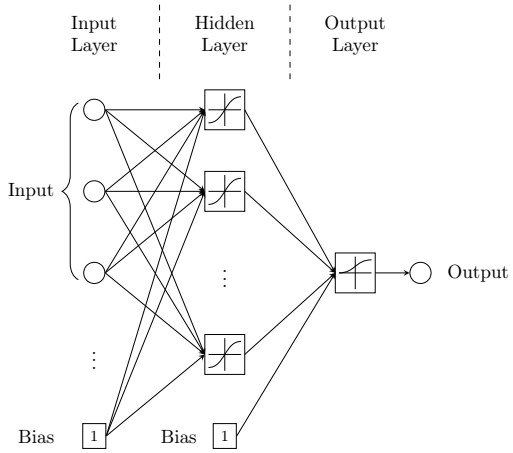


Figure 3: Multilayer feed-forward network structure. The network structure is predetermined and only neuron weights are trained.

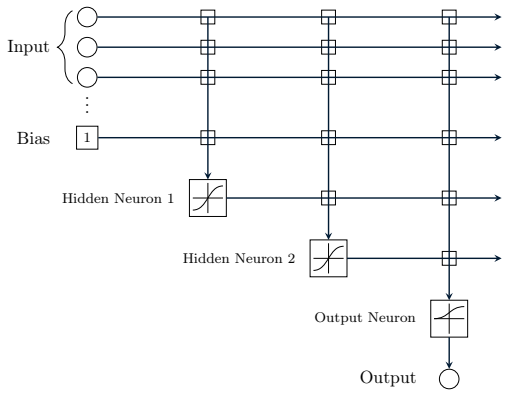
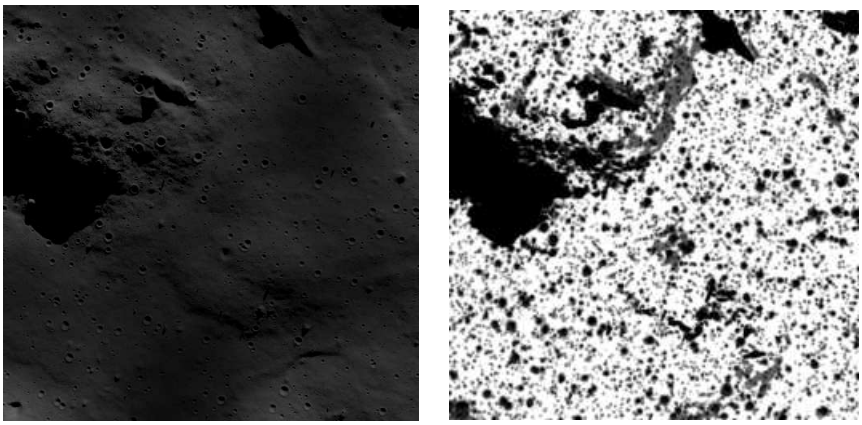
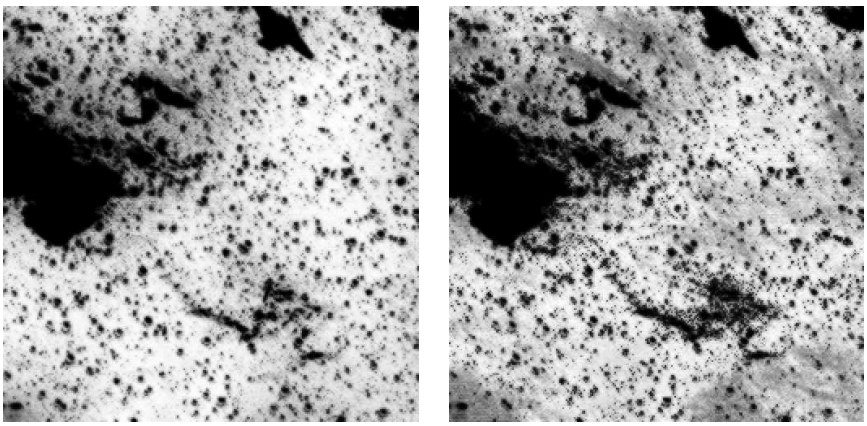


Figure 4: Cascade network structure. Hidden neurons are progressively added during training. Each white square represents a weight.



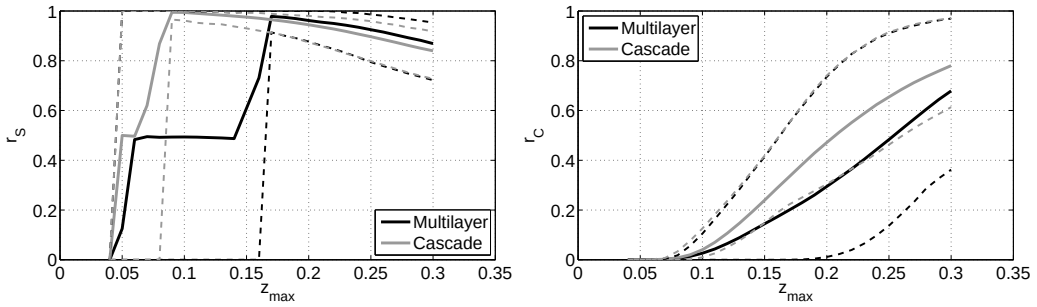
(a) Original image. (b) Ground truth hazard map.

Figure 5: Ground truth solution computation. Artificial image, Lowell crater floor DEM. Original image and ground truth map. Safe areas are depicted in white ($z = 0$), while black represents shadowed regions ($z = 1$). Intermediate gray levels represents unsafe regions failing one single or both safety tests (roughness and slope).



(a) Multilayer network. (b) Cascade Network.

Figure 6: Computed hazard map. Artificial image, Lowell crater floor DEM. Safe areas are bright (low z).



(a) Safety Ratio. (b) Completeness Ratio.

Figure 7: Architectures comparison, $z_{\max}$ threshold tuning. Safety and Completeness ratios define the overall system performance. Solid lines are mean values, dashed lines are lower and upper boundaries.

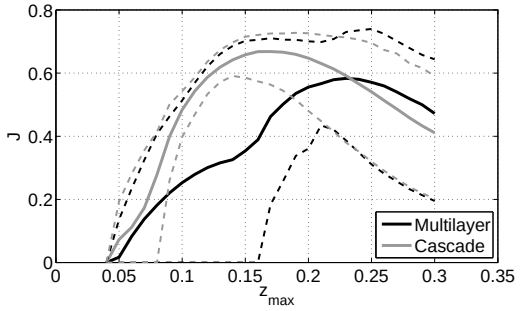
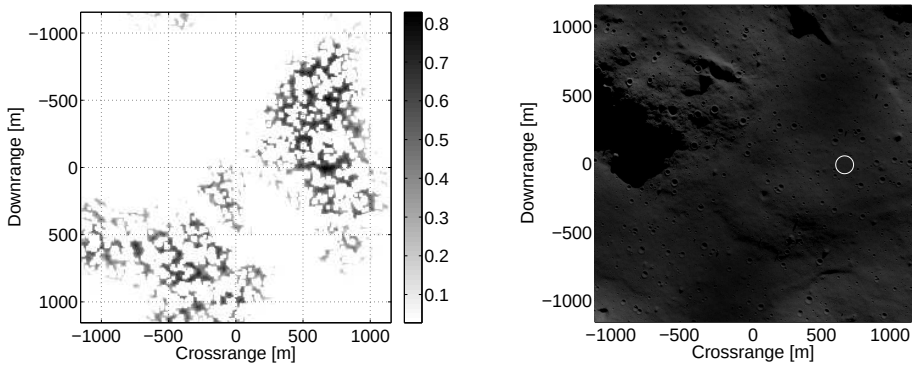
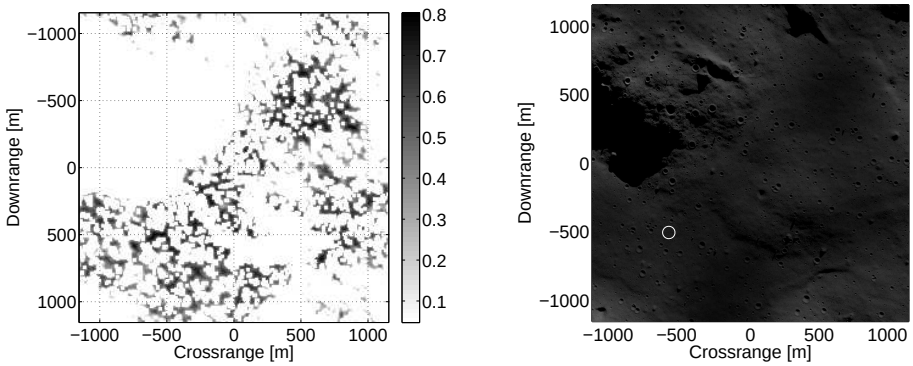


Figure 8: $z_{\max}$ threshold tuning. Overall performance index (higher is better). Solid lines are mean values, dashed lines are lower and upper boundaries.



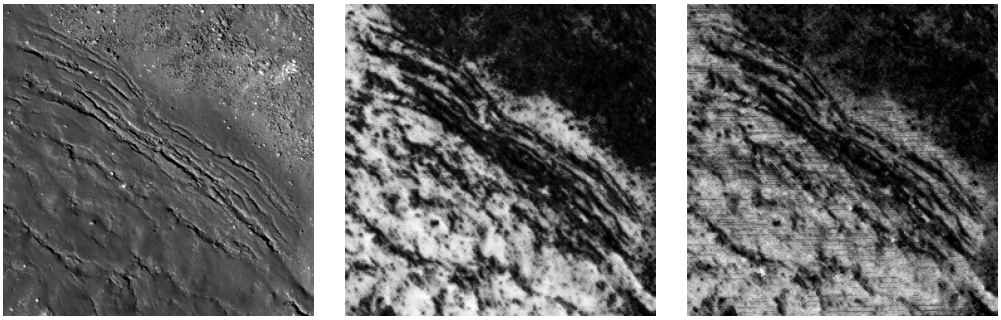
(a) Landing sites ranking. (b) Selected target.

Figure 9: Target ranking and selection. Multilayer ANN ($z_{\max} = 0.23$).



(a) Landing sites ranking. (b) Selected target.

Figure 10: Target ranking and selection. Cascade ANN ($z_{\max} = 0.17$).



(a) Original frame. (b) Haz. map, multilayer network. (c) Haz. map, cascade network.

Figure 11: Lunar real surface image application, Larmor Q crater floor, NAC frame M151726155R, courtesy of NASA/GSFC/ASU.

(a) Original frame. (b) Haz. map, multilayer network. (c) Haz. map, cascade network.

Figure 12: Comet 67P Churyumov-Gerasimenko, Imhotep region (Photo: ESA).

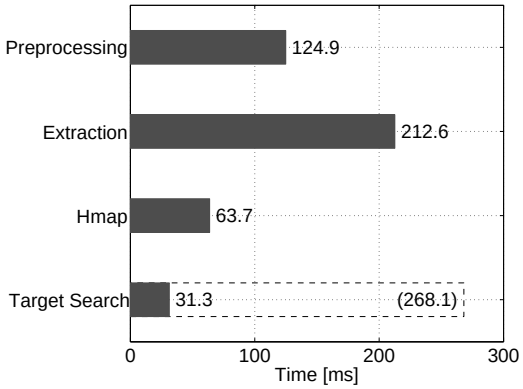


Figure 13: Computation time breakdown. It is easy to see the bottleneck of the indexes extraction stage. Theoretical maximum time required by target search is shown in dashed line.