Design and development of a stereo vision-based navigation and guidance system for a space rover

M. Carpentiero

Università di Roma, La Sapienza, Rome, Italy, 00184

The lag in telecommunications between a planetary exploration rover and the Earth does not allow the real-time control of the platform. Autonomous guidance and navigation capabilities are therefore mandatory requirements for planetary exploration missions. This paper deals with the definition, design, assembling and testing of a low-cost guidance and navigation system intended to be into the rover RAGNO (Rover for Autonomous Ground Navigation and Observation), an autonomous platform developed at Sapienza Università di Roma to familiarize with vehicles operating in hazardous or poorly known environment. The navigation function, while possibly completed by magnetometers and additional sensors, is mainly based on visual techniques, as they can reliably offer the required accuracy with suitable cost and complexity. For the purpose of obstacles identification and localization, basic to rovers operating in unknown environment, a stereoscopic navigation system has been selected. The observation from two points of view allows to localize a detected object by means of a simple triangulation of corresponding point pairs (features). The basic relations and the techniques implemented to detect and to match the features of the two images are briefly explained. As a result, the rover is able to autonomously understand the surrounding scenario. Once the obstacles have been localized, a safe and optimal path to reach the desired targets must be planned. To this aim, the implementation of a guidance algorithm based on manifold cell discretization, graph theory and length optimization has been presented. This algorithm, a customization of the A-star search method, is compared to a more classical one, based on the Lyapunov approach with the introduction of artificial potential functions centered in the identified obstacles.

Nomenclature

\[ P = (X, Y, Z) \] = point coordinates in the reference frame intrinsic to the scene and centered on the left camera
\[ P_L = (u_L, v_L) \] = projection of \( P \) in the left image reference frame \((u, v)\), with the origin at the top left corner
\[ P_R = (u_R, v_R) \] = projection of \( P \) in the right image reference frame \((u, v)\), with the origin at the top left corner
\[ p = (p_u, p_v) \] = coordinates of the optical centre O in the image reference frame
\[ P_h = (X, Y, Z, 1)^T \] = homogeneous coordinates of the scene point \( P \)
\[ P_i = (u, v, 1)^T \] = homogeneous coordinates of the image point \( P_L \) or \( P_R \)
\[ C_{RL} \] = center of projection of right/left camera
\[ f \] = focal length (in pixel)
\[ b \] = baseline between the two cameras
\[ Z \] = optical depth
\[ R_{LR} \] = rotation matrix from the right image reference frame to the left one

I. Introduction

Planetary exploration through mobile laboratory platforms has always been one of the most important topic of space engineering. The first examples are the American lunar vehicles and the Russian remotely control Lunokhod 1-2. Since the early 90s the space agencies focus on the exploration of Mars. NASA was the first to land a platform on the Red planet with the Sojourner (1996, Mars Path Finder). Till today other three rovers have landed on Mars: Spirit and Opportunity (2004, Mars Exploration Rover) and Curiosity (2011, Mars Science Laboratory). These platforms are the unique examples of autonomous mobile robots designed for planetary exploration. Autonomy has

1 Ph.D student, Department of Astronautical, Electrical and Energy Engineering. marco.carpentiero@uniroma1.it

American Institute of Aeronautics and Astronautics
been accomplished by integrating a panoramic binocular vision system (NavCams). NavCams\textsuperscript{1} software is able to identify and localize close obstacles by processing the acquired stereo image. This navigation system has been improved in Curiosity, where other four stereo pairs (Hazard avoidance cams) has been added in order to obtain a wider vision of the surrounding environment. Future Mars platforms like NASA’s Mars 2020 and ESA’s ExoMars 2018 are going to integrate a further improvement of the stereo-based navigation system. The reasons of the great success of the stereoscopic vision are multiple. At first, stereo vision is an attractive technology for rover navigation because is passive, i.e. sunlight provides all the energy needed for daylight operations. The second reason is that only a small amount of power is required as for the cameras electronics as to obtain knowledge about the environment. In addition, by installing enough cameras or cameras with a wide field of view, no moving parts are required to capture the 360° surrounding environment. Having fewer motors reduces the number of components that could fail. Thanks to the constant improvements in computer vision techniques, hardware performances and optical sensors quality, future space missions are likely to exploit stereo-based systems for applications aside from the rover navigation. Typical examples are the proximity maneuvers, like rendez-vous\textsuperscript{2,3,12,13} or docking, but also the more general in-orbit servicing like autonomous robotics arm grasping and manipulation of objects with unknown shapes.

This paper deals with the application of the stereoscopic vision to rover navigation in order to make the platform totally autonomous. The design and development phases are explained in details together with the results obtained during a test session. In particular, Section II introduces the rover RAGNO used in his work: its starting and current state of the art is described in details. Section III deals with the basic concepts of the stereovision and how this concepts are translated in software techniques in order to identify and localize the closest objects of the scene. Section IV is focused on the guidance strategy needed to plan a safe path leading the rover toward the final goal position. Section V shows the results obtained during the test campaign. The last Section VI summarizes the main points of this work and suggests future applications of both autonomous vehicles and stereovision.

II. The rover RAGNO

RAGNO\textsuperscript{4}, standing for Rover for Autonomous Ground Observation and Navigation, is a four wheels small rover originally designed in 2011 at the Guidance and Navigation Lab, Sapienza Università di Roma in order to familiarize with robotics platforms, multibody systems and remote control. RAGNO can be defined as a modular vehicle because different subsystems, for example a robotic arm, can be allocated onboard according to the mission needs. In the present project RAGNO has been equipped with a stereo viewer and a sensors platform containing a triaxial gyroscope and a triaxial magnetometer. This hardware integration allows to convert the rover from a remote-control system (Fig 1a) to an autonomous one (Fig 1b). The reading of sensors measures and the control of the angular velocity of the four wheels are realized by two Arduino shields. All the data are then sent to the guidance and navigation Matlab software through a serial communication.

![Figure 1. Initial (a) and final (b) RAGNO state of the art](image)

III. The stereoscopic navigation system

The stereo-based navigation system, based on two commercial webcams, is devoted to detect and localize the closest obstacles in the scene. The acquisition and processing of stereo images are managed by a Matlab\textsuperscript{®} routine running on the onboard computer. The acquisition process requires the use of a calibrated viewer while the processing algorithm takes into account the theory of the stereovision, also known as the epipolar geometry. Possible radial and tangential distortions have not been considered in this work.

A. Stereoscopic vision concept

An ideal stereoscopic viewer is composed of two identical cameras with parallel optical axes (Figure 2). This set-up allows to observe the scene from two points of view and to compute the optical depth of a detected point of the

American Institute of Aeronautics and Astronautics
scene (P) by means of a simple triangulation between P and its projections on the right (P_R) and the left camera (P_L) image planes. P_L and P_R - which identification in the images and association with P is for now assumed - are also defined as corresponding points.

Figure 2. Stereoscopic vision system arrangement (a) and pinhole camera with related similar triangles (b)

Once selected a reference frame rigid to the left camera, it is possible to express the triangulation relations considering both cameras like pinholes. For a pinhole camera (Fig 2b) the relation between a point P belonging to the scene and the relevant image point P_L can be found through the similarity criterion for triangles, stating that two triangles are similar if they have two proportional sides and the angle between these sides equal. Applying this similarity criterion to the \( \triangle \) (CKH)-\( \triangle \) (COH') and \( \triangle \) (CHP)-\( \triangle \) (CH'P') pairs of triangles it is possible to write:

\[
\begin{align*}
u &= f \frac{X}{Z} + p_u \\
v &= f \frac{Y}{Z} + p_v
\end{align*}
\]

(1)

Eq. (1) can be rewritten in matrix form by means of homogeneous coordinates:

\[
Z P_L = Z P = K P = \begin{bmatrix} f & 0 & p_u \\ 0 & f & p_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}
\]

(2)

Eq (2) is also labelled perspective law because it projects the point P from the reference frame characteristic to the scenario to the image reference frame. The \( K \) matrix is defined as the intrinsic matrix of the camera and its elements can be determined through a calibration procedure. The perspective law includes two equations with three unknowns (X, Y, Z), thus resulting unsolvable. This is the reason why it needs a second camera to obtain the optical depth of the point P. The application of the perspective law to the stereo viewer leads to the following relations:

\[
\begin{align*}
Z P_L &= K_L P \\
Z P_R &= K_R \begin{bmatrix} 1 \\ R^T \\ 0 \end{bmatrix} (P - C_L^R)
\end{align*}
\]

(3)

where \( R \) and \( C \) matrices take into account the relative position and orientation of the second camera. In the ideal case sketched in Figure 2a, with the baseline between the cameras orthogonal to the optical axes, which are parallel, and a similar arrangement for image planes and resolution, it follows that the intrinsic matrices are equal and that the rotation matrix corresponds to the identity.

Indeed Eq. (3) becomes:

\[
\begin{align*}
Z P_L &= K L P \\
Z P_R &= K R (P - C_L^R)
\end{align*}
\]

(4)

to be expanded as:

3

American Institute of Aeronautics and Astronautics
\[
\begin{align*}
\begin{cases}
    u_l &= f \frac{X}{Z} + p_u \\
    v_l &= f \frac{Y}{Z} + p_v \\
    u_R &= f \frac{X-b}{Z} + p_u \\
    v_R &= f \frac{Y}{Z} + p_v
\end{cases}
\end{align*}
\] (5)

Since cameras are supposed identical, the v-equations of (5) can be neglected. By subtracting the first and the third it is finally possible to obtain the third dimension, i.e. the optical depth Z:

\[
Z = \frac{f b}{u_l - u_R} = \frac{f b}{d}
\] (6)

which evaluation depends on the focal length, on the relative geometry between the cameras and on the parameter \( d \) called pixel disparity: the closer the scene point \( P \), the higher the disparity. Figure 3b shows an example of pixel disparity (disparity map) for the sample scenario depicted in Fig 3a, where there are two objects at different distances from the stereo viewer.

Figure 3. Sample of a stereo image (a) and the associated disparity map (b)

According to Eq (6), it can be seen how the foreground box is characterized by a greater pixel disparity than the background objects. The raw disparity map shown in Fig 3b does not allow to detect the obstacles easily. In fact, there are many sparse color spots, for example those corresponding to the floor, which do not identify an existing obstacle. Therefore, a cleaning algorithm should be implemented to filter out this noise and to identify the true obstacles. This process could require a high computational time because the disparity map should be analyzed pixel by pixel. So, another approach – based on the object recognition techniques - has been chosen in this work in order to save the computational time.

B. The software architecture of the stereo-based navigation system

The architecture of the navigation software developed for obstacles detection and localization consists of three blocks:
- stereo image acquisition
- stereo image processing
- obstacles detection and localization in the \((X, Z)\) motion plane

The resolution of the Eq. (6) requires the knowledge of the viewer’s baseline and of the cameras focal length. The former is a geometric parameter and it can be chosen by the user while the latter is an intrinsic characteristic and it must be determined through a calibration procedure. The Zhang’s method\(^{6}\) has been adopted for the calibration of the stereo-viewer. This method allows to determine the intrinsic parameters of both cameras and their relative orientation by solving the perspective law (2) for a set of points called markers. These markers belong to a predefined 2D calibration object and their \((X, Y, Z)\) coordinates are known. The most used calibration object is a rectangular chessboard with squared cells of known dimensions (Fig 4). The markers are identified in the vertices of each cell. By writing (2) in a reference frame rigid to the chessboard, the cameras orientation \((\mathbf{R})\) and position \((\mathbf{t})\) with respect to this frame must be taken into account:

Figure 4. Calibration object
\[ P_t = \lambda K R^T (P - t) = \lambda K \begin{bmatrix} R^T & -R^T t \end{bmatrix} \begin{bmatrix} P \\ 1 \end{bmatrix} = \lambda K \begin{bmatrix} R^T & -R^T t \end{bmatrix} P_h = H P_h \quad (7) \]

The \( H \) is a rectangular matrix (3x4) and is called homography. It is possible to make \( H \) square by taking into account that markers have the same Z coordinate and so it can be set to zero without loss of generality. In this way, the Eq. (7) reduces to the following:

\[
\begin{bmatrix}
0 \\
1
\end{bmatrix}
= \begin{bmatrix}
u \\
x \\
y
\end{bmatrix} = \lambda K \begin{bmatrix} r_1 \ r_2 \ r_3 \ t_1 \end{bmatrix} \begin{bmatrix}
0 \\
X \\
Y
\end{bmatrix} = \begin{bmatrix} h_1 \ h_2 \ h_3 \end{bmatrix} \begin{bmatrix}
Y
\end{bmatrix} = H P_h \quad (8)
\]

The Zhang's method allows to determine the stereo parameters by calculating the homography \( H \) through the knowledge of a set of \((P_L, P_R)\) points pairs. In order to obtain an accurate calibration of the three axes of each camera it needs to acquire chessboard’s images from different points of view. The results of Zhang’s calibration procedure are the following:

\[
K_{\text{L}} = \begin{bmatrix}
605.9893\pm1.1312 & 0 & 331.9116\pm0.5466 \\
0 & 605.2356\pm1.1355 & 238.8710\pm0.6377 \\
0 & 0 & 1
\end{bmatrix} \text{pixel}
\]

\[
K_{\text{R}} = \begin{bmatrix}
603.0425\pm1.0914 & 0 & 316.9373\pm0.7559 \\
0 & 602.4318\pm1.0880 & 231.9409\pm0.5975 \\
0 & 0 & 1
\end{bmatrix} \text{pixel}
\]

\[
R_{\text{L}} = \text{R}_{\text{R}} = \begin{bmatrix}
0.9992 & 0.0066 & -0.0397 \\
-0.0066 & 1 & 0.0015 \\
0.0397 & -0.0015 & 0.9992
\end{bmatrix}
\]

\[
t = \begin{bmatrix}
148.5633 \\
4.7982 \\
4.6790
\end{bmatrix} \text{mm}
\]

The above results show that the principal points (i.e. the center of projection) of the webcams do not correspond with the center of the respective CCD sensor. The values of the rotation matrix and the translation vector are almost similar to the expected ones. In fact the stereo viewer has been built with a baseline of 15 cm while the two cameras are aligned along the X axis.

1. **Stereo image acquisition**

The built in Matlab® USB webcam package allows the user to define the webcam object and to set its properties like the image resolution and the value of the focus. For the stereo viewer used, the standard resolution 640x480 and the autofocus mode have been chosen. The acquisition of the stereo image generates two 3D matrices which dimensions are 480x640x3. The third dimension refers to the number of colors used to define the single image i.e. the RGB three-color. Each element of the image matrix is defined as an 8-bit integer so it assumes a value in the [0-255] range. As the processing phase requires a greyscale image, the following conversion is needed:

\[ I = 0.2989 R + 0.5870 G + 0.1140 B \quad (9) \]

2. **Stereo image processing**

The resolution of the triangulation problem (6) assumes a previous knowledge of the corresponding points’ coordinates \((P_L, P_R)\). The identification and matching of these points’ pairs represents the main and most computationally expensive step of the whole navigation algorithm, since it needs a significant stereo image processing. In recent years, thanks to the constant improvement in computing power and software skills, several matching techniques have been developed with the goal to reduce the required computational time and to improve the robustness of the algorithms.

The typical process can be divided into two different phases:
research, extraction and characterization of particular points, called features, in both images
• comparison of left and right extracted features and matching of the corresponding pairs

A feature can be defined as a small region of the image characterized by one or more repeatable properties. There are different types of features depending on the researched properties:
• **edge**: it is a point of the image characterized by a discontinuity in the brightness gradient. This feature allows to identify the edges of a scene’s object
• **corner**: it is a point identified by the intersection of two edges where the gradient has a significant curvature. This feature allows to identify the corner of a scene’s object
• **blob**: it is a region of the image characterized by peculiar properties (e.g. brightness) with respect to the surrounding area.

Every kind of feature can be extracted by means of different detection algorithms, while each extracted point needs to be coupled to a descriptor vector with these properties:
• **distinguishability**: the descriptor has to make the generic feature distinguishable from the others. The level of distinguishability depends on the dimension of the descriptor
• **robustness**: the descriptor must be detectable in any lighting conditions and regardless of photometric distortions or noise

The choice of the descriptor depends on the goal of the image processing. For the purpose of obstacles identification, it needs a combination of detector-descriptor which does not require high computational time. After an accurate analysis of the existent extraction algorithms, the SURF (Speeded Up Robust Feature) and BRISK (Binary ROBUST Scale Invariant) detectors have been chosen.

SURF is a blob detector with an associated descriptor of 64 elements. The detection algorithm is a speeded-up version of the classic SIFT detector. The generic feature is extracted by analyzing the image in the scale space and by applying sequential box filters of variable dimension. The features are then localized in the point of the image where the determinant of the Hessian matrix $H(x,y)$ is maximum. The descriptor vector is then determined by calculating the Haar wavelet response in a sampling pattern centered in the feature.

BRISK is a corner detector with an associated binary descriptor of 512 bit. The generic feature is identified as the brightest point in a sampling circular area of $N$ pixels while the descriptor vector is calculated by computing the brightness gradient of each of the $N(N-1)/2$ pairs of sampling points.

Once left and right features have been extracted, their descriptors are compared in order to determine the corresponding points pairs. The matching criterion consists in seeking for the two descriptors for which their relative distance is minimum. This distance corresponds to the Euclidean norm for SURF case and to the Hamming distance for the BRISK one. The latter distance is intended as the minimum number of substitutions needed to make two binary strings equal. The matching process seems to require a very high computational time because each left feature should be compared with all right ones. This may be true for object recognition processes where the algorithm has to identified a particular object by comparing its acquired image with a database of images. Instead, in the case of stereovision applied for obstacles detection, the computational time can be reduced by taking into account the theory of the epipolar geometry. In fact, it states that there exists a geometric constrain between the left and right projection of the scene point $P$. As a consequence, the space where a matching feature has to be researched reduces to a portion of the image.

**Figure 5.** Epipolar geometry for real and ideal stereo viewer. **In the ideal setup the epipolar point is at infinity because the sheaf of epipolar lines is non-regular. Lines are in fact parallel.**

As shown in Fig 5, for a fixed $P_l$, the corresponding $P_r$ lies on the line $l$, called epipolar, whose inclination depends on the geometry of the viewer. The position of $P_r$ varies depending on the depth of the scene point $P$. In the case of a
real stereo viewer, the epipolar line is delimited by the *epipole point* $e_1$ and the *vanishing point* $V$. The first is the center of the regular sheaf of epipolar lines and corresponds to the projection of $C_L$ on the right image plane while the second is the projection of a scene point $P$ with infinite depth. For the ideal stereo viewer used, the epipolar lines are horizontal therefore two matching features will have roughly the same $v$-coordinate. The searching process can be limited to a subset of right image rows thus obtaining a time savings.

Figure 7 and Figure 8 show the results of the processing phase for the sample space-like scenario in Figure 6. The irregular shape together with the color non-homogeneity of the close object entail that more SURF features than BRISK have been detected. By comparing the results obtained with features extraction to those of disparity map, it can be note that the former do not require a post processing noise filtering because there are few outliers which can be cut out by mean of a selection operation. On the contrary, the obtained disparity map shows the obstacle shape very well but a lot of noisy values too. A post processing filtering is therefore mandatory in order to make the map easily exploitable.

![Sampled stereo image (a) and disparity map (b)](image)

*Figure 6. Sampled stereo image (a) and disparity map (b)*

![Left and right extracted (a) and matched (b) SURF features](image)

*Figure 7. Left and right extracted (a) and matched (b) SURF features*

![Left and right extracted (a) and matched (b) BRISK features](image)

*Figure 8. Left and right extracted (a) and matched (b) BRISK features*

Regarding the computational effort, the results are summarized in Table 1 and Table 2. The onboard computer is a quadcore PC with a 2.4 GHz CPU and 8 GB RAM.

<table>
<thead>
<tr>
<th></th>
<th>Extracted (L/R)</th>
<th>Matched</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRISK</td>
<td>236/200</td>
<td>65</td>
</tr>
<tr>
<td>SURF</td>
<td>526/496</td>
<td>216</td>
</tr>
</tbody>
</table>

*Table 1. Extracted and matched features*

<table>
<thead>
<tr>
<th></th>
<th>Detection</th>
<th>Matching</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRISK</td>
<td>238</td>
<td>79</td>
<td>317</td>
</tr>
<tr>
<td>SURF</td>
<td>141</td>
<td>79</td>
<td>220</td>
</tr>
</tbody>
</table>

*Table 2. Computational time of detection and matching process (in ms)*

American Institute of Aeronautics and Astronautics
C. Obstacles detection and localization

Once the viewer has been calibrated and the correspondences have been found, the triangulation equation Eq. (6) can be solved. As the mission is supposed in a 2D space, the triangulation results are reported in the (X,Z) rover reference frame. This frame has the same axes of the camera reference (Fig 2a) but it is applied in the vehicle center of gravity. In order to select only the most interesting points among the set of triangulated ones, a threshold value is imposed on the optical depth Z. All the points with $Z > 2m$ have been cut out thus saving only the foreground features. The criterion adopted for obstacles identification is based on the evaluation of the features density. The (X,Z) plane is divided into square cells for each of which the density of contained features is calculated. A threshold value $d_{th}$ is then established. All the cells with a density greater than $d_{th}$ are marked as occupied by an obstacle. The threshold value depends on the discretization step: the higher it is and the lower the threshold has to be. Figure 9 shows the result of selection and identification process: background features have been cut out together with the low density ones.

Figure 9. Obstacle localization in 50cm step discretized (X,Z) plane (a) and depth marks in left image (b).

IV. The guidance system

The output of the navigation system, i.e. the coordinates of the localized obstacles, becomes one of the inputs of the guidance system. The purpose of the designed guidance algorithm is to plan a safe path that brings the rover from its starting position toward the target. Different guidance strategies can be found in literature. One of the most used solutions in space-related problems like rendez-vous, descent and landing is based on the Lyapunov’s stability theory. This approach provides to define a custom function, the artificial potential $V$, which describes the interaction between the body and the surrounding environment. The motion of the RAGNO platform can be compared to a sample positive charged particle in an electric field generated by other particles: the sample particle is attracted by the negative (the target) and repulsed by the positives (the obstacles). In the electric analogy, the total action exerted by the charges is the Coulomb’s force and it corresponds to the gradient of the potential function $V$. For the purpose of rover path planning, as the problem is kinematic, the following potential function of the platform velocity has been elaborated:

$$V = \begin{cases} V_{\text{att}} + V_{\text{wp}} = \frac{1}{2} \varepsilon \rho_i^2 + \frac{\#\text{obs}}{2} \sum_{j=1}^{\#\text{obs}} \frac{1}{\rho_{ij}^2} 
\left( \frac{1}{\rho_{ij}} \cdot \frac{1}{\rho_0} \right)^2 \rho_i^2 & \rightarrow \rho_{ij} \leq \rho_0 \\
V_{\text{att}} = \frac{1}{2} \varepsilon \rho_i^2 & \rightarrow \rho_{ij} > \rho_0
\end{cases}$$

(10)

The potential is a non linear function of rover distance from the obstacles ($\rho_{ij}$) and the target ($\rho_0$), and has a global minimum at the goal position. The j-th repulsive action vanishes when rover-obstacle distance is larger than the $\rho_0$ threshold; $\varepsilon$ and $\eta$ parameters can be modified in order to tune the magnitude of the two components. The integration of the gradient $\nabla V$ allows to obtain the safe trajectory that the rover has to follow.

Figure 10a shows the results of a simulation of a simple scenario. The red obstacle is included in the safety blue area while the three curves refer to different values of $\eta$ and $\varepsilon$. The legend reports the computational time needed for each simulation. It can be noted that the greater is $\eta/\varepsilon$, the stronger is the repulsive component of potential $V$, so that resulting trajectory is longer and safer.
This guidance strategy allows the user to select the preferred function $V$, while necessarily including the drawback if possible local minima ending up as traps. Indeed, depending on the position of the obstacles, the integration of the gradient function can generate an oscillating solution around wrong minima points (an example is shown in Fig 10b).

An alternative solution to the potential guidance has been proposed in the frame of this project\textsuperscript{11}. It is a customized version of the A-star search algorithm that is widely used in path finding process. Once $(X,Z)$ plane has been discretized in square cells, the path can be seen as sequence of straight lines that connect the starting to the goal position passing through the centers (nodes) of selected and adjacent cells. The line that links two adjacent nodes identifies a graph. The generic displacement is determined by minimizing the following cost function:

$$
J = d_N(n_0,n_{opt}) + d_E(n_{opt},n_{tgt})
$$

(11)

where $d_N(n_0, n_{opt})$ denotes the nodal distance between the starting node $n_0$ and the candidate optimal node $n_{opt}$ while $d_E(n_{opt}, n_{tgt})$ is the Euclidean distance between the candidate optimal node and the target one $n_{tgt}$. The algorithm defines two lists of nodes. The first one, named OPEN, is the list within which the optimal node is sought, including all the nodes reachable by the current candidate. Once the optimal node has been extracted by minimizing Eq (11), the list is not deleted but it is updated with the adjacent nodes of the next candidate. The extracted node is deleted by the OPEN and it is inserted into the second list, called CLOSED. The algorithm ends when the target node appears in the OPEN list. The constant update of this list implies that more than a path is assembled during the optimization process. Figure 11a shows this concept clearly. As the target (green) is aligned with the rover starting cell (cyan), the shortest path initially identified by the algorithm is the straight line between rover and target (light blue). After extracting the node #39, the presence of obstacle cells (red) erases the current path and the algorithm establishes node #16 as the optimal one. The OPEN list is updated with new nodes (grey) until the target node #94 appears. Figure 11b shows the set of complete paths identified by the algorithm, all of them optimal because they have the same length. In this case, an additional parameter, i.e. the number of rotations that RAGNO has to actuate in order to follow an optimal trajectory, is taken into account. The optimal path is now defined as the one with the minimum length and the minimum number of rotations required and, as a result, the light brown trajectory is extracted.

Figure 11. A-star path planned (a) and paths tree generated during the research (b)

The computational time of this algorithm is about 300 ms. This time is affected by the complexity of the scenario. In fact, the greater are the obstacles cells and the lower are the cells to be examined thus obtaining a useful time savings.
The opposite behavior is shown by the potential strategy where the presence of more than an obstacle affects as the repulsive potential (there are more repulsive terms) as the integration process. Another advantage of the graph-based strategy is that local minima disappear as the minimization of the cost function (11) does not require an integration operation. In addition, the A-star trajectory is easier to follow than the one computed by artificial potential approach because the rover has to actuate only two movements i.e. a pivoting or a small forward displacement. The required control effort is smaller and more suitable for RAGNO onboard actuation system.

V. Test sessions

Performance of the designed guidance and navigation system in terms of autonomy and robustness has been evaluated during outdoor test sessions. Two scenarios have been taken into account: the first (Fig 12) is given by a single obstacle in front of the rover while the second (Fig 13) presents a hidden obstacle behind the foreground rock, thus resulting not visible in the first stereo image acquisition.

In order to verify the effective tracking of the planned path the current trajectory must be reconstructed by mean of an estimation process. In order to estimate the position of RAGNO step by step, both hardware and software integration is needed. The hardware includes the following onboard sensors:

- **wheels incremental encoders**, measuring the rotation angle of the wheels. If the sample rate is high, the angular velocity can be obtained by numeric derivative of two successive measures. As a consequence, the linear velocity and the yaw rate can be calculated too.
- **triaxial gyroscope**, measuring the angular velocity about three body axes (yaw, pitch, roll). In this case the yaw rate is the most significative measure because it describes the rotational dynamics about Y-axis of RAGNO. A change in yaw rate implies a change in heading angle. Gyro measures are affected by bias error so a calibration process has been carried out in order to filter it out. Calibrated measures allow to calculate the yaw angle evolution through a numeric integration of two successive samples.
- **triaxial magnetometer**, measuring the total magnetic field i.e. the sum of environmental and platform generated field. Since the magnitude decreases as $r^3$ where $r$ is the distance from the source the latter term can be filtered out by installing the sensor far from the field source (i.e. electric motors and onboard computer). In the case of the performed test sessions, Earth field is altered by different kinds of time variant perturbations as the test site is located near to a high-speed railway, and a low confidence has been indeed assigned to the magnetometer. In addition, the raw magnetometer measurements are affected by bias and scale factor errors, so that a calibration procedure is mandatory.

The acquired measures constitute the inputs of the implemented estimation algorithm (Eq. (12)). The linear Kàlmàn filter has been chosen since it provides real time state estimation in low computational time.

$$\dot{\mathbf{x}}_k = \mathbf{A} \mathbf{x}_k + \mathbf{B} \mathbf{u} = 0 + \mathbf{0}_{00}V 00 \mathbf{x}_k$$

The current state $\mathbf{x}_k$, consisting in yaw angle, angular rate and linear velocity, is estimated by summing the predicted state, i.e. the solution of the platform dynamics model, and the innovation term i.e. the difference between the current ($z_k$) and expected (H $\dot{\mathbf{x}}_k$) measures. The innovation term is weighed with the Kàlmàn gain matrix $K$. Since accelerations are not taken into account, the dynamics model is reduced to a simple kinematics one (Eq. (13)):

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} = 0_{13} \begin{bmatrix} \mathbf{R}_{\text{wheel}} & -\mathbf{R}_{\text{wheel}} \\ \frac{2L_{\text{semiaxis}}}{3} & \frac{2L_{\text{semiaxis}}}{3} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{\text{wheel}} \\ \mathbf{0}_{\text{wheel}} \end{bmatrix}$$

In order to obtain the (X,Z) position coordinates and the heading angle $\psi$, the following relations (Eq. (14)) must be computed:

American Institute of Aeronautics and Astronautics
The results of Kàlmàn filtering are shown in Fig 12 and Fig 13, that refer to successfully performed test. The areas represented by blue cells have been introduced all around the obstacle (after identifying its location) to add a safety zone.

\[
\begin{align*}
\begin{cases}
X_k = X_{k-1} + V_k \sin(\psi_k) dt \\
Z_k = Z_{k-1} + V_k \cos(\psi_k) dt \\
\psi_k
\end{cases}
\end{align*}
\]

(14)

Figure 12. Planned (black) and estimated (red) trajectories of the first test scenario

Figure 13. Planned (black) and estimated (red) trajectories of the second test scenario

In the last scenario, the hidden obstacle does not appear in the first stereo acquisition. The stereo image processing can not be performed when RAGNO is driving toward the goal because it needs consistent computational time. This means that it is necessary to integrate an auxiliary sensor which is able to detect a close object during the travel of the platform. An ultrasound sensor satisfies this need. As shown in Fig 14, when the sensor detects a close object, it sends a stop bit to the onboard computer which orders the rover to arrest. The whole guidance and navigation process restarts and a new safe path is planned. An example of successful test session has been uploaded to the YouTube “GN Lab” channel (https://www.youtube.com/watch?v=XOt2iRUeDag).

VI. Conclusions and further developments

The theme of autonomous robots is becoming more and more prevailing both in space and civil sector due to the wide range of applications. This growth is supported by the constant improvements in computer hardware, sensors technology and computer software. In addition, the recent developments in the field of computer vision and artificial intelligent allow to integrate and implement human-like systems and logics. In this paper an example of this application has been presented: the RAGNO, low-cost prototype of an autonomous vehicle has been proofed to be able to accomplish a transfer mission while avoiding the obstacles in the surrounding environment. The stereoscopic vision together with the processing algorithm are inspired by the human visual system and brain’s perception method. The advantages and drawbacks of this navigation system have been discussed in details.
Low-cost, simple and reliable autonomous rovers can change planetary exploration scenario. Till now, missions have been characterized by rovers working uncooperatively in specifically assigned and limited areas. The availability of low-cost autonomous platforms allows to afford missions involving the use of a swarm of cooperative rovers targeting – at the same time - the same celestial bodies. The logic of cooperation is inspired by nature swarm (birds, fishes or ants) and it is based on the fact that each member shares a set of information that might help the others in their mission. The advantages of this exploration approach are clear: parallel missions collect, without overlapping, a greater amount of coordinated data; at the same time, versatility and robustness to failure are largely increased.

References