VISION-BASED PERCEPTION AND SENSOR DATA INTEGRATION FOR A PLANETARY EXPLORATION ROVER

Zereik E.\textsuperscript{1}, Biggio A.\textsuperscript{2}, Merlo A.\textsuperscript{2}, and Casalino G.\textsuperscript{1}

\textsuperscript{1}DIST, University of Genoa, Via Opera Pia 13, 16145 Genoa, Italy
\textsuperscript{2}Thales Alenia Space - Italy, Strada Antica di Collegno 253, 10146 Turin, Italy

ABSTRACT

Current space mission plans foresee a strong robotic presence, with the purpose to rely on automatic systems for all tasks resulting difficult or dangerous to man. Stereo vision and data fusion play a fundamental role within the perception capabilities that a planetary rover must own in such a way to be effective in space missions. In fact, thanks to this skill, the robot can properly interact with the environment and can be autonomous and reliable enough to be entrusted with operations fundamental for the mission success. In the present paper, a project for the development of a robotic platform, which has to work as a terrestrial prototype, is generally described. More specifically, the stereo vision algorithm for the three-dimensional world point reconstruction is detailed, as well as the DEM (Digital Elevation Model) computation. Hints about a possible data fusion with data from a laser scanner are provided, together with some preliminary experimental results.

Key words: perception; vision-based DEM; laser scanner DEM; data fusion.

1. INTRODUCTION

As plans for space missions involving a massive robotic presence are widely being made, many activities in the field are being carried out. In particular, robotic autonomy represents a crucial point for future space missions, for the highly desirable capability of such systems to autonomously perform operations (important for the mission and maybe difficult or dangerous for astronauts) without any supervision from a human being. Within such a context, perception skills become essential to endow the robot with the capability to interact with an unknown environment. In this sense vision plays a key role in all automatic systems that have to autonomously perform complex operations. In this paper a prototype of an autonomous rover is described, particularly focusing on the environment perception; in fact, in order to correctly navigate on the planetary surface, the robot must “understand” its working area and build a DEM (Digital Elevation Model) describing it, in such a way to detect potential obstacles located on its path, thus avoiding them and safely reaching its goal. The overall algorithm is strongly based on stereo vision and showed good results; in the present work, this stereo vision-based 3D reconstruction and the DEM building are described and many experiments showing the system performances are detailed. Noise affecting the vision system can be rejected by the employment of different filtering techniques, in order to further improve the entire perception procedure. This aspect is particularly important in space activities, as human operator does not need to command the robot step by step but, on the contrary, it can execute more complex tasks in an autonomous manner, freely and safely moving within its working area: obstacles and terrain roughness detected by stereo vision are evidenced in the DEM and thus a simple path planning algorithm can be employed in order to compute a correct path toward the rover goal. Moreover, in order to obtain a more reliable and robust world 3D reconstruction with respect to noise, a multi-sensory data fusion can be exploited (e.g. with the employment of a laser scanner) in such a way to integrate information coming from multiple sources, thus neglecting the effect of noise on the camera images. For further details about multi-sensory schemes and data fusion refer to [7], [6] and [11].

2. ROBOTIC SETUP

This project is aimed to realize a rover platform suitable to test algorithms and useful capabilities considered essential for an effective space rover; within this context, the major focus is given to the external environment perception and its consequently mapping, executed by the rover to safely move and interact with its working area on the planetary surface. To this purpose, the available robotic setup is made up of a commercial PowerBot platform [8], shown in Fig. 1, equipped with sensors such as a set of bumper, sonar and laser scanner to precisely detect obstacles on the rover path and to quickly stop the robot motion whenever an emergency case occurs. Moreover, PowerBot has been equipped with two different stereo vision systems: a 	extit{Bumblebee2} stereo camera, placed on the rover back (straight above its top panel), which is ex-
exploited for navigation and localization purposes (i.e. a visual odometry module on-board the robot can estimate the occurred motion while navigating across the environment), and a Bumblebee XB3, relying on three distinct cameras, for the stereo 3D reconstruction and the actual DEM calculation. These stereo camera systems are depicted in Fig. 2; information about these vision systems can be found at [9]. Such a “three-eyed” camera system is placed onto the top of a mast, at a height of about 1.70 m, mounted on a Pan/Tilt unit (model PTU-D46-17, see [10]) in such a way to widen the range of the sensed (by the camera) world: thanks to this Pan/Tilt unit mobility, in fact, the rover is able to perceive the world all around itself, in a complete $360^\circ$ wide range. The overall realized robotic platform is shown in Fig. 3.

3. 3D WORLD RECONSTRUCTION

Exploiting the Bumblebee functionalities, the first step of this rover perception algorithm is represented by the surrounding world three-dimensional reconstruction (refer to [2], [5] and [1]); this procedure is highly affected

---

**Figure 1. PowerBot robotic setup**

(a) Bumblebee XB3

(b) Bumblebee2

**Figure 2. Rover stereo camera systems**

**Figure 3. Overall robotic platform**

**Figure 4. Pan/Tilt Unit driving the rover perception**
by errors of different nature: issues such as bad lighting conditions, wrong computed stereo correspondences, uncertainties in the camera model parameters, noise in the input images badly influence it. Thanks to the employment of a Bumblebee stereo system, the effect of some of these issues is strongly reduced: for example, the precise knowledge of the camera parameters and of the overall system stereo geometry highly increases the accuracy and the effectiveness of the stereoscopic reconstruction. Moreover, the possibility to exploit three cameras instead of two as in the classic stereo geometry can represent a great improvement within a perception system strongly relying on vision; however, this very interesting multi-view reconstruction is currently not implemented and should be addressed in the future, together with the possibility to choose different pairs of cameras (for example with a different stereo baseline) according to the specific situation to perceive (a varying field of view). Whenever the rover receives a command requiring it to make a new perception of the surroundings, the vision-based algorithm searches the right stereo camera (i.e. the Bumblebee XB3) within the system devices and starts acquiring pairs of stereo images chosen at specific location for the camera. In fact, the Pan/Tilt Unit, shown in Fig. 4, is commanded in order to assume a predefined number of specific position, in such a way to allow the rover to retrieve a wide shot of the overall environment around itself. These camera positions are computed in a very simple manner: first of all, knowing the whole desired spatial range around the rover to be covered, the number of necessary perception step $n_P$ is computed, simply as

$$n_P = \frac{R}{C_{fov}}$$  \hspace{1cm} (1)

where $R$ represents the spatial range to be covered and $C_{fov}$ stands for the camera field of view (in this case about 60°). For example, if the rover has to explore and perceive the external world in a range of 360°, 6 are the needed perceptions and so 6 pairs of stereo images are acquired by the system, one for each different camera pose. To this aim, the Pan/Tilt Unit must be driven in such a way to assume proper positions in order to allow the camera to take stereo pictures covering all the spatial range and, at the same time, to obtain images significantly describing the environment. The Pan/Tilt Unit provides 2 degrees of freedom that are to be controlled in a proper way; in this case the tilt angle is kept at a predefined value (about 30° pointing toward the ground) while the pan one, establishing the changes of the stereo camera position, assumes a different value for each image acquisition step. Such a value is determined as

$$\text{pan}_i = \text{pan}_0 + \frac{i}{n_P} \frac{R}{2n_P}$$  \hspace{1cm} (2)

where $i$ indicates the current perception step (e.g. in a case of 6 perception steps, $i$ ranges from 0 to 5), where as $\text{pan}_0$ represents the starting Pan/Tilt position (and so the stereo camera one) when acquiring the first image pair. This starting pan value obviously depends on the number of perceptions and on the spatial range to be covered and can be computed as

$$\text{pan}_0 = -\frac{R}{2} + \frac{R}{2n_P}$$  \hspace{1cm} (3)

An example of the positions assumed by the camera, while executing a roundup of the surrounding world, is depicted in Fig. 5 Each of these acquired pairs is used in order to retrieve the 3D structure of the world; in fact, the purpose of stereo vision is to perform range measurements based on images obtained from slightly offset cameras. Basically, three are the essential steps in stereo processing:

1. Establish the correspondences between image features in different views of the scene (i.e. between the two camera images).
2. Calculate the relative displacement between feature coordinates in each image.
3. Determine the 3D location of the feature relative to the cameras, using the knowledge of the camera geometry.

Consider, for example, the following Fig. 6, showing an image pair obtained from the horizontally displaced cameras of the Bumblebee camera module. Identifying two different points $A$ and $B$ in both images, it clear that point $A_{\text{left}}$ corresponds to point $A_{\text{right}}$ and, similarly, point $B_{\text{left}}$ corresponds to the point $B_{\text{right}}$. Measuring out the horizontal distance between the left edge of the images and the points, the distances in the left image result greater than the distance to the corresponding point in the right image. Such a distance (i.e. the point disparity, defined as the difference between the coordinates

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig5.png}
\caption{Camera position while performing a complete range world perception}
\end{figure}
Figure 6. Indoor stereo Images

of the same features in the left and right image) allows to determine the object distance from the camera module. Thanks to the Bumblebee stereo geometry, distances from the top of the image to the matching features are exactly the same in both images: this is because the cameras are horizontally aligned; for this reason, only the horizontal displacement is relevant to the calculation of the 3D world points (note, for example, that points closer to the camera have disparity values greater than the other points in the scene). There are quite different methods to find correspondences between two images; here the Sum of Absolute Differences correlation method is employed and based on the following procedure:

For every pixel in the image

1. Select a neighborhood of a given square size from the reference image
2. Compare this neighborhood to a number of neighborhoods in the other image (along the same row)
3. Select the best match
4. End

Figure 7. Image and world coordinate systems

The comparison of neighborhoods or masks is done using the following Sum of absolute difference formula:

\[
d_{\text{max}} \min_d \sum_{x=-\frac{m}{2}}^{\frac{m}{2}} \sum_{y=-\frac{m}{2}}^{\frac{m}{2}} [I_{\text{right}}(x+i, y+j) - I_{\text{left}}(x+i+d, y+j)]
\]

where:
- \(d_{\text{min}}\) and \(d_{\text{max}}\) are the minimum and maximum disparities
- \(m\) is the mask size
- \(I_{\text{right}}\) and \(I_{\text{left}}\) are the right and left images.

The distances from the cameras are determined using the displacement between images and the stereo system geometry. The position of the matched feature is a function of the displacement, the focal length of the lenses, resolution of the CCD and the displacement between cameras. Fig. 7 illustrates the coordinate system in which images and world measurements are represented; the image origin is in the top left corner of the upright image, while the origin of the world measurements is in the pinhole of the reference camera. Hence, converting depth maps into distance images, the consequent three-dimensional reconstruction of the world points is retrieved. Fig. 8 shows a single perception input image pairs (Fig. 8(a) and 8(b)) and the relative 3D point reconstruction (Fig. 8(c) and 8(d) showing two different views of the same reconstructed scene). For each stereo pair, the corresponding three-dimensional world points are reconstructed; at this point, in order to retrieve the world three-dimensional structure all around the rover current position, all the computed world 3D views are to be combined. To this aim, the reconstructed points for each perception are projected onto a common fixed frame \(<w>\); in such a way, the Pan/Tilt unit motion is taken into account. In Fig. 9 the back part of the stereo camera mounted on the PTU is shown and the relative positioning of the reference frames employed to model the PTU motion is illustrated. The already mentioned frame \(<w>\) is assumed to be fixed and independent of any rover or PTU motion (for example this reference frame can be placed in the rover starting position at the beginning of the mission and kept as a still absolute reference). Note that all the reference frames \(<\text{base}>\), \(<\text{pan}>, <\text{tilt}>\) and \(<\text{ptu}>\) are considered co-incident in the same point \(A\), indicated in Fig. 9, representing the intersection of the Pan/Tilt unit two rotation
Transformation matrices between \( <\text{ptu}> \) and \( <\text{base}> \) and between \( <\text{camera}> \) and \( <\text{tilt}> \) are obviously known and constant, while the matrices \( T_{\text{pan}} \) and \( T_{\text{tilt}} \) are functions of, respectively, the pan and tilt angles. As long as these two are null, the subsequent matrices are constant but, whenever the PTU moves according to the commanded perception steps, they have to be accordingly updated. Note how these pan/tilt rotations not only induce changes in the orientation components of the matrices \( T_{\text{pan}} \) and \( T_{\text{tilt}} \), but also in the translation part: in fact, since the camera does not rotate with respect to its frame origin but to another point, the projected points result slightly shifted. At each stereo image acquisition of a commanded perception procedure, as already suggested, the matrices are correctly updated and the 3D point projected onto the world frame \( <w> \) are computed: first of all the difference between the \( i \)-th world point projected on the camera frame and distance between the origins of the world and camera frames is performed, as explained by Fig. 10, to retrieve the vector linking the \( i \)-th world point \( P_i \) to the origin of the world frame \( <w> \); after this, the \( i \)-th point is simply pre-multiplied by the rotation matrix \( w_c R \) describing the relative positioning of the involved frames, as indicated in:

\[
_w P_i = w_c R (c P_i - w P)
\]  

(5)

For the displaying of the 3D reconstructed world, an OpenGL viewer has been realized: in Fig. 11 a complete range perception around the rover, while navigating outdoor, is shown. Note that the edges of the perceived world are cut in the 3D reconstruction with respect to the input images: as the more distant are the points, the more they are affected by noise, the suitable choice of relying only points falling within a specific depth from the camera has been made. As it can be seen from Fig. 11, however, some noise is still affecting the reconstruction and so the subsequent DEM construction too is influenced by it (that’s why an effective and robust method for error rejection is to be applied). Moreover, note that in the overlapped areas of the three-dimensional reconstruction (i.e. zones made up by points that fall within the camera field of view in two subsequent different view of the same perception), points are currently overwritten (points of the
second view- in terms of its acquisition time- are kept and visualized); obviously enough, this is a still open issue and a proper policy correctly managing such a point overlapping should be implemented.

4. DEM COMPUTATION

At this point, the DEM map of the perceived environment is to be built up: in this rover vision system, the DEM is a $400 \times 400$ pixel planar image and each single DEM pixel represents a little $5 \, \text{cm} \times 5 \, \text{cm}$ square of world points. Hence, in order to retrieve the correspondent planar representation of the perceived world, the two planar coordinates of each 3D point are exploited to find the location of this point on the DEM map and to compute its $(i, j)$ image coordinates. Finally, the third coordinate of the 3D point is used to assign a color code to each pixel: in Fig. 12, blue points represent navigable areas (that is terrain portion free from any obstacle or roughness), while red pixels stand for obstacles and so zones not to be crossed by the rover; black areas represent unknown and still unexplored points. Consider the

![Figure 12. DEM computed in relation to the previously described rover outdoor perception](image)

pixels around the box on the ground in Fig. 12. Black pixels just behind the box are due to the occlusion produced by the box itself: terrain points behind it are not visible and so the corresponding DEM portion is marked as unknown. Since the 3D reconstruction is, as already suggested, prone to errors due to noise, mismatched correspondences, bad illumination and so on, a suitable strategy just like a smart filtering of the input data is to be identified and implemented for the system. At the present time, a simple Median calculation is executed: for each pixel $(i, j)$, all the 3D point values falling on this current pixel are stored within a preliminary Median DEM Map. When all the computed points have been considered and correctly sorted within such a map, to each DEM pixel
(i, j) only a single value is assigned, that is the median element of the vector including all the points belonging to the current DEM pixel.

5. LASER SCANNER AND DATA FUSION

As the robot is endowed also with a 3D laser scanner, such a device can be exploited to build up DEMs of the surrounding environment; this is done through a Sick LMS200 mounted on a Maxton continuously rotating pan unit, as shown in Fig. 13. This laser scanner can provide a resolution of 0.25° on rotation and an accuracy of 1 mm up to 30 m. Whenever a perception is required to the device, after a complete revolution, the 3D laser scanner retrieves a cloud of points, each indicated as \([x, y, z]^T\), with respect to its reference frame, located in the middle of the Sick scanner; the computed points are then correctly transformed and expressed with respect to a common reference frame, in such a way to be comparable with those evaluated by the vision system. Finally, the DEM is created with the desired granularity (usually 0.05 m). At this point, as the system can rely on two different DEMs describing the same environment but coming from two different sensors, many policies are possible. The simplest one is that of trusting only one of the two sensors (vision or laser scanner), discarding data from the other one. In such a case, only a filtering of the cloud point values is requested and then the DEM is computed. Anyway, a more reliable policy can be adopted, exploiting all available data: a sort of “priority” is assigned to each sensor and DEM points from the higher priority sensor are firstly filtered. Then, only if the point is invalid after the filtering process, data from the second sensor DEM are exploited; if the point remains still invalid after this procedure, then the output point will be invalid too. Obviously enough, a third and more effective method can be implemented, that is the actual “merging” of sensor data: each computed point of each sensor could be assigned to a “confidence” value according to some criteria (for example one can trust less points of the vision DEM far away from the camera or located at the very sides of its field of view in each scan) and then a simple weighted fusion could be performed. However, this last technique is still under consideration and is being analyzed and tested.

Finally, as already said, DEMs are first filtered before the merging; this is because, being such DEMs usually affected by salt and pepper noise, it is useful to choose a proper filtering technique, in order to improve data that are to be used to plan the rover motion within safe areas. With this aim in mind, different filters can be employed, all of them being however punctual (e.g. the already mentioned Median filter), also thanks to their shown efficiency.

6. DISCUSSION AND FINAL CONCLUSION

Thanks to the stereo camera, the three-dimensional world can be reconstructed, exploiting a simple stereo matching based on the perfect knowledge of the camera parameters and improved by a plain Sum of Absolute Difference correlation algorithm; performed tests highlighted how the stereo reconstruction is good enough in such a way to effectively find obstacles or roughness in the terrain that is to be faced by the rover. The computed 3D points are then employed to retrieve the DEM map corresponding to the actual position of the rover. Note that both the shown 3D reconstruction and DEM have been built on the basis of the vision-based algorithm only, without any integration with other sensors. Both in the 3D reconstruction and in the DEM computation, some noisy points can be detected, above all at the edges of the vision field: they are due to issues such as bad illumination, false correspondences in the stereo images and their effect can be reduced via the introduction of procedures such as a filtering technique or a better method improving the overall perception algorithm. At this time, the system takes into account all the 3D points belonging to the same DEM pixel and chooses their median value as the correct map point; improvements in this sense are to be analyzed and tested. Data from vision can then be merged with information from other sensors, for example exploiting data coming from a laser scanner that helps the camera system whenever in bad conditions: the map validity can be checked using information from this sensor and the resulting map can then be sent to a standard path planning algorithm that has the job to find the safest and best (also in terms of path length) way toward the rover goal. Such a data fusion procedure is being analyzed and implemented. Moreover, another possibility to investigate consists in a better use of the three-eyed stereo system: instead of exploiting only two cameras of the three available, thus performing a traditional stereo vision algorithm, more robustness can be added to the system via a multi-view (three, in the present case) reconstruction. In fact, an interesting research topic is that of exploiting the trifocal tensor, that may be used to transfer points from a correspondence in two views to the corresponding point in a third view. For details on this technique refer to [2]; moreover, for further information about multi-view world reconstruction see [4] and [3]. Anyway, preliminary results of this system, partially shown in the previous sections, appear to be promising enough for the...
continuation of the project; improvements as performing a reliable data fusion or developing a filtering technique robust to noise are surely important steps to be executed, even if preliminary tests showed the rover capability to detect and avoid obstacles along its path via its effective vision-based perception skills.

REFERENCES


