Recent Innovations in Mechatronics (RIiM) Vol 6. (2019) No 1.
DOI: 10.17667/riim.2019.1/9.

Hopping Rover Navigation Method for Rugged Environments

Gabor Kovacs, Yasuharu Kunii and Hideki Hashimoto
Faculty of Science and Engineering
Chuo University
Tokyo, Japan

***Abstract*—In this paper a navigation method is presented for space exploration robots using hopping motion in environments with large elevation differences. A monocular camera system is used to reconstruct the flight trajectory and environment around the robot using Structure from Motion while traveling. The created environmental point cloud is projected to 2D to create a variable resolution image and image processing is used to find the most suitable position for the next landing based on normals with the help of gradient maps and error estimation. The method is evaluated in a simulation environment against the previously used protrusion based method to show that the proposed system can extend the operation of the robot to terrains with large elevation differences while still successfully avoid obstacles and dangerous areas.**

***Keywords*—Hopping Rover, Navigation, Image Processing, Rugged, Space Exploration**

I. Introduction

With the recent success of space exploration rover MINERVA-II launched by the Japan Aerospace eXploration Agency (JAXA) that was able to navigate on the surface of an asteroid for the first time using hopping motion, non-traditional movements of robots are being developed [1]. Asteroids and other small celestial bodies have low gravity and thus making it possible to utilize jumping/hopping motion without large energy consumption. While traditional wheeled robots have limited ability to overcome obstacles and climb steep surfaces a hopping robot can easily navigate through environments with large obstacles or elevation differences. Other benefits of a hopping robot are reduced size and complexity both of which are essential for space missions. A decrease in size and weight makes it possible to launch multiple robots in the same space mission and also increase robustness by reducing the number of moving parts that can fail. A working prototype of a hopping rover can be seen in Fig. 1. A loaded spring is used for hopping and the robot can turn to the desired direction using a single wheel [2]. For visual odometry multiple approaches has been proposed including stereo camera systems [3], monocular camera system that use stereo cameras only for the initialization step [4] and systems that use multiple monocular cameras facing in different directions [5]. The system presented in this paper uses a single monocular camera with Structure from Motion and a trajectory- and scale estimation step based on the known gravity and parabolic motion constraints to reconstruct the flight path and surrounding environment. The navigation method is based on selecting the most suitable landing position from this environment for the next hop after each landing. It needs a special path selection algorithm, which is different from the usual obstacle avoidance methods of mobile robots. A general path planning concept was proposed in [6], which enables the system to learn the most suitable guiding style adapting to the given environment. The system proposed in this paper moves like a grasshopper. Another unusual method is presented in [7], which aims to mimic animal movement behavior. In the previous implementation of the proposed system the landing position criteria was evaluated based on protrusion that limited the navigation to relatively flat areas with obstacles [8]. The motivation for the development of the proposed system is to extend the operation of the hopping rover to rugged environments and to be able to explore hills, mountains and other rough terrain. To achieve this normal information of the environment is used to evaluate the surrounding area. Section II presents an overview of the whole system and section III details the proposed new method for selecting the next landing point. In section IV the viability of the proposed method is proved, compared to the previous method and the results are concluded.

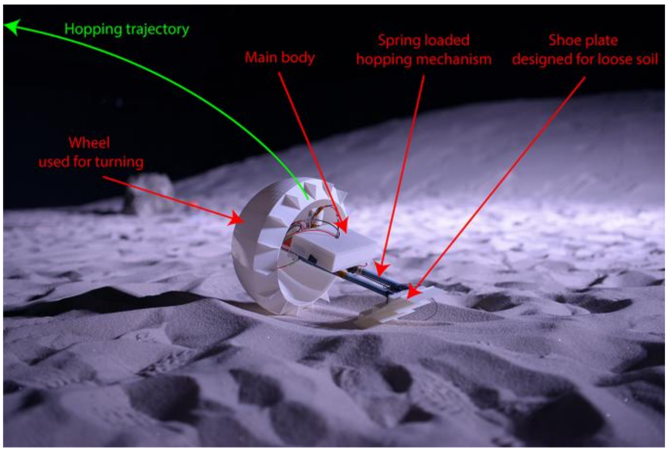
Hopping trajectory
Main body
Spring loaded hopping mechanism
Shoe plate designed for loose soil
Wheel used for turning

Fig. 1. Working prototype of the hopping rover

II. Overview of the Navigation System

Fig. 2 shows the overview of the whole hopping rover navigation system. Although this paper focuses on the landing

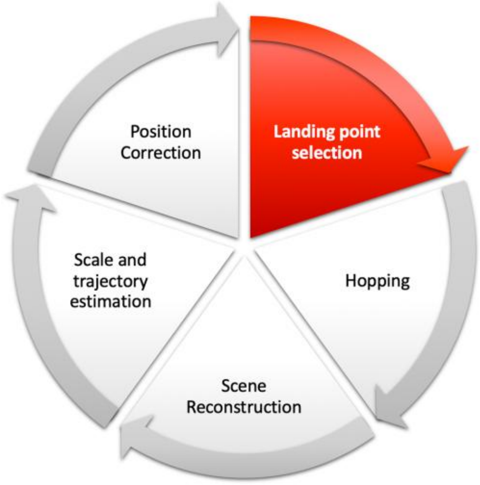

Fig. 2. Overview of the navigation system

point selection part the whole system is briefly presented in order to better understand the limitations and goals of deigning the landing position selection method.

A. Scene reconstruction

During each hop a wide angle camera attached to the rovers body takes images that are used for mapping and localizing the rover. In order to create a highly detailed map of the surrounding environment during exploration missions our system uses Structure from Motion (SfM) to create a sparse point cloud and also reconstruct the camera poses [9]. SfM provides more accurate and detailed results compared to Simultaneous Localization and Mapping (SLAM) algorithms while sacrificing computation time. However since in space exploration real time operation is not required the rover has time to process the data after each landing. With the known camera poses a MultiView Stereo (MVS) pipeline can calculate the position of each pixel creating a dense point cloud [10]. Fig. 3 shows an example of hopping with the flight trajectory in a simulated environment. The image frames taken during flight and used for reconstruction can be seen in Fig. 4.

B. Trajectory and scale estimation

Although SfM and MVS can reconstruct the camera positions and environment relative to each other from the monocular camera images, the real scale has to be determined. Since we know that no other force work on the rover other than gravity during hopping, the center of gravity travels on a parabolic path. Introducing parabolic motion constraints with time of flight information and the known gravity acquired by sensors, the real scale can be determined and the point clouds transformed.

C. Position correction

After each landing the rover can bounce off the surface in an unpredictable way that introduces position error. Since the camera is too close to the ground after landing, feature point matching cannot be used reliably to calculate this deviation and therefore the launch position of every hop is assumed to be the landing position of the last one. To ensure accuracy after several jumps, a position correction step is introduced by aligning the dense point clouds of successive hops using a modified Iterative Closest Point (ICP) algorithm [11]. Since the ICP algorithm has a tendency to converge to a local minimum instead of the correct alignment when there is partial overlap between the point clouds, only a section of both dense clouds are aligned iteratively and after each step the non-overlapping parts are eliminated.

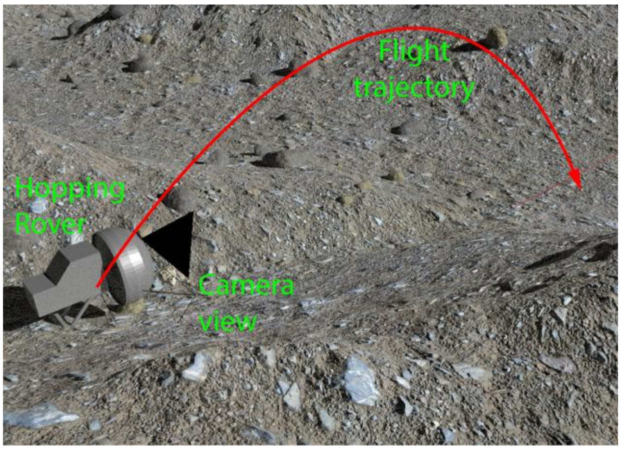

Fig. 3. Hopping in a simulated environment

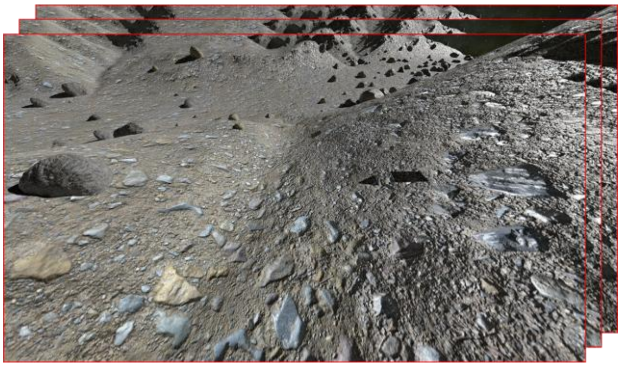

Fig. 4. Successive image frames used for reconstruction

III. Landing Point Selection Based on Normals

Traditional path planning methods have a general understanding of the environment and obstacles to be avoided. However in our system the environment is only reconstructed from the monocular camera images after each hop and therefore the robot only knows the desired direction of travel and selects the next landing position based on that. The most important aspect for selecting suitable areas is to make sure the rover avoids hitting obstacles. These obstacles are rocks and boulders of various shapes and sizes scattered on the generally loose ground. In the previous implementation of our system the evaluation criteria was based on protrusion. Since the rover

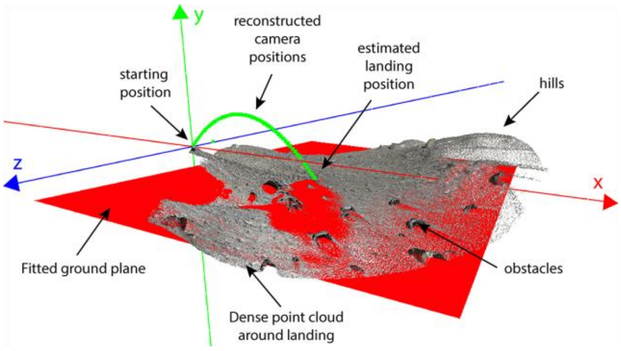

Fig. 5. Ground plane fitted to the surrounding area

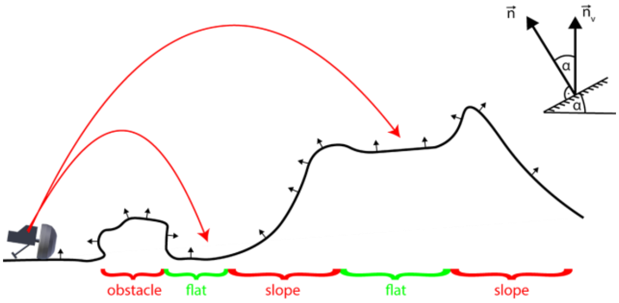

Fig. 6. Evaluation based on normals

was only intended to work in relatively flat areas a plane can be fitted to the dense point cloud to mark a ground plane using the RANSAC (RANdom SAmple Consensus) algorithm [12]. Using RANSAC is beneficial because it disregards outliers and makes to fitted plane closer to the real ground. The distance of every point is then calculated from the plane and if they are over a certain threshold value the point is considered an obstacle. This method however is only useful if the surrounding area is flat since it excludes potential landing areas found at different heights such as plateaus or valleys. Fig. 5 shows the dense point cloud created from the jump seen in Fig. 3 and the ground plane that is fitted to the points that are less than $10m$ away from the estimated landing position. From this image it is clear that due to the variation in elevation a plane is unable to separate the ground from obstacles. In order to make sure that the rover is able to travel in rugged terrain a different evaluation method is proposed based on the normal information. The normal vector of each vertex is produced during dense reconstruction along with depth maps. Let a denote the maximum slope angle the rover can safely land on. A vertex can be classified as a flat area if the vertical (parallel to the gravity vector) component of the normal vector $n_v \geq cosa$, and dangerous to land on otherwise. A 2D representation of the obstacle and slope detection can be seen in Fig. 6.

A. Variable resolution image creation

Image processing algorithms are powerful and provide flexibility in manipulating the environment information so the

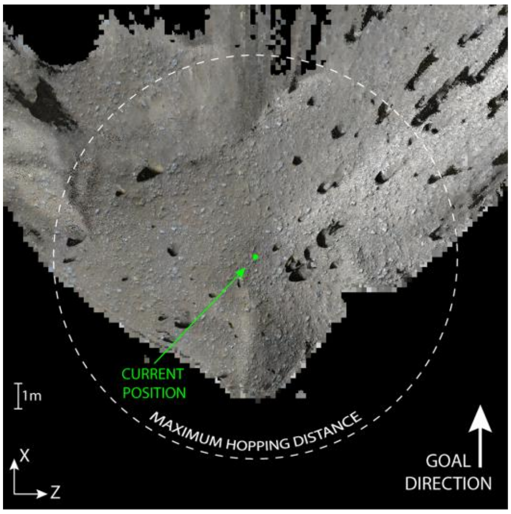

Fig. 7. Variable resolution image created from the dense point cloud

dense point cloud vertices are projected to the XZ plane that has a normal vector parallel to the gravity vector. A square image is created in a way that the goal direction is upwards and the image center corresponds to the current location of the rover. The image covers a somewhat larger area than the maximal allowed horizontal jumping distance in every direction. Each vertex is assigned to a pixel based on its coordinate values. Where multiple points correspond to the same pixel, the arithmetic mean of values for each color channel is assigned. Since the vertices were calculated from the camera image frames the distance between points linearly increases with distance from the rover. If the chosen image resolution is high the fine details of closer points come alive but more black pixels will appear in more distant areas where no vertex is assigned to that pixel. This could be treated as noise and reduced with median filtering, however this would also blur the image details. On the other hand by using lower resolution images black pixels can be avoided while sacrificing image quality. To solve this problem a variable image creation method is introduced. The projection is done with different image resolutions forming a pyramid from 100x100 to 800x800 pixels. After the images are separately created, the lower resolution images are linearly upscaled to the highest resolution. For every black pixel of the original high resolution image the lower resolutions are checked in order, and if nonzero values are found the missing information is filled in. This method creates a variable resolution image that can be seen in Fig. 7. The false color Fig. 8 shows which level each pixel is obtained from. A binary image mask is created with the same variable resolution containing the information about flat areas.

B. Error estimation

In order to make sure the rover does not hit any obstacle while hopping, the uncertainties of movement has to be taken into consideration. To estimate the possible deviation from the

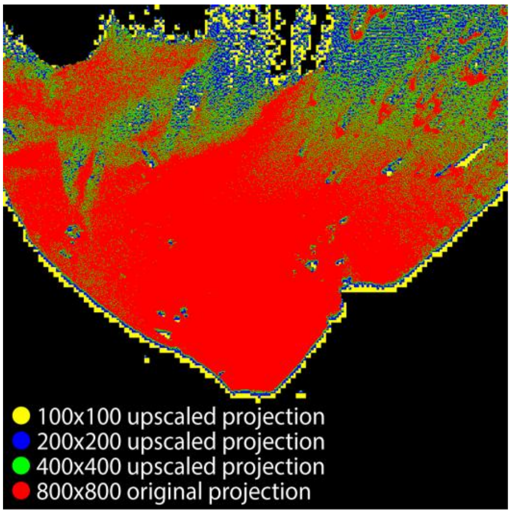
● 100x100 upscaled projection
● 200x200 upscaled projection
● 400x400 upscaled projection
● 800x800 original projection

Fig. 8. False color image showing the used resolutions

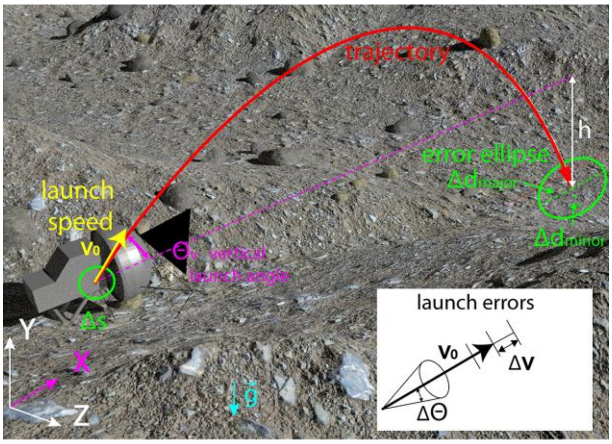

Fig. 9. The error ellipse and its components

desired landing position the following errors are used that are gathered during experiments: $\Delta\Theta$ angular error coming from launching from an uneven surface, Δv launch velocity error and Δs uncertainty of the current position. The error of the sensed gravity Δg and error of the current and target position height difference Δh are negligible and considered zero to simplify calculations. The rover uses a compressed spring to release the energy used for hopping and Δv is linear to spring compression Δx consequently meaning that Δv is independent of the hopping distance along with the other errors. From these parameters an error ellipse can be calculated for each possible landing position. Fig. 9 shows the error ellipse and its components. The equation of the horizontal hopping distance is:

$$f(v_0, \Theta_v, g, h) = d = \frac{v_0^2}{2g}\left(1 + \sqrt{1 + \frac{2gh}{v_0^2 sin^2(2\Theta_v)}}\right) sin(2\Theta_v) \quad (1)$$

where d is the horizontal distance of the landing point, v_0 is the launch speed, g is the gravity constant, h is the height difference of the target landing and current position and Θ_v is the vertical launch angle. The errors are assumed to have Gaussian distribution and the variables are independent from each other which means the variance formula can be used to calculate the error propagation, and the major axis of the error ellipse is:

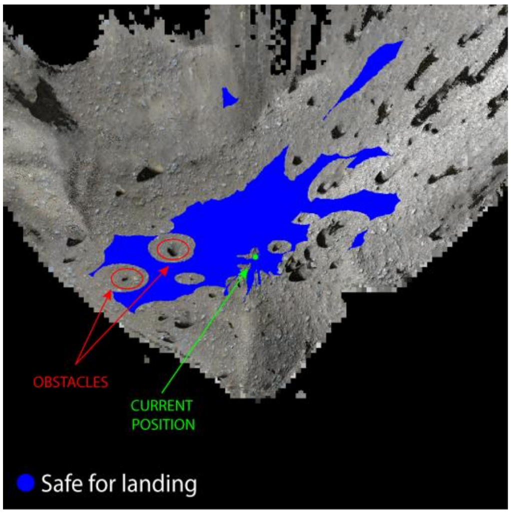

Fig. 10. Areas marked safe for landing

$$\Delta d_{major} = \sqrt{\left(\frac{\partial f}{\partial \Theta_v}\Delta\Theta\right)^2 + \left(\frac{\partial f}{\partial v_0}\Delta v\right)^2 + (\Delta s)^2} \quad (2)$$

the minor axis can be calculated as:

$$\Delta d_{minor} = \sqrt{d^2 sin^2(\Delta\Theta) + (\Delta s)^2} \quad (3)$$

The error ellipse is calculated for every pixel of the flat area image mask mentioned previously. If the area defined by the ellipse contains black pixels in the mask either due to missing information or excluded by normal value, the ellipse center pixel is marked as not safe to land in a newly created binary mask. Fig. 10 shows the areas marked safe for landing overlayed on the original map. The safety distance from obstacles increases with the hopping distance. The method was also able to mark areas safe on different elevations.

C. Selecting the most suitable landing point

From the created mask containing the possible landing locations the best candidate needs to be selected based on a number of criteria. A maximum hopping distance is set due to physical limitations of the hopping mechanism, however larger hops are preferred. A minimum distance is also set to make sure a sufficient number of images are taken during flight. The rover should hop towards the goal directions however it is also important to reduce the risk of hitting obstacles or landing on a sloped surface. To quantify these requirements a number of grayscale images a created. By applying Euclidean distance transform to the mask of the possible landing points a safety image can be created where the value of each pixel represents

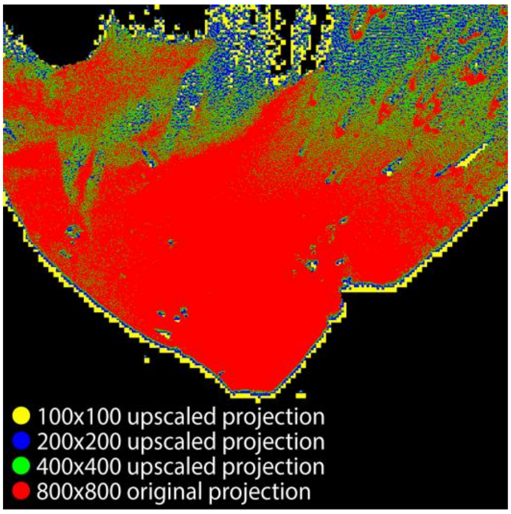
● 100x100 upscaled projection
● 200x200 upscaled projection
● 400x400 upscaled projection
● 800x800 original projection

Fig. 11. Result of the distance transform

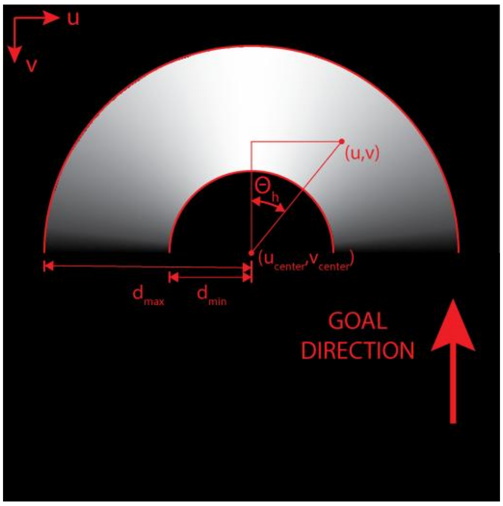

Fig. 12. Angular gradient map

the distance from the closest black pixel seen in Fig. 11. An angular gradient map is created to allow a maximum of 90° deviation from the goal direction using the following equation:

$$I_{ang}(u, v) = \begin{cases} cos(\Theta_h) = \frac{v_c - v}{\sqrt{(u_c - u)^2 + (v_c - v)^2}}, & \text{if } d_{max} \geq \sqrt{(u_c - u)^2 + (v_c - v)^2} \geq d_{min} \text{ and } v \leq v_c \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Where u, v are pixel coordinates, u_c,v_c are the image center coordinates, d_{max} is the maximum allowed hopping distance and Θ_h is the horizontal launch angle defined as the angle between the goal direction and the line between (u, v) and (u_c, v_c). The angular gradient map is shown in Fig. 12. In a similar fashion the preference for more distant point can be quantified with the following equation:

$$I_{dist}(u, v) = 1 - e^{-g_{dist} \frac{\sqrt{(u - u_c)^2 + (v - v_c)^2} - d_{min}}{d_{max} - d_{min}}} \tag{5}$$

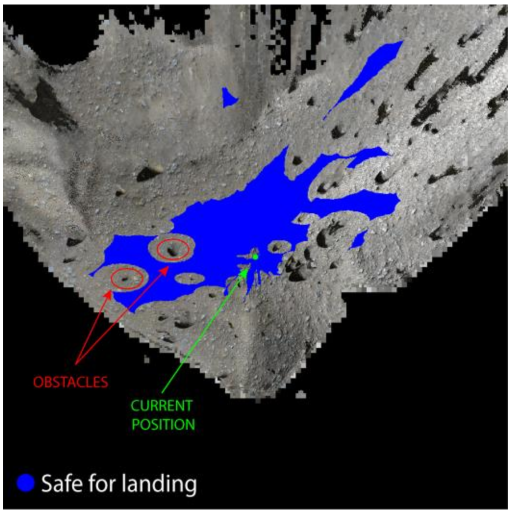

Fig. 13. Distance gradient map

where d_{min} is the minimum hopping distance and g_{dist} is a gain constant. The resulting distance gradient map can be seen in Fig. 13. The final selection map is created by combining the distance transform image with the angular and distance gradient maps. The images are normalized and the pixel-wise geometric mean is calculated. It is possible to put emphasis on certain criteria by calculating weighted mean where the weights correspond to the importance of a criteria. The next target for landing is the location of the maximum value in the selection map that is transformed back to real world coordinates and the necessary launch speed is calculated from (1). Fig. 14 shows the selected landing position and calculated error ellipse.

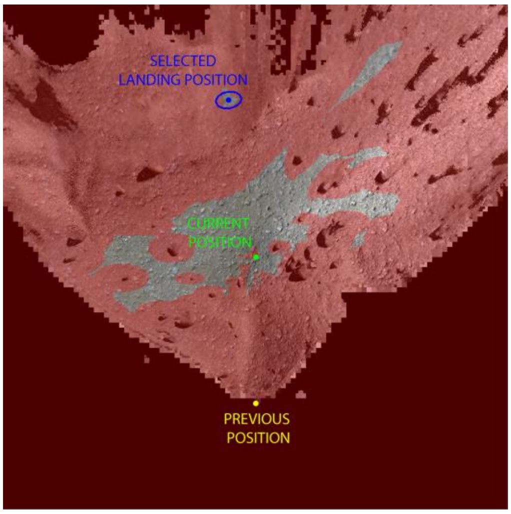

Fig. 14. Selected landing position

IV. Evaluation and Conclusion

A variety of different environments were used to test the proposed method including computer generated terrains

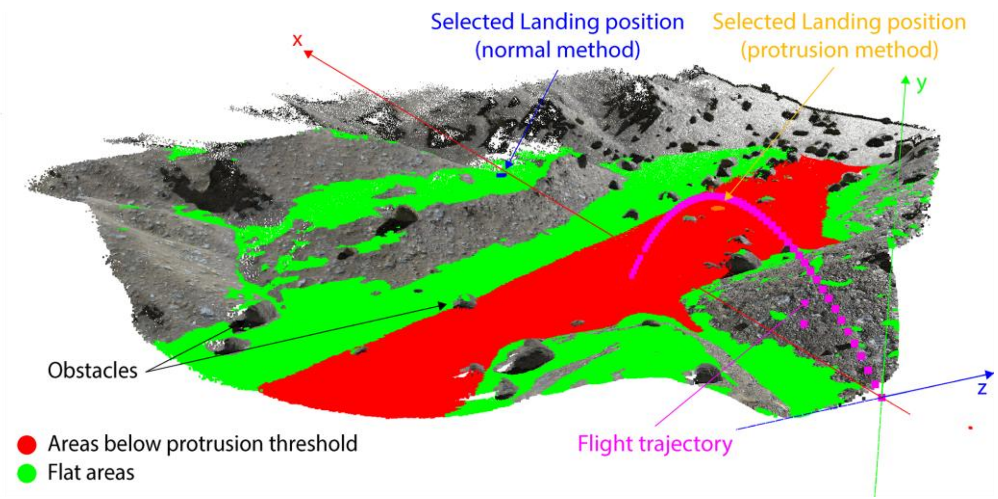
Selected Landing position
(normal method)
Selected Landing position
(protrusion method)
x
y
z
Obstacles
Flight trajectory
● Areas below protrusion threshold
● Flat areas

Fig. 15. Landing position selection methods based on protrusion and normals

with scattered rocks and highly detailed photoscanned surface models. The main motive for the proposed system was to create a navigation method that extends the rover operation environment in rugged terrains. To verify the viability of the new normal vector based method it was compared to the previously suggested protrusion based method. The hopping was simulated in a 3D rendering software [13] and the image frames were taken at a fixed interval including effects such as motion blur and rolling camera shutter. The created point cloud was then used to select the next landing position with both the conventional and proposed method. Fig. 15 illustrates the advantages of using normals. The voxels marked with red are the ones below the protrusion threshold and the green voxels represent the areas identified as flat. Both methods are able to identify and avoid obstacles, however the normal method covers a substantially larger area. The landing point selected by the protrusion method in this example is $5.4m$ away with 48° horizontal deviation from the desired goal direction of the x axis and the landing position selected by the normal method is $6.3m$ away with only 10° deviation and located on a plateau of the hill ahead. Since the protrusion method is unable to cover areas with different elevation the path planning will ultimately be stuck in the valley the rover is currently at. From the performed experiments we can conclude that the proposed method is able to navigate the rover in difficult terrains while avoiding obstacles. The variable resolution image provides the necessary data even in distant areas and larger hops can be performed while preserving detail for global map creation. The main limitation of the system is that path planning relies on the dense point cloud created in the previous hop since no other source of information is available about the environment. Future research topics include a more sophisticated landing error calculation model where the impact speed and angle are used to predict possible secondary bounces after landing.

Acknowledgment

A part of this work has been supported as joint project research subject by the Institute of Science and Engineering in Chuo University.

References

[1] H. Yabuta, “Arrival, touchdown and sequel to the voyage of hayabusa2”, Nature Astronomy, vol.3., 2019, pp.287.

[2] K. Yoshikawa et al., “A new mechanism of smart jumping robot for lunar or planetary satellites exploration”, IEEE Aerospace Conference, 2017 p. 1–9.

[3] L. Raura, A. Warren and J. Thangavelautham, ”Spherical planetary robot for rugged terrain traversal,” 2017 IEEE Aerospace Conference, Big Sky, MT, 2017, pp. 1-10.

[4] K. Otsu, T. Maeda, M. Otsuki and T. Kubota, “A Study on Monocular Visual Odometry using Parabolic Motion Constraints”, The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec) –in Japanese, 2016. 2A2-17b5.

[5] E. So, T. Yoshimitsu and T. Kubota, “Visual Odometry for a Hopping Rover on an Asteroid Surface using Multiple Monocular Cameras”, Advanced Robotics, 2011, pp. 893-921.

[6] P. Baranyi, I. Nagy, P. Korondi and H. Hashimoto, ”General guiding model for mobile robots and its complexity reduced neuro-fuzzy approximation”, IEEE International Conference on Fuzzy Systems, San Antonio, TX, USA, 2000, pp. 1029-1032 vol.2.

[7] B. Kovacs, G. Szayer, F. Tajti, M. Burdelis and P. Korondi, ”A novel potential field method for path planning of mobile robots by adapting animal motion attributes”, Robotics and Autonomous Systems vol. 82, 2016, pp. 24-34.

[8] G. Kovacs, Y. Kunii, T. Maeda and H. Hashimoto, “Trajectory Estimation and Position Correction for Hopping Robot Navigation using Monocular Camera”, unpublished

[9] S. Bianco, G. Ciocca and D. Marelli, “Evaluating the Performance of Structure from Motion Pipelines”, Journal of Imaging, vol.4., 2018, pp. 98

[10] Y. Furukawa and J. Ponce, ”Accurate, Dense, and Robust Multiview Stereopsis,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 8, pp. 1362-1376, Aug. 2010.

[11] P. J. Besl and N. D. McKay, ”A method for registration of 3-D shapes,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[12] R. Raguram, O. Chum, M. Pollefeys, K. Matas and J.M. Frahm, “Usac: A universal frame- work for random sample consensus.” IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013, vol. 29, pp. 2022–2038.

[13] Blender, https://www.blender.org/ , Accessed 2019-10-15