

Article

An Efficient Sampling-Based Path Planning for the Lunar Rover with Autonomous Target Seeking

Gang Chen ¹, Hong You ¹ , Zeyuan Huang ^{1,*} , Junting Fei ¹, Yifan Wang ² and Chuankai Liu ^{3,4}

¹ School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China; buptcg@163.com (G.C.); 18813133130@163.com (H.Y.); 18810558852@163.com (J.F.)

² School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China; wangyifan@bupt.edu.cn

³ Beijing Aerospace Control Centre, Beijing 100094, China; ckliu2005@163.com

⁴ Key Laboratory of Science and Technology on Aerospace Flight Dynamics, Beijing 100094, China

* Correspondence: huangzeyuan1994@163.com; Tel.: +86-135-2183-2124

Abstract: This paper presents an efficient path planning method for the lunar rover to improve the autonomy and exploration ability in the complex and unstructured lunar surface environment. Firstly, the safe zone for the rover's motion is defined, based on which a detecting point selection strategy is proposed to choose target positions that meet the rover's constraints. Secondly, an improved sampling-based path planning method is proposed to get a safe path for the rover efficiently. Thirdly, a map extension strategy for the unstructured and continually varying environment is included to update the roadmap, which takes advantage of the historical planning information. Finally, the proposed method is tested in a complex lunar surface environment. Numerical results show that the appropriate detecting positions can be selected autonomously, while a safe path to the selected detecting position can be obtained with high efficiency and quality compared with the Probabilistic Roadmap (PRM) and A* search algorithm.

Keywords: lunar rover; selection of detecting point; path planning; map extension



Citation: Chen, G.; You, H.; Huang, Z.; Fei, J.; Wang, Y.; Liu, C. An Efficient Sampling-Based Path Planning for the Lunar Rover with Autonomous Target Seeking. *Aerospace* **2022**, *9*, 148. <https://doi.org/10.3390/aerospace9030148>

Academic Editor: Jekanthan Thangavelautham

Received: 20 December 2021

Accepted: 6 March 2022

Published: 8 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Moon is rich in minerals and energy resources and has been one of the most attractive destinations for space exploration [1]. As a specialized mobile robot on the lunar surface, the lunar rover can adapt to the harsh environment, cross obstacles, and bear large loads [2]. Therefore, it plays a prominent role in lunar exploration and has been widely used in plenty of lunar missions. The former Soviet Union's lunar probes "Luna 11" and "Lunar 21" carried lunar rovers, which completed tasks including lunar reconnaissance, topographic photography, and mineral composition analysis with various scientific instruments [3]. The "Apollo 15" and "Apollo 16" of the United States carried manned lunar rovers [4]. Recently, China launched two lunar rovers the "Yutu" and "Yutu-2", to detect the lunar surface [5]. In the near future, more missions including lunar rovers have been proposed, for instance, the USA's Artemis and China's lunar program IV, for more thorough exploitation and utilization of the lunar resources [6].

Generally, the lunar surface exploration is carried out based on the information of the lunar environment after a safe landing. Common exploration tasks include a geological survey of the lunar surface, soil sampling, rock mineral analysis, etc., and the process of them is mainly divided into four steps:

- Perception of the lunar environment. It mostly relies on satellites and the vehicular cameras to gather information about the lunar environment, from which a digital elevation model (DEM) is generated that includes details like rocks, craters, slopes, etc. [7].
- Selection of the detecting point. A suitable target detecting point is usually selected based on the information of the lunar environment and the structure of the lunar rover.

- Path planning. The path of the rover is planned in the safe area to ensure the arrival at the detecting point, in which the key is the avoidance of obstacles.
- Execution of the detecting task. After the arrival of the target point, sampling and other scientific missions are carried out by using the sampling manipulator and other scientific instruments.

Normally, the first three steps are performed by the rover, while the fourth step is executed by the manipulator, which is beyond the scope of this study. In recent years, the perception technology of the surface environment has made great progress, and can be conducted with high precision [8]. However, the selection of the detecting point and path planning for the lunar rover still mainly rely on manual control from ground personnel, which makes it inefficient to transmit large amounts of surrounding information and path planning data between the lunar rover and the ground console. Moreover, due to the time delay of transmission between the Moon and the Earth, the manual command of the ongoing task may violate the feasibility in the unstructured lunar surface environment, and even threaten the lunar rover's security [9]. In recent years, plenty of research has been carried out on these two abovementioned topics.

For the selection of the detecting point, to ensure it is appropriate and reachable, the safe zone for the rover's motion should be determined first. For this purpose, Li [10] analyzed the safe zone by extracting the topographic slope of the lunar surface. Garrido [11] solved the safe zone by calculating the maximum and minimum slope based on the tensorial way at any point on the lunar surface. Zhou [12] generated the goodness raster map based on the slope cost function to solve the safe zone. These studies analyzed the safe zone for rover's motion only based on slope, while small obstacles, which cannot be expressed by the slope, are eliminated macroscopically. Moreover, the energy consumption of those small obstacles is ignored for the selection of a detecting point within the obtained safe zone. Kose [13] chose it according to the illumination requirements of the Raman spectrometer, while Hewitt [14] selected it by analyzing the topographic features of each grid in the raster map, such as point variance, average height, etc. The abovementioned studies consider the equipment requirements or terrain features separately, which are not feasible for the actual task because the real lunar environments are complex.

Once the detecting point is selected, a safe path from the current position to the detecting point should be searched within the safe zone. Orger [15] planned the path of lunar rovers based on the A* algorithm, without taking into consideration the physical size of the lunar rover. The "Yutu" adopted the A* algorithm to find the optimal path in the DEM of the lunar surface [16]. Wang [17] applied the A* algorithm for the path planning and optimization of the lunar rover to improve the path quality. However, the efficiency of the A* algorithm is limited in a large-scale environment, since additional manual correction is required for practical engineering applications. Seo [18] realized the path planning of the lunar rover from the actual landing position to the expected landing position based on the D* algorithm, which is limited in practicability since the shape of the obstacles is assumed to be ideal. Saito [19] proposed a new path planning method based on the physical size of the lunar rover and the DEM of the real environment, but the efficiency is limited due to its slow convergence speed. The limitations of the above methods make it difficult to be applied in lunar rover exploration with limited computing resources.

Compared with the above methods, the PRM is more efficient and practical because it converts the path planning in higher dimensional space into the topological space by constructing a roadmap. However, PRM is weak in the quality of the paths in the dense obstacle region, which is called the "narrow channel problem". Amato [20] proposed the Obstacle-Based PRM method to improve the sampling density in the narrow channels by sampling near the obstacle. David [21] proposed a bridge test method by testing whether the sample points are in the narrow channel, which improves the sampling density in the narrow channel. The above methods can only solve the insufficiency of sampling points in the narrow channel, yet the unreasonable distribution of sampling points remains unimproved. Raouf [22] proposed the Sequential Linear Paths (SLP) approach that divided

the working area into obstacle areas and open areas. In obstacle areas, a basic path planning algorithm such as PRM is used, while in open areas, the path is planned as a line segment to avoid wasting too many computing resources. However, the obstacle area cannot wrap the obstacle appropriately; in practice there are still some open areas within the obstacle area.

In addition, the lunar environment always varies during the lunar rover exploration due to the unstructured features, which cannot be represented in the original planning information (such as the safe zone and roadmap, etc.). Repeatedly solving the safe zone and constructing a roadmap for the varying environment would reduce the efficiency. To solve this problem, Speyerer [23] developed a probe task planning tool called R-Traveler, which realized the optimal energy consumption during task execution. Gao [9] proposed an automated activity planning method based on intelligent planning technology. While the above research improves the planning efficiency to some extent, the original planning information is not utilized sufficiently for new exploration tasks. To solve the abovementioned problems, this study will mainly focus on improving the autonomy and the efficiency of the detecting point selection and path planning for the lunar rover, especially in the unstructured and complex lunar environment. A detecting point autonomy selection strategy is proposed taking into consideration various influencing factors, such as topographic features, working requirements of detecting instruments, etc. Then, a sampling-based path planning method combined with linear guidance is proposed to minimize the consumed time. Finally, taking full advantage of the original planning information and eliminating the redundant content from newly supplemented information, a map extension strategy is designed to tackle the environmental changes.

This paper contains three main innovations:

- Construct the selection strategy for mission-oriented detecting points. The detecting point of the rover is solved by weighing the effect of slope and obstacles synthetically, and the requirements of detecting instruments are taken into consideration;
- Construct the sampling-based path planning method under linear guidance. Only the local roadmap near the obstacle is constructed, and the linear guidance is applied in the path planning; therefore, the efficiency and quality of path planning can be greatly improved;
- Design the map extension strategy to meet the requirements of extending and updating the roadmap brought by environmental uncertainty and multitasking.

The rest of this paper is organized as follows: Section 2 constructs the detecting point selection strategy for lunar rover missions. Section 3 describes the sampling-based path planning method under linear guidance, and the map extension method by using the local characteristics of the path planning in detail. Section 4 carries out a simulation to verify the correctness and effectiveness of the proposed strategy. Finally, conclusions are summarized.

2. Autonomous Selection of Detecting Points within the Safe Zone

High autonomy is essential for the exploration tasks like soil sampling, rock-mineral analysis, etc., so that the rover is capable of selecting the mission-oriented detecting point independently. In this paper, a strategy for selecting detecting points is proposed, which includes three steps: Firstly, determine the lunar rover's safe zone by considering the safety requirements. Then, establish an optimal indicator according to the characteristics of the detecting instruments, which is used to evaluate candidate detecting points. Lastly, a suitable point is selected autonomously within the safe zone by an optimization algorithm.

To simplify the description, a general model of the lunar rover is designed by reference to the structure of lunar rovers, the United States "Chariot", and the Chinese "Yutu". It is $3l$ meters long, $3l$ meters wide, $2l$ meters high, and M kilograms in weight (see Figure 1). In addition, the key symbols and abbreviations involved in this chapter are listed in Table 1:

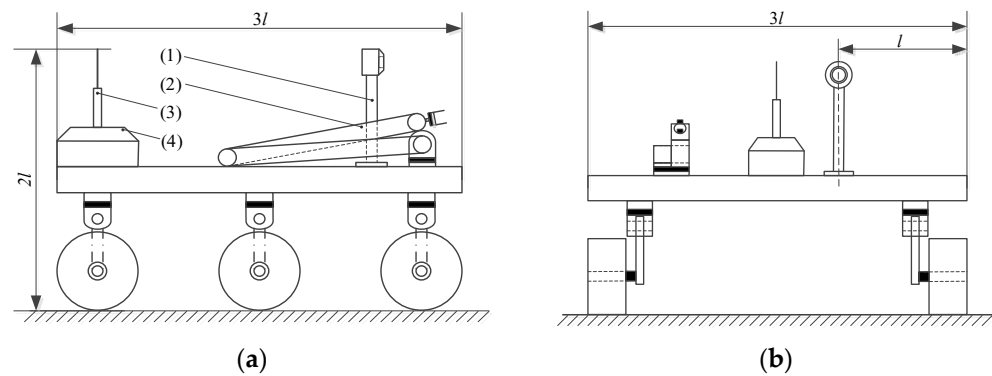


Figure 1. Structure model of the lunar rover. It carries (1) the front camera, (2) the vehicle-mounted manipulator, (3) the signal receiver, (4) the system controller. (a) The left view. (b) The main view.

Table 1. Key symbols and abbreviations in Section 2.

Symbol or Acronym	Paraphrase
safe zone	An area where the lunar rover can safely travel.
S_{dat}	The datum plane in generating the DEM which describes the real topography of the lunar surface.
R_{cur}	The current region where the rover is located.
S_{fit}	The reference plane which is fitted according to the digital elevation in R_{cur} .
S_{det}	The reference plane which is fitted according to the digital elevation in the detecting region.
f_{cli}	The slope-climbing cost function of the lunar rover, which is used to evaluate the traversability of a grid.
f_{obs}	The obstacle-crossing cost function of the lunar rover, which is used to evaluate the energy consumption for crossing over obstacles.
FEI	Flatness Evaluation Index, which indicates the flatness of a region around the detecting point. Generally, this region is smaller than a grid.

2.1. Solution of the Safe Zone

When the rover travels on the lunar surface, the slope and obstacles (i.e., rocks and soil pits) would cause more energy consumption or even rollovers. To ensure the rover's driving safety as well as avoid consuming excessive energy, we determine its safe zone by integrating the slope-climbing cost and the obstacle-crossing cost.

2.1.1. The Slope-Climbing Cost Function

Generally, the DEM of the lunar surface is provided by the satellites and the camera on the rover. It is represented by N elevation points containing 3D position information, and can be expressed as a 3D array:

$$\mathbf{P}_{\text{ele_total}} = \{\mathbf{P}_{\text{ele_i}}\} = \{x_i, y_i, z_i\}, i = 1, 2, \dots, N \quad (1)$$

where $\mathbf{P}_{\text{ele_i}}$ is the position vector of the i -th elevation point, N is the total number of elevation points, and N relates to the modeling accuracy of the DEM.

We mesh the DEM into grids, where the size of each grid is $l \times l$. Thus, according to the rover's physical structure (see Figure 2), the region R_{cur} is $3l \times 3l$.

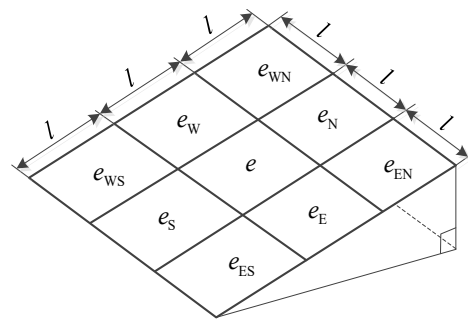


Figure 2. Schematic of the lunar rover’s current region R_{cur} .

Then we apply the least square method [24] to fit the plane S_{fit} , using all elevation point $P_{ele_3 \times 3} = \{x_j, y_j, z_j\}, j=1, 2, \dots, N_{cur}$ in region R_{cur} (where N_{cur} is the total number of elevation points in R_{cur}). That is to minimize F in the following plane equation:

$$F = \sum (Ax_j + By_j + C - z_j)^2 \tag{2}$$

where, A, B, C are undetermined coefficients which can be obtained by solving Equation (3).

$$\begin{cases} \frac{\partial F}{\partial A} = 0 \\ \frac{\partial F}{\partial B} = 0 \\ \frac{\partial F}{\partial C} = 0 \end{cases} \text{ so, } \begin{cases} A\sum x_j^2 + B\sum y_jx_j + C\sum x_j = \sum z_jx_j \\ A\sum x_jy_j + B\sum y_j^2 + C\sum y_j = \sum z_jy_j \\ A\sum x_j + B\sum y_j + Cn = \sum z_j \end{cases} \tag{3}$$

The spatial position of S_{fit} related to the datum plane S_{dat} is shown in Figure 3.

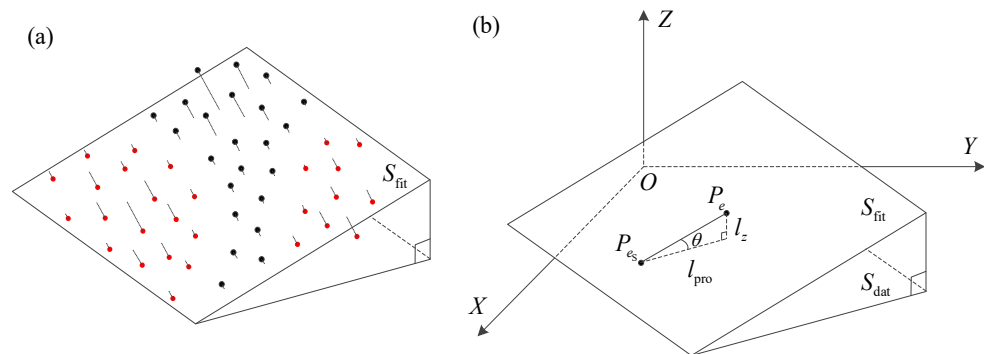


Figure 3. (a) Schematic of the elevation points relative to S_{fit} . (b) Schematic of the relative position of S_{fit} with respect to S_{dat} .

In Figure 3, $\theta \in [0^\circ, 90^\circ]$ is the included angle between S_{fit} and S_{dat} ; P_e and P_{es} have the coordinates (x_e, y_e, z_e) and (x_{es}, y_{es}, z_{es}) , respectively; l_z is the height difference between P_e and P_{es} ; l_{pro} is the distance between P_e and P_{es} projected to S_{dat} ; l_z and l_{pro} can be expressed as follows:

$$\begin{cases} l_z = |z_e - z_{es}| \\ l_{pro} = [(x_e - x_{es})^2 + (y_e - y_{es})^2]^{1/2} \end{cases} \tag{4}$$

Thus, the angle θ measuring the slope of grid e can be obtained by:

$$\theta = \arctan\left(\frac{l_z}{l_{pro}}\right) \tag{5}$$

The structure of the lunar rover determines that it is difficult to grip the lunar surface well. When the slope angle θ of the grid e is too large, the rover may overturn. Therefore, the range of θ should be constrained to guarantee the rover’s safety, which is set as $[0, \theta_{max}]$.

Since the energy consumption can affect the sustainable operation (i.e., service life) of the lunar rover, we use it to evaluate the traversability of a grid. According to the requirement of sustainable operation, the slope-climbing cost function of the lunar rover is defined as

$$f_{cli} = \begin{cases} Mg_{lun}l\sin\theta = Mg_{lun}l_z & (0 \leq \theta \leq \theta_{max}) \\ +\infty & (\theta > \theta_{max}) \end{cases} \quad (6)$$

where g_{lun} is the gravitational acceleration on the lunar surface. Using Equation (6), the energy consumption for overcoming gravity can be evaluated when the rover passes a grid.

2.1.2. Obstacle-Crossing Cost Function

In the process of fitting plane S_{fit} macroscopically, some small obstacles are eliminated, such as small rocks, small soil pits, etc. The energy cost to cross them cannot be ignored in real missions, but is not included in f_{di} . Therefore, an obstacle-crossing cost function needs to be considered in the mission analysis.

In this section, we construct the obstacle-crossing cost function based on the energy consumption to cross the maximum obstacle, which is characterized by the extreme height of each elevation point relative to S_{fit} in the grid e .

The distance l_{ele_j} of any elevation point $P_{ele_3 \times 3_j} = (x_j, y_j, z_j)$ in $P_{ele_3 \times 3}$ moving along the normal direction to the plane S_{fit} can be obtained by

$$l_{ele_j} = \frac{(Ax_j + By_j - z_j + C)}{\sqrt{A^2 + B^2 + 1}} \quad (7)$$

By combining the distances $l_{ele_j}, j = 1, 2, \dots, N_{cur}$ corresponding to all elevation points, a set I_{ele} is obtained. The maximum height l_{obs} of the obstacles in the plane can be given by

$$l_{obs} = \max\{I_{ele}\} - \min\{I_{ele}\} \geq 0 \quad (8)$$

where $\max\{I_{ele}\}$ and $\min\{I_{ele}\}$ are the maximal and the minimal in set I_{ele} , respectively.

The highest obstacle that can be crossed is defined as l_{max} , which is related to the rover's structure. If the obstacle's height l_{obs} is greater than the maximum surmountable obstacle height l_{max} , the lunar rover cannot pass through grid e . A safe pass can only occur when $l_{obs} \leq l_{max}$. The obstacle-crossing cost function of the lunar rover can be expressed as:

$$f_{obs} = \begin{cases} Mg_{lun}l_{obs} & (l_{obs} \leq l_{max}) \\ +\infty & (l_{obs} > l_{max}) \end{cases} \quad (9)$$

2.1.3. Solution of the Safe Zone

Based on the slope-climbing cost function and the obstacle-crossing cost function, an evaluation function f_{gri} can be constructed, which represents the cost of crossing a grid, given by

$$f_{gri} = \begin{cases} f_{cli} + f_{obs} & (\theta_{min} \leq \theta \leq \theta_{max} \text{ and } l_{obs} \leq l_{max}) \\ +\infty & (\text{else}) \end{cases} \quad (10)$$

According to Equation (10), whether a specific grid is passable by the rover can be determined. For example, the black grids in Figure 4a are not passable, where the value f_{gri} is calculated as $+\infty$. On the contrary, white grids are passable where f_{gri} is not infinite values. However, some white grids only determined by Equation (10) are still unreachable for the rover because they are obstructed by their surrounding black grids. We call them unreachable areas—grey regions in Figure 4b.

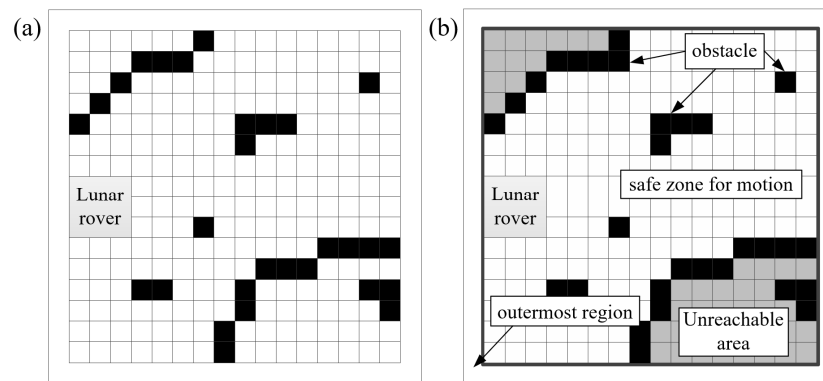


Figure 4. Schematic of the S_{fit} of the lunar rover. (a) The passable region by white grids. (b) The constituted actual safe zone.

To obtain the safe zone, we introduce the seed-filling method to exclude the unreachable area from all the white grids calculated by Equation (10). Specifically, the location of the lunar rover is chosen as the seed; then, we apply the breadth-first search algorithm [25], one of the common methods to find the largest connected domain, to search the grids adjacent to the seed. Grids with the same property are merged into one set, and the safe zone can be determined. Note that, due to the structural constraints of the lunar rover, the passing cost of the outermost region (see Figure 4b) cannot be calculated since it is not considered a safe zone.

At this point, the autonomous solution of the safe zone has been obtained.

2.2. The Selection Strategy of the Detecting Point

In this section, we propose an automatic strategy to select the mission-oriented detecting point for the rover. In general, a flat detecting region is beneficial to the detecting instruments. Therefore, we introduce the FEI to evaluate the flatness of a detecting region, and then we use the particle swarm optimization (PSO) algorithm to search for a flat region as the candidate detecting point.

2.2.1. The FEI

For missions like lunar soil sampling and rock mineral analysis, the X-ray mass spectrometer is the main equipment, and two necessary conditions should be met for the normal operation [26]:

- Detecting orientation. When the X-ray mass spectrometer analyzes soil on the Moon surface, it needs to receive the reflected light of the X-ray emitted by itself. Therefore, its axis should coincide with the normal to the fitted plane corresponding to the detecting point.
- Detecting distance. The distance between the mirror surface of the X-ray mass spectrometer and the detecting point should be smaller than the maximum exploration range. However, some features of the detecting region, such as convex and concave, may lead to volatility in distance. To meet the distance requirement, the detecting point should have high flatness.

The required detecting orientation can be achieved by adjusting the pose of the vehicle-mounted end effector, on which the X-ray mass spectrometer is installed. Therefore, this paper merely focuses on the detecting distance requirement.

The flatness of a region can be measured based on the DEM because it includes detailed topographic information. The standard deviation is a measure of the fluctuation in height of elevation points, and that of a flat region is usually small. Thus, the standard deviation is taken as the FEI to estimate the flatness.

Assume that point $P_{arb}(x, y, z)$ is one of the candidate detecting points, then the detecting region is the circular region with a radius r at P_{arb} , where r is determined by the

mirror radius of the X-ray mass spectrometer. This region contains N_{arb} elevation points, which make up the set $P_{\text{ele_Xray}} = \{x_k, y_k, z_k\}$, ($k = 1, 2, \dots, N_{\text{arb}}$) (see Figure 5).

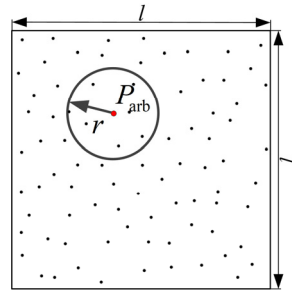


Figure 5. The planform of the elevation points near point P_{arb} in a grid.

The N_{arb} elevation points are used to fit the plane S_{det} as stated in Section 2.1.1, and the distance l_k from the points $P_{\text{ele_Xray}_k}(x_k, y_k, z_k)$, ($k = 1, 2, \dots, N_{\text{arb}}$) in the set $P_{\text{ele_Xray}}$ to S_{det} is calculated by Equation (7). The FEI σ is defined by the standard deviation of l_k ($k = 1, 2, \dots, N_{\text{arb}}$)

$$\sigma = \sqrt{\frac{1}{N_{\text{arb}}} \sum_{k=1}^{N_{\text{arb}}} (l_k - \bar{l}_k)^2} \quad (11)$$

where, \bar{l}_k is the mean of all the distances l_k ($k = 1, 2, \dots, N_{\text{arb}}$).

2.2.2. Selection of the Detecting Point

A suitable detecting point should be selected in the region which meets the condition of detecting distance in the safe zone. For a large scale of the lunar environment, in which the safe zone is usually in several thousand square meters, it requires a heavy lift in time and resources for manual work. To overcome those shortcomings, the PSO is applied to select the detecting point autonomously in this paper, which stands out with fast convergence and high precision [27].

Randomly select N_{par} elevation points in the safe zone, denoted as $P_{\text{ele_par}} = \{x_n, y_n, z_n\}$, ($n = 1, 2, \dots, N_{\text{par}}$). N_{par} can be determined according to the size of the safe zone. The first two dimensions $P_n = (x_n, y_n)$, ($n = 1, 2, \dots, N_{\text{par}}$) of the set $P_{\text{ele_par}}$ are extracted to initialize the particle swarm. The fitness function $f(P_n)$ of the particle is constructed according to the FEI which is shown as Equation (12):

$$f(P_n) = \frac{1}{k \cdot \sigma(P_{\text{ele_par}}) + 1} \quad (12)$$

where the value range of $f(P_n)$ is $(0, 1]$; k is an amplification factor. The high value of $f(P_n)$ means that the area where the elevation points of particles are located is relatively flat.

According to the requirements of the flatness of the detecting point, the fitness threshold of the particle G_{best} can be set, and the fitness of the final optimum particle needs to be greater than or equal to G_{best} . The upper limit of the iteration K_{max} is set according to the requirement of calculation efficiency.

So far, the best mission-oriented detecting point can be selected. The entire process of the detecting point selection strategy is shown in Figure 6.

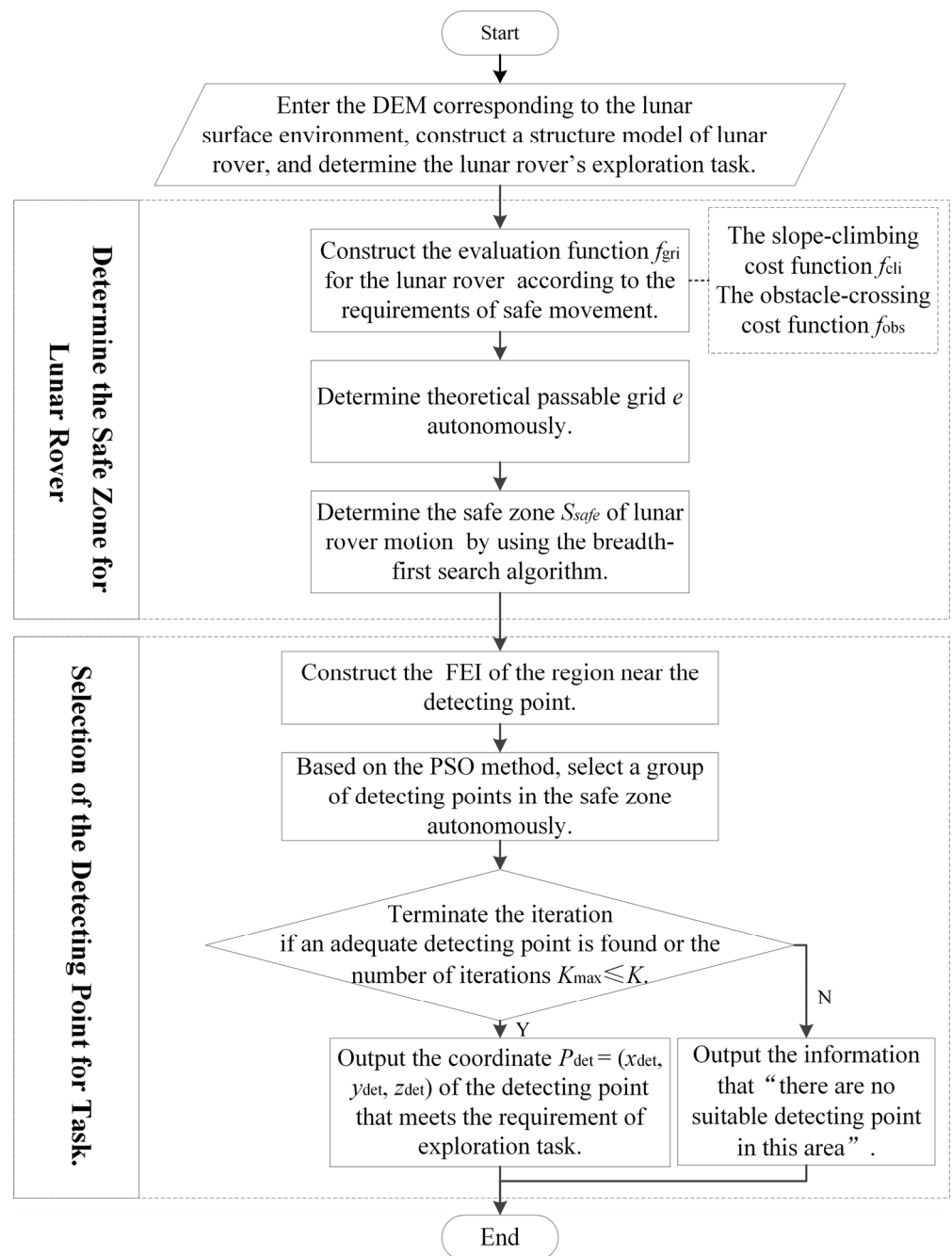


Figure 6. The flow of detecting point selection of lunar rover.

3. The Sampling-Based Path Planning Method under Linear Guidance (The Integrated Sampling-Based and Linear-Guided Path Planning Method)

Sampling-based global path planning algorithms such as PRM and visibility graph method are efficient in path planning in a static working environment [28]. These algorithms firstly construct roadmaps, then search the path quickly with the roadmaps to obtain feasible paths. However, many sampling points and connecting paths fall in open areas far from obstacles. These points have no obvious effect on improving the connectivity of the roadmap.

In this section, a sampling-based path planning under the linear guidance method is proposed, taking advantage of the higher efficiency of the linear path planning in open areas. It includes three steps: First, divide the safe zone into open areas and dense obstacle areas, and the dense obstacle areas are sampled to construct local roadmaps. Second,

the initial path is constructed based on the linear path planning, which is then modified and simplified to avoid colliding with obstacles according to the sampled local roadmaps. Finally, the original map and several supplemented maps for new tasks are combined by the map extension strategy, which fully utilizes the local roadmap to improve the efficiency of roadmap reconstruction.

3.1. Sampling in Areas with Dense Obstacles

3.1.1. Division of the Safe Zone

To divide the safe zone clearly, the area within ε (ε is determined by the grid size) around the obstacle is defined as the dense obstacle area. The detailed division process of the open area and the dense obstacle area is introduced as follows:

In a workplace C_{work} containing obstacles, the center of grid (i, j) is denoted as $A(x_{i,j}, y_{i,j})$. Take A as the center (see Figure 7) and search for the nearest obstacle grid as follows:

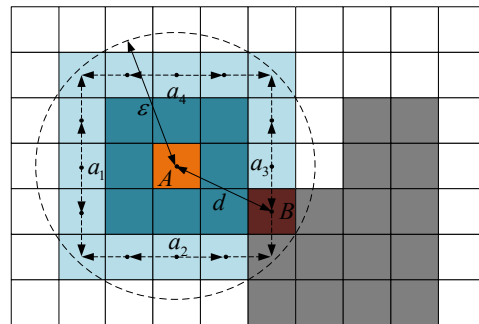


Figure 7. Schematic of the searching order of grids within one layer.

1. First, search all grids in the n -th ($n = 1, 2, 3, \dots, N$) layer that encloses A . The maximum number of the layer $N = \lceil \varepsilon/l \rceil$ ($\lceil \cdot \rceil$ indicates an upward rounding).
2. During the search of grids in each layer, the four grids in the horizontal and vertical directions are searched preferentially, and are shown as a_1, a_2, a_3, a_4 in Figure 7. If neither of the searched grids is the obstacle grid, the upward and downward adjacent grids of a_1 and a_3 are searched, respectively. Similarly, the left and right adjacent grids of a_2 and a_4 are searched, respectively.

The first obstacle grid detected is the nearest obstacle grid, for example, the dark red grid centred at point B in Figure 7. The distance $d(A, B)$ can be calculated between the nearest obstacle grid's center $B(x_{\text{obs}}, y_{\text{obs}})$ and $A(x_{i,j}, y_{i,j})$:

$$d(A, B) = \sqrt{(x_{i,j} - x_{\text{obs}})^2 + (y_{i,j} - y_{\text{obs}})^2} \quad (13)$$

If no obstacle grid is detected by the end of the search, the grid does not belong to the dense obstacle area, and then $d(A, B)$ is set as ∞ .

According to the value of $d(A, B)$, the dense obstacle area and the open area can be expressed as a set, given by

$$D_{\text{dense}} = \{A | d(A, B) \leq \varepsilon\} \quad (14)$$

$$D_{\text{open}} = \{A | d(A, B) = \infty\} \quad (15)$$

3.1.2. Sampling Points in the Areas with Dense Obstacles

The number and distribution of the sampling points directly affect the efficiency of path planning and the connectivity of the roadmap. A large number and a worse distribution can increase not only the complexity of the search problem, but also the computation time. However, fewer sampling points, although costing less computational time, might fail

concerning a feasible path. If each sampling point includes rich information of obstacles, the total number of sampling points can be reduced. In this paper, the shape points and the distance points are used to characterize the contour features of the obstacles.

For the obstacles with the same size, the complexity of the shape determines the number of shape points. Prominent parts of the obstacle depict the effective contour of the obstacle and contain rich connectivity information. As shown in Figure 8a, parts surrounded by dashed lines can improve the connectivity of the roadmap. Considering the geometric feature, we sample points in the prominent part of obstacles, which can be represented by partial grids (see the orange grids in Figure 8b).

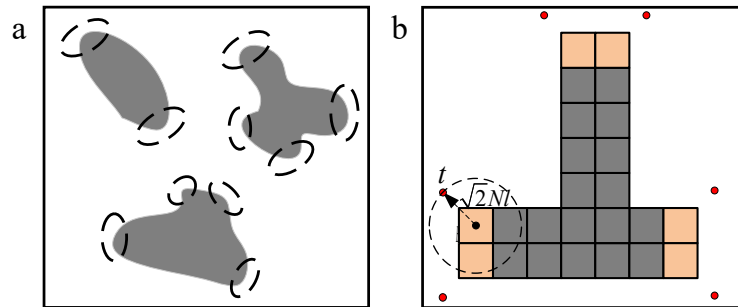


Figure 8. (a) The prominent part of the obstacle. (b) Sampling of shape points.

According to Section 3.1.1, a grid $t \in D_{dense}$ in the dense obstacle area can be ensured, whose center locates on the diagonal of the nearest obstacle grid B . The distance between the center of t and that of B satisfies the following relationship

$$d(t, B) = \sqrt{2}Nl, N = \lceil \varepsilon/l \rceil \tag{16}$$

The center of the grid t is chosen as the shape point because it can represent the bulged part of the obstacle to some extent. The above process corresponds to the line 03~08 in Algorithm 1. We define the set of shape points as follows:

$$T_1 = \left\{ t \mid d(t, B) = \sqrt{2}Nl, t \in D_{dense} \right\} \tag{17}$$

The number and distribution of shape points are the same for obstacles with similar shapes but different areas. As the area increases, the distance between shape points increases, and the connected paths in the roadmap are lengthened, as shown in Figure 9a.

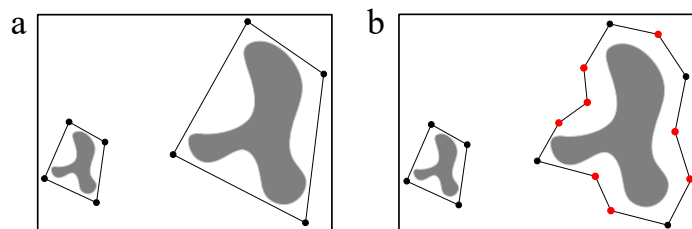


Figure 9. Schematic of connected paths and sampling points. (a) The connected paths in the roadmap are constructed only with shape points. (b) The connected paths in the roadmap are constructed with shape points and distance points.

However, in the long narrow channels, the shape points do not enhance the connectivity of the roadmap significantly. To solve this problem, distance points are introduced in the paper; see the red points in Figure 9b. The set composed of the distance points t_2 is denoted as T_2 , on which the following constraints are imposed:

- $t_2 \in \{D_{dense} \setminus T_1\}$;

- $d(t_2, t_1) > \varepsilon$ for $\forall t_1 \in T_1$;
- $d(t_2, t'_2) > \varepsilon$ for $\forall t'_2 \in T_2$ and $t_2 \neq t'_2$.

To meet the above three conditions simultaneously, the distance points are selected by traversing all the elements in T_1 as follows (the lines 09~15 in Algorithm 1):

- (1) Determine the annular area with radius $\varepsilon \sim 2\varepsilon$ and center $t_x \in T_1$ (see Figure 10).
- (2) Search the center of grid $z_i \notin T_1 \cup T_2$ in the overlap between the annular area and the dense obstacle area. If no center point is detected within the overlap, turn to (1).
- (3) Calculate the Euclidean distance $d(t_x, z_i)$ between z_i and t_x . Notice that z_i may not be unique, in which corresponding to the minimum distance $\min d(t_x, z_i)$ is taken as a distance point t_y as follows

$$t_y = \operatorname{argmin}\{\varepsilon \leq d(t_x, z_i) \leq 2\varepsilon\} \quad (18)$$

Then, put the point t_y into the set T_2 .

- (4) Take t_y as the center of the annular area with radius $\varepsilon \sim 2\varepsilon$; turn to (2).

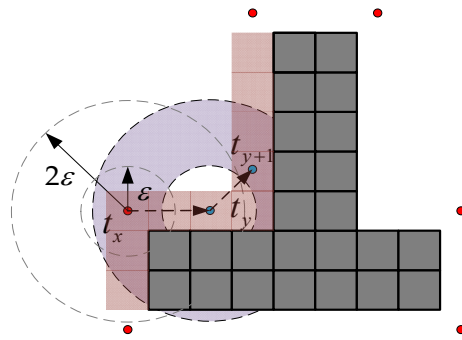


Figure 10. Schematic of sampling of the distance points.

Algorithm 1: Sampling Points

```

01   Require: Initialize  $D_{dense}$ ,  $t_1 = (0,0)$ ,  $t_2 = (0,0)$ ,  $l$  is the side length of grid,  $\varepsilon$  is the
    threshold of the area with dense obstacles.
02   if  $D_{dense} \neq \phi$  do
03     for each the center of the grid  $A \in D_{dense}$ 
04       find out the nearest obstacle  $B$  grid to  $A$ 
05       if  $d(A, B) = \sqrt{2}Nl$ ,  $N = \lceil \varepsilon/l \rceil$ 
06          $T_1 \leftarrow A$ 
07       end
08     end
09     for each  $t_1 \in T_1$ 
10       for each  $t_2 \in D_{dense} \setminus T_1 \cap D_{dense} \setminus T_2$ 
11         if  $\varepsilon < d(t_1, t_2) < 2\varepsilon$  do
12            $T_2 \leftarrow t_2$ 
13         end
14       end
15     end
16   end

```

Based on the obtained sampling points, the local roadmap can be constructed. All the sampling points in a circle with centre t_i and radius r (r can be determined according to the connecting requirements of the roadmap for path planning) are connected to the point t_i ($t_i \in T_1 \cup T_2$) to get N_r edges. The edges which do not collide with obstacles are stored to the set E . The roadmap G is made up of the set of sampling points $T_1 \cup T_2$ and the set of collision-free edges E . By constructing a roadmap, the path planning in continuous

space can be solved in the topological space; thus, the search complexity of path planning is reduced.

3.2. Path Planning under Linear Guidance

There is no connecting path in the open area since all the sampling points are located in the dense obstacle area, and the connecting paths of the roadmap are distributed around the obstacle. This section adopts a linear path in the open area as the path guidance.

3.2.1. Planning the Linear Path

Considering the generality, we assume that there are some arbitrary obstacles in the map and specify the rover's current location and detecting point. It starts by establishing a linear path between the two points:

$$y = \frac{y_{\text{current}} - y_{\text{detecting}}}{x_{\text{current}} - x_{\text{detecting}}}(x - x_{\text{current}}) + y_{\text{current}} \quad (19)$$

where, $(x_{\text{current}}, y_{\text{current}})$ is the coordinates of the current location and $(x_{\text{detecting}}, y_{\text{detecting}})$ is the coordinates of the detecting point.

If no obstacle intersects with the linear path, this linear path is the final path, as A_1B_1 in Figure 11. Otherwise, it is necessary to find out the feasible paths that can bypass obstacles. The intersecting points with the edge of obstacles are denoted as $(a_1, b_1, a_2, b_2, \dots, a_n, b_n)$ (see Figure 11).

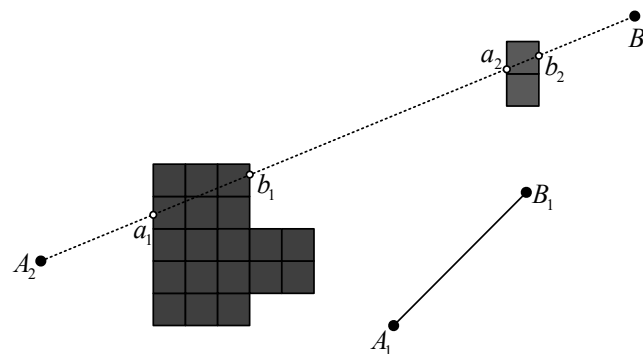


Figure 11. Schematic of a linear path.

3.2.2. Searching for Local Feasible Paths in the Roadmap of Dense Obstacle Areas

If the linear path intersects with obstacles, the A* heuristic search algorithm is adopted to search the local feasible path in the roadmap G constructed in Section 3.1. The local feasible path can help to bypass the obstacle. The sampling points nearest to a_1 and b_1 are used as the starting and the ending point to search the path. The maximum distance between each sampling point and the obstacle is strictly required to be ε in Section 3.1; there must be one or two sampling points t that meet $d(t, a_1) < 2\varepsilon$. Therefore, the search range for the nearest sampling point of a_1 , which is denoted as C_{a_i} , is the circle with center a_1 and radius 2ε . The nearest sampling point searched as a'_1 is recorded, and so is b'_1 . The local feasible path is $a'_1 - > x_1 - > x_2 - > b'_1$ (see Figure 12), and this process corresponds to line 07~22 in Algorithm 2.

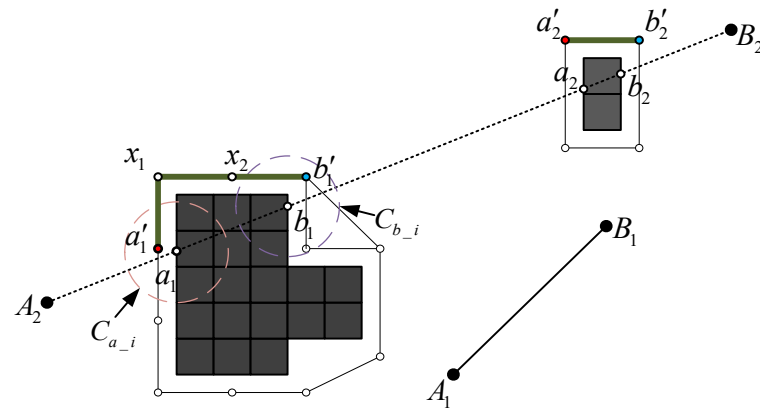


Figure 12. Schematic of the local feasible path in areas with dense obstacles.

3.2.3. Solution of the Best Feasible Path

An obstacle-free path can be obtained by modifying the linear path with local feasible paths. However, it is rather tortuous. In the paper, a forward trial method is used to enhance the obstacle-free path by linearizing it.

Take A_2B_2 for example to explain this method.

Step1: Take A_2 as the starting point; connect with the next node a_1' . Try to connect A_2 with x_1 ; if A_2x_1 does not intersect with any obstacle, a_1' is discarded, and so on. Get the first furthest point x_1 and the first valid shortcut A_2x_1 .

Step2: Take x_1 as the new starting point; then repeat Step 1. Get the second furthest key point a_2' and the second valid shortcut x_1a_2' .

Step3: Take a_2' as the new starting point; then repeat Step 1. Get the third valid shortcut $a_2'B_2$.

So far, the final feasible path $A_2 \rightarrow x_2 \rightarrow a_2' \rightarrow B_2$ has fewer turn points and is the best possible shortcut obtained (see Figure 13), and this process corresponds to line 24~34 in Algorithm 2.

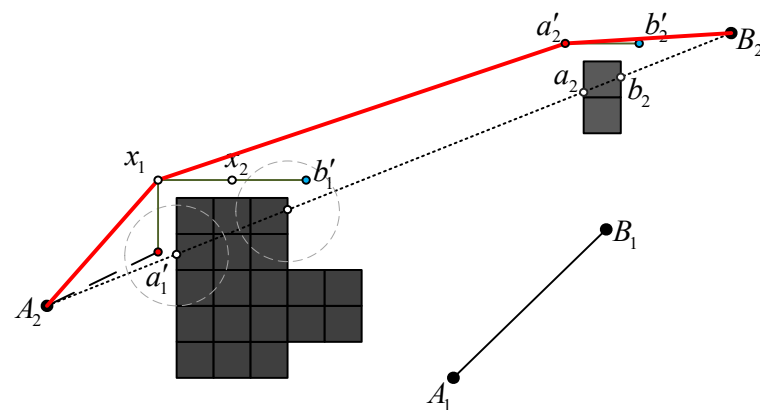


Figure 13. Schematic of the final feasible path.

3.3. Map Extension

With the deeper exploration of the lunar rover, the explorable area continues to expand. The detection of a new area or the change of a local area can render the existing roadmap unusable, so that the roadmap needs to be updated. Reconstructing a large-scale and complete roadmap of the lunar environment would take a long time, which would lead to a sharp decrease in planning efficiency. To improve the efficiency, map extension is carried out combined with the roadmap's local feature.

Algorithm 2: Path Planning Strategy under Linear Guidance

```

01   Require: Specify the current location of the rover  $(x_{\text{current}}, y_{\text{current}})$  and the detecting
    point  $(x_{\text{detecting}}, y_{\text{detecting}})$ .
02   Initialize the roadmap  $G$ , the obstacle areas  $O$ , the area with dense obstacles  $S$ , the
    set of the key points  $T$ .
03   Initialize the finite set of the points of local path  $P = []$  and the set  $\text{final\_path} = []$ .
04   Plan a linear path from  $(x_{\text{current}}, y_{\text{current}})$  to  $(x_{\text{detecting}}, y_{\text{detecting}})$ .
05   Find intersection points of the straight line and the  $O$ :
06    $P_{\text{its}} = (a_1, b_1, a_2, b_2, \dots, a_n, b_n)$ 
07   if  $P_{\text{its}}$  is not NULL do
08     for  $i = 1$  to  $n$ 
09       Take all elements of  $T$  in the search range  $C_{a_i}$  as the set  $T_{a_i}$ .
10       for each element  $t_i$  of  $T_{a_i}$ 
11         Calculate the distance  $d(t_i, a_i)$  between  $t_i$  and  $a_i$ .
12       end
13       if  $t_i$  correspond to the minimum of  $d(t_i, a_i)$ 
14          $a'_i = t_j$ 
15       end
16       Take all elements of  $T$  in the search range  $C_{b_i}$  as the set  $T_{b_i}$ , and
    the same process as  $T_{a_i}$ .
17       Search the path  $p_i$  from  $a'_i$  to  $b'_i$  in  $G$  by using A* algorithm.
18       for each node  $(x_{\text{node}}, y_{\text{node}})$  of the  $p_i$ 
19          $P \leftarrow (x_{\text{node}}, y_{\text{node}})$ 
20       end
21     end
22   end
23    $\text{final\_path} \leftarrow (x_{\text{current}}, y_{\text{current}})$ 
24   for  $j = 2$  to  $\text{length}[P]$ 
25     for  $i = j$  to  $\text{length}[P]$ 
26        $x_j = P[j]$ 
27        $x_i = P[i]$ 
28       if the straight path  $x_j x_i$  intersects with obstacle do
29          $\text{final\_path} \leftarrow x_{i-1}$ 
30          $j = i - 1$ 
31       end
32     end
33   end
34    $\text{final\_path} \leftarrow (x_{\text{detecting}}, y_{\text{detecting}})$ 

```

3.3.1. Data Processing of Map Extension

The following process needs to be done to execute the map extension:

- Map preprocessing. The whole map Z is the same as the initial map at the beginning.
- Roadmap construction. According to Equation (10), the initial safe zone for motion is determined. The set of sampling points T_0 is got and the roadmap G_0 is constructed by Section 3.1.
- Set collections T_{set} and G_{set} , the sampling points set T and the roadmap G of the initial map or the individual map, respectively.

The part that needs to be updated is treated as an individual map, and the safe zone S_{new} , the sampling points set T_{new} , and the roadmap G_{new} of it are obtained according to Sections 2.1 and 3.1. According to the relative position R between the individual map coordinate origin and the whole map origin, the coordinates of each point are updated. For example, if the position of the key point in the individual map is R_T , then its position R'_T in the whole map can be updated as follows:

$$R'_T = R + R_T \quad (20)$$

Store the updated sampling points set T_{new} , the roadmap G_{new} in T_{set} and G_{set} , respectively.

The process of path planning for the extended map is the same as that for the initial map:

1. Plan the initial path with a linear line in the whole map.
2. Once the initial path intersects with the obstacles, search for a local feasible path using the A* algorithm in the roadmap to bypass the obstacle. Note that, if the obstacle locates in the individual map, the local feasible path will be searched in the roadmap corresponding to the individual map.
3. Use the forward trial method to get the final feasible path.

For example, the initial path is planned from point A to point B as shown in Figure 14, and it intersects with two obstacles. One is in the individual map and the other is in the initial map. The local feasible paths are searched in the roadmap, respectively, as red lines, and the final feasible path is obtained as blue lines.

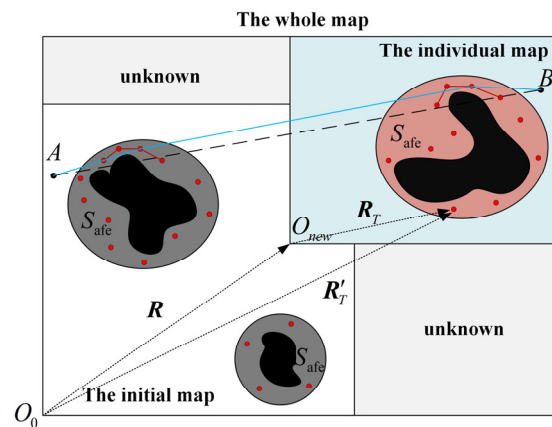


Figure 14. Schematic of the map extension.

Assuming that there are m maps of size $n \times n$ without overlapping each other, and they can form a whole map of size $m \times n \times n$. The time complexity is $O((P)^2 \log P)$ by using the A* algorithm to search a local path in the roadmap of size P , which is constructed from the whole map directly. Taking one of the maps as an initial map and the rest of them as individual maps, the time complexity is $O(m(\frac{P}{m})^2 \log \frac{P}{m})$ with the map extension. As the search scope is narrowed with map extension to some extent, the map extension strategy can improve the planning efficiency further by integrating the individual map with the initial map.

3.3.2. Path Planning with Map Extension

The process of path planning with map extension is shown in Figure 15:

Step 1: Input the initial map and the initial location V_{start} of the lunar rover.

Step 2: Determine the safe zone S_{safe} based on the slope-climbing cost function Equation (6) and obstacle-crossing cost function Equation (9).

Step 3: Sample the sampling points T and construct the roadmap G according to the strategy proposed in Section 3.1.

Step 4: If S_{safe} , T , G are extracted from the initial map, S_{safe} will be used to initialize the safe zone S_{whole} of the whole map, and two collections T_{set} and G_{set} are constructed to store T and G . Otherwise, S_{safe} , T , G will be appended to S_{whole} , T_{set} , G_{set} after being updated according to the relative position R of the individual map in the whole map.

Step 5: Determine whether to add a new individual map. If required, input the new individual map and the relative position R in the whole map, then turn to step 2. Otherwise, turn to step 6.

Step 6: Select the detecting point V_{goal} in the safe zone S_{whole} of the whole map based on the strategy of the selection of detecting point proposed in Section 2.2.

Step 7: Obtain the final feasible path using the sampling-based path planning method under linear guidance.

Step 8: Output the path planning result, and wait for the command of the next exploration mission. If the command is received, turn to Step 5.

Based on the above process, a path planning method with map extension can be executed.

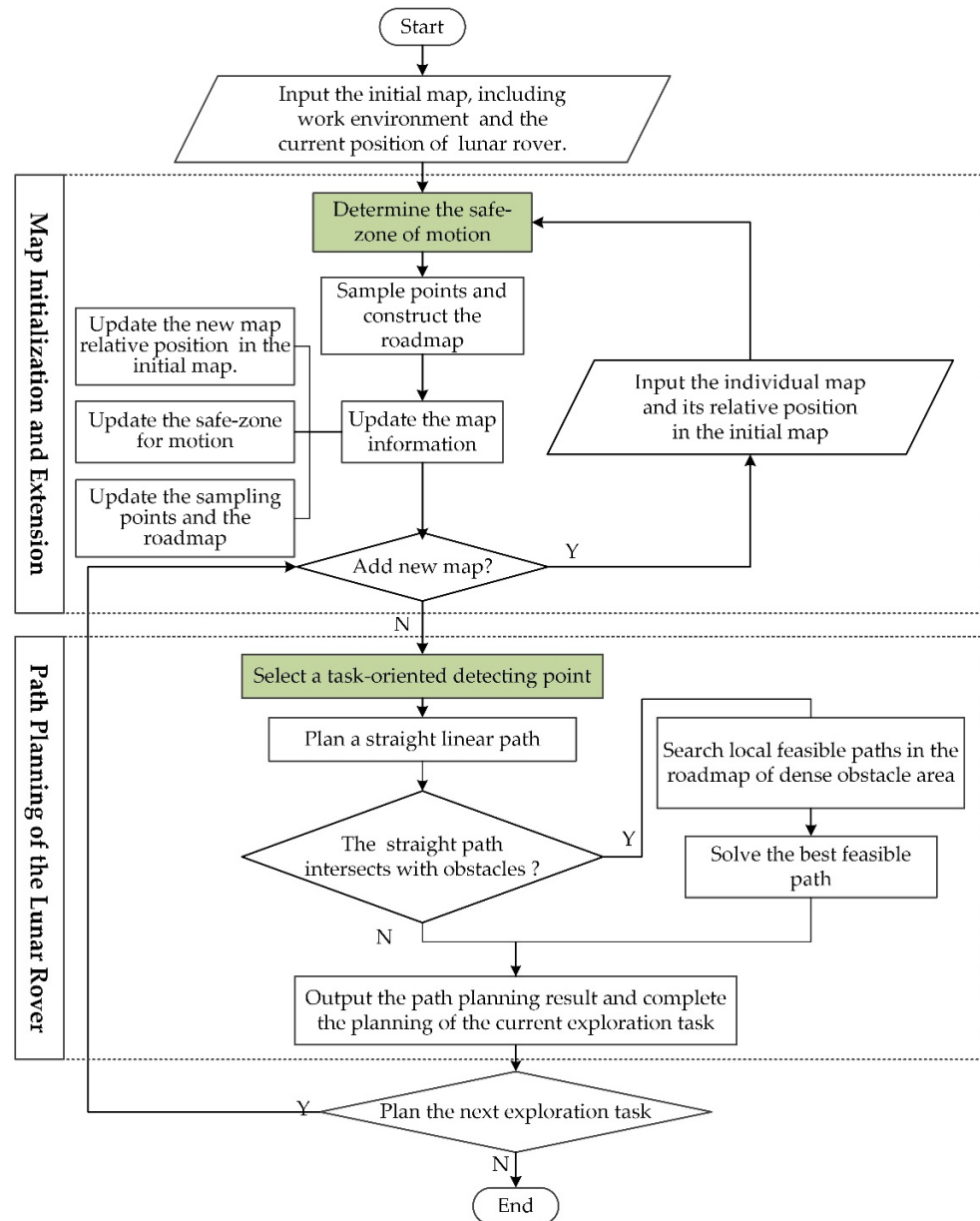


Figure 15. The flow of path planning with map extension.

4. Simulation

Simulations were executed based on the DEM of the lunar surface environment with 16 m length and 14.5 m width (see Figure 16). This DEM totally contains 687,969 elevation points to describe detailed features of the lunar environment.

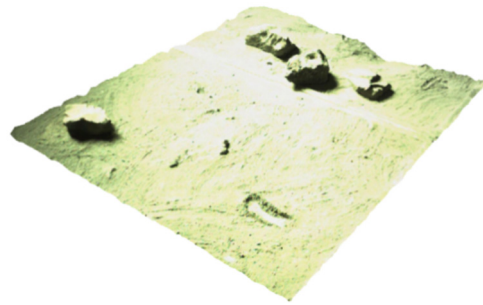


Figure 16. The 3D model of lunar surface environment.

4.1. Verification of Detecting Point Selection Strategy

The DEM was rasterized to select the detecting points firstly. The length and width of the lunar rover were set as $3l = 1.5$ m. The lunar surface environment was rasterized to 928 grids with 0.5 m interval, 29 rows, and 32 columns (see Figure 7). Thus, the vertical projection of the rover on the ground was equivalent to nine grids, where the length of each grid $l = 0.5$ m.

4.1.1. Autonomous Solution for the Safe Zone

The total cost for each grid was calculated according to Equation (10) to get passable grids, as the white grids in Figure 17a. By searching these white grids based on the breadth-first searching, the safe zone was obtained, as the white grids in Figure 17b.

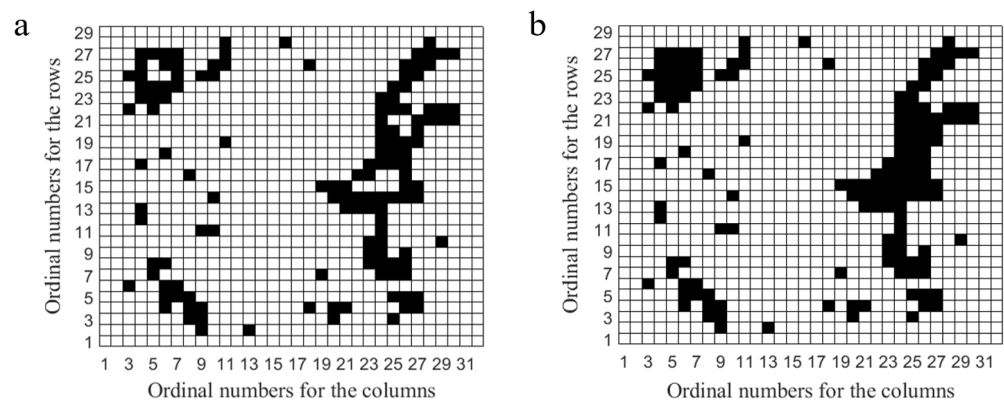


Figure 17. (a) Passable grids are shown with white grids. (b) The safe zone is shown with white grids.

Note that, since the grids near the edge of the map cannot be evaluated by Equation (10), they did not participate in the evaluation.

4.1.2. Autonomous Selection for the Detecting Points

Based on Equation (11), the flatness of the whole area was solved by traversing each point in the safe zone, which lasted 2141.521 s, on the 64-bit Windows 7 operating system with Intel(R) Core (TM) i5-3470 CPU at 3.20 GHz and RAM of 8 GB. The gray image for the flatness of the safe zone is shown in Figure 18, in which the regions with light color are relatively flatter regions.

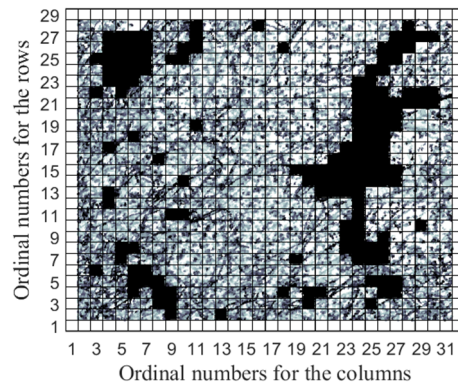


Figure 18. The gray level map for the flatness of the safe zone.

Twenty elevation points were randomly selected in the safe zone, and their x and y coordinate values were encoded to get the initial particle swarm. According to task requirements, the particle fitness threshold was set as $G_{best} = 0.95$ and the speed variation range was set as $[-2, 2] \text{ ms}^{-1}$. Based on the PSO, the detecting point was selected after nine iterations (the maximal number of iterations was set as $K_{max} = 50$), which lasted 12.343 s. The iterative search processes of detecting points are shown in Figure 19a, and the corresponding particle positions and the fitness changing process are shown in Table 2.

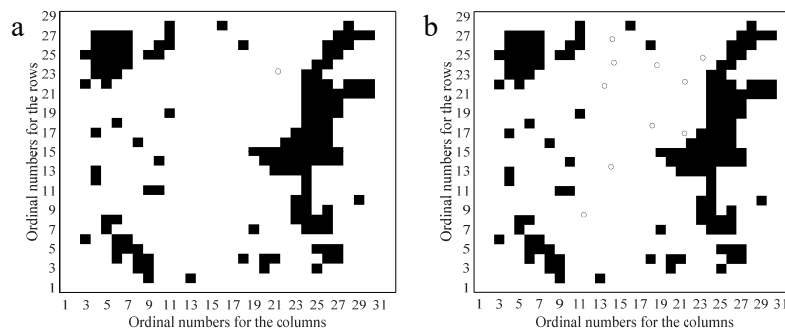


Figure 19. Detecting point selection based on PSO. (a) Single detecting point selecting process. (b) Multiple detecting point selecting results. Note that, to display the location of the detecting point clearly, grid lines are omitted.

Table 2. The particle position and fitness for each iteration in the search process.

Ordinal Number of Iterations	x [m]	y [m]	Fitness
1	11.205	20.518	0.7587
2	26.648	28.179	0.8386
3	30.581	1.713	0.8425
4	26.688	27.819	0.8477
5	5.944	28.221	0.9374
6	5.901	28.217	0.9390
7	5.885	28.201	0.9430
8	5.885	28.201	0.9430
9	21.296	23.311	1.0000

Based on PSO, several searched detecting points were shown in Figure 19b. The ordinal number for the searches (ONS), the number for the iterations (NI), the computing time consumption (CTC), the fitness of the initially selected detecting point (FISEP), and the fitness of the final selected detecting point (FFSEP) corresponding to each search are shown in Table 3.

Table 3. Search information of detecting points for 10 random searches based on particle swarm optimization.

ONS	NI	CTC/(s)	FISEP	FFSEP
1	6	7.153	0.7877	0.9798
2	6	6.617	0.7766	0.9611
3	14	11.332	0.7729	1.0000
4	17	13.039	0.7671	1.0000
5	46	29.827	0.7989	0.9711
6	5	6.229	0.8167	0.9771
7	11	10.310	0.7977	1.0000
8	26	20.434	0.8300	1.0000
9	23	17.916	0.7947	0.9883
10	5	6.509	0.4505	1.0000

From Table 3, we can see that FFSEPs from the above 10 random searches are all greater than G_{best} . This means all the final searched detecting points locate in flatter areas than the initial position, and meet the requirements of the detecting instrument. Besides, the average value of time consumption for the above 10 random searches is 12.937 s. This is two orders of magnitude lower than the time consumption with traversal solution 2141.521 s. We also notice a loose linear relationship between NI and CTC; that is, the increment of CTC is gradually less than the linear increment of NI. This shows that, compared with the linear time consumption traversal search, our strategy, which introduces the particle guidance mechanism from PSO, effectively speeds up the search for detecting points.

4.2. Verification of the Sampling-Based Path Planning Method under Linear Guidance

4.2.1. The Sampling Method of Points in the Dense Obstacle Region

The 23 obstacles in the environment were numbered (see Figure 20).

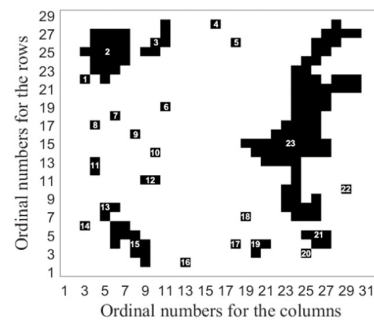


Figure 20. The map with numbered obstacles.

The obstacles No. 1–23 were sequentially added to the map, and the number of sampling points with the increase of the obstacle was obtained (see Figure 21).

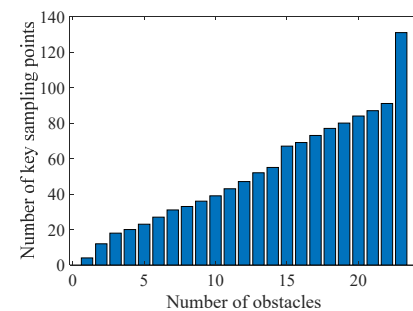


Figure 21. The number of sampling point changes with the increase of obstacle.

It can be seen that obstacles with large area or irregular shape will lead to a significant increase in the number of sampling points, especially obstacle No. 2, 15, and 23. When all obstacles were added to the map, a total of 133 sampling points were obtained (see Figure 22).

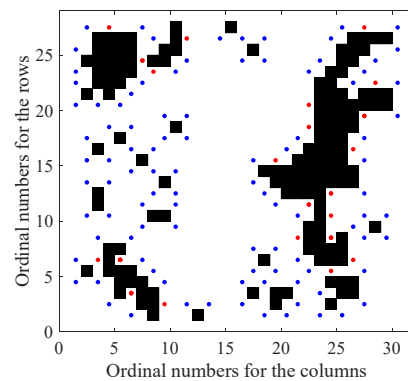


Figure 22. Distribution of key sampling points.

Shape points (blue dots in Figure 22) are distributed in the protruding part of the obstacle, while distance points (red dots in Figure 22) are distributed at the long edge or in the concave part of the obstacle. Both of these types of sampling points constitute the key feature of the obstacles in the whole map. More importantly, some points locate in narrow channels (between obstacle No. 2 and 3, 21 and 23, etc.), which helps to form connected paths for the roadmap. There are no sampling points in the open area, so that the redundancy of the roadmap is reduced.

4.2.2. Roadmap Construction and Path Planning

In this section, the A* searching algorithm and the standard PRM are applied to verify the proposed method. The A* searching algorithm is the latest baseline method for the path planning of the lunar rover, which is applied in “Yutu-2”, the lunar rover in China’s Chang’e 4 mission to the Moon. In addition, the process of constructing a roadmap in our method is similar to that of PRM, the efficient and popular method in robotic path planning. Thus, we chose these two methods to compare with our method, and verify the advantages of our method in time consuming and path quality.

After 133 sampling points were selected within the safe zone (Figure 22), the roadmap was constructed, shown as red lines in Figure 23a. For comparison, we also randomly sampled 133 points in the safe zone to construct the roadmap of PRM, shown as red lines in Figure 23b. We can see that the connected paths of PRM cannot completely cover the safe zone. For example, there are numerous unconnected areas on the top left of obstacle No. 2, above No. 15, and on the right of No. 23. It is difficult to plan a feasible path to reach these areas because the planning result of PRM only consists of connected paths. In contrast, the roadmap constructed by our method forms a good envelope around obstacles, resulting in good connectivity. Furthermore, our method allows the planning result to be out of the roadmap in open areas, avoiding unnecessary turns caused by inappropriate sampling points.

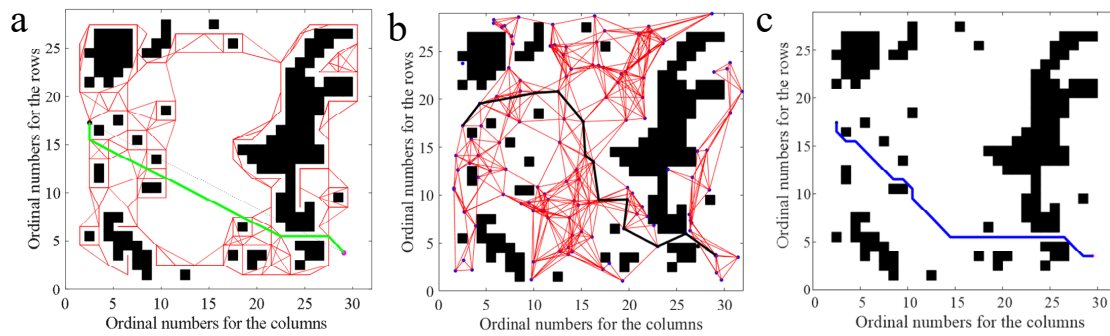


Figure 23. One of the path planning results. (a) The green line is the planned path using the sampling-based path planning method under linear guidance. (b) The black line is the planned path using the standard PRM method. (c) The blue line is the planned path using A* search algorithm.

One hundred groups of starting points and detecting points were randomly selected to plan the path in the safe zone by our method, standard PRM method, and the A* search algorithm, respectively. The planning results for one group are shown in Figure 23, i.e., the green line in (a), the black line in (b), and the blue line in (c).

The planning result in (a) is the shortest path with the least turns, which is beneficial to the accurate control and energy saving of the rover. The quality of the planning path in (b) is poor, especially for the detour around obstacle No. 6 and several turns at the open areas. As mentioned above, this is because PRM relies on an unreasonable roadmap, resulting in a very tortuous path. The A* algorithm searches the path directly without resorting to a roadmap. However, there are still unwanted turns around No. 8 and 12, so the path quality is slightly worse than (a).

The time consumption and length of each path planning are shown in Figures 24 and 25, where the length of zero means a planning failure.

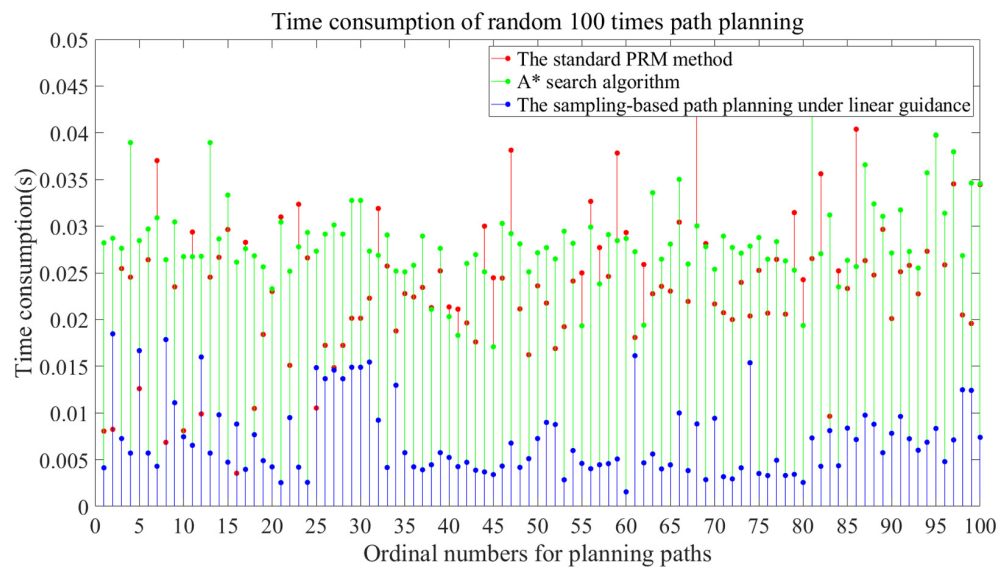


Figure 24. Comparison of time consumption in path planning.

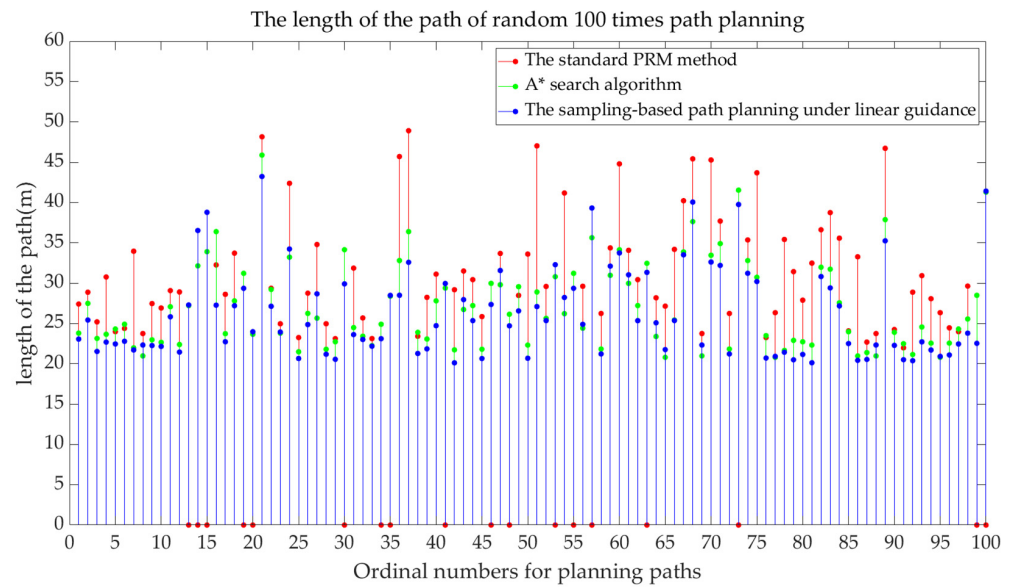


Figure 25. Comparison of the length of the path.

Figure 24 shows that, compared with the other two methods, our method has great advantage in time because local paths to bypass obstacles are searched in parallel in the roadmap. Figure 25 shows that our method can get slightly better results in length. Table 4 lists the detailed comparison of these three methods. It can be seen that our method has a higher success rate compared with the standard PRM method. The average path-planning time consumptions of our method reduce 69.25% and 74.71% compared with PRM and A* search algorithm, respectively. The average length shortens 19.89% compared with PRM and is nearly equal with the A* search algorithm, which always finds the shortest path.

Table 4. Comparison of algorithm performance.

Method	The Construction Time of Roadmap [s]	100 Times Path Planning		
		The Average Path-Planning Time [s]	The Average Length of the Planned Path [m]	Success Rate
The standard PRM method	0.034	0.023	31.18	82%
A* search algorithm	-	0.028	25.79	>99%
Our method	0.031	0.007	24.98	>99%

4.2.3. Verification of Map Extension

In this section, we verified the efficiency of our proposed map extension strategy for the continual addition of environmental information. Assuming that Figure 16 is the initial map, we gradually added three new sub-maps to the initial map, where the red, blue, and yellow boxes represent the first, second, and third updates, respectively, as shown in Figure 26a. We conducted 50 groups of planning experiments, each of which carried out full-map reconstruction planning and our map-extension planning for three map updates, respectively. Here, full-map reconstruction means that the added submap and the initial map are regarded as a new whole map for roadmap reconstruction. The roadmaps for each map update, as well as a group of planning results, are shown sequentially from Figure 26b–d.

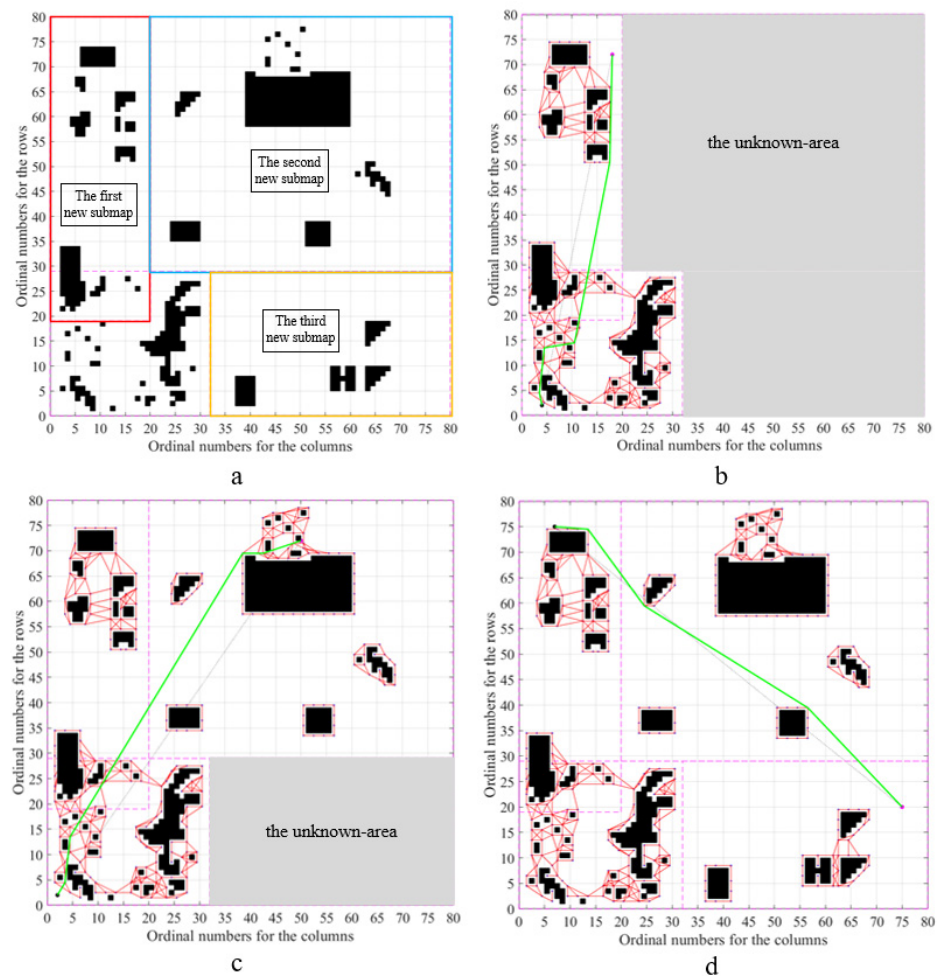


Figure 26. (a) Schematic of the map update. (b) Map after the first update. (c) Map after the second update. (d) After the third update. The red lines represent the roadmap and the green line is one of path planning results in (b–d).

Table 5 lists the average time consumption of roadmap reconstruction with the above two planning methods. We can see that the time of full-map reconstruction is related to the complexity of the whole map after adding the current submap. Therefore, with the gradual expansion of the map, the reconstruction time will be longer. In contrast, the time of map-extension reconstruction is only related to the complexity of the current submap. Note that the second sub-map is larger than the other two, so the reconstruction time for the second update was significantly longer than the first. However, no matter how long the second reconstruction took, the time of the third reconstruction is independent of the second reconstruction. Therefore, our method can get rid of the influence of the whole map scale on reconstruction and improve planning efficiency.

Table 5. Time consumption of roadmap reconstruction.

Map Update	Initial Map	1st Update	2nd Update	3rd Update
Map-extension reconstruction [s]	0.031	0.02	0.05	0.023
Full-map reconstruction [s]	0.031	0.086	0.275	0.303

Figure 27 shows the time consumption of 50 planning results to compare the efficiency of the above two planning ways. The average planning time is 0.0155 s in full-map

reconstruction planning, while it is 0.0107 s in map-extension planning, which is reduced by 31%. That is mainly because the local path will be searched in the corresponding roadmap if the linear path intersects with the obstacle in the submap, which reduces the searching space.

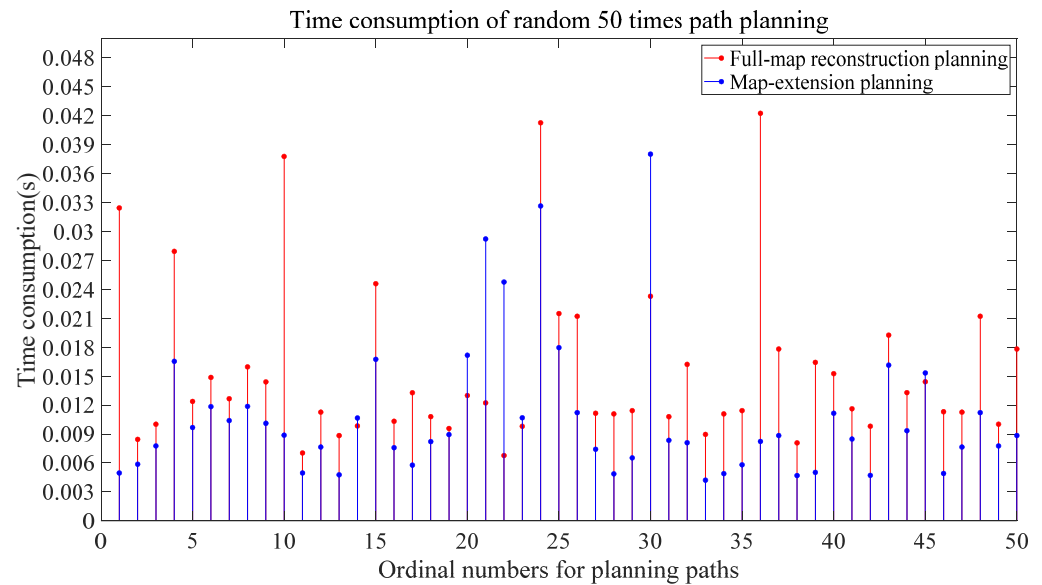


Figure 27. Comparison of time consumption in path planning within the third-updated map.

According to the above simulation results, we can see that our map extension strategy improves the efficiency of path planning from both roadmap reconstruction and path search when the map is continually supplemented.

5. Discussion

In the present study, we focused on the two steps in lunar surface exploration, detecting point selection and path planning, to improve the autonomy and efficiency for the rover in lunar exploration.

Unlike most existing manual selection or single-factor selection strategies, our autonomous selection strategy took terrain features and operating conditions of detection instruments into account. This is because the number of possible detection points is greatly limited by the mobility of the rover. In Section 4.1, we took a numerical simulation by a 16 m × 14.5 m lunar surface map. For the 10 experiments with random initial conditions, the average consumed time and final fitness value are 12.937 s and 0.988, respectively, which represent fast planning speed as well as high quality in the selected detecting point. Therefore, this strategy is more practical concerning improving the autonomy of the rover, as well as avoiding the manual misjudgment and information transmission delay between the Moon and the Earth.

Moreover, in our proposed method, the combination of the sampling-based methods and linear guidance can make the path planning more efficient. In Section 4.2, an experiment as a comparison with the above lunar surface map was executed by using our method, the A* algorithm, and the PRM method. The results show that our method has great advantages in both time and quality of the path. Compared with the PRM method, the average time consumption and length of the path are reduced by 69.25% and 19.89%, respectively. That is because our method greatly improves the connectivity of dense obstacle areas, and simplifies the process of path searching in open areas. Compared with the A* algorithm, not only is the average time consumption reduced by 74.71%, but unnecessary turns of the path are also avoided. Furthermore, the map extension strategy can avoid the disadvantages of the full-map reconstruction planning in the unstructured, continually varying environment. In the simulation environment, the efficiency of roadmap reconstruction after map update

has been improved by orders of magnitude, while the time for full-map reconstruction still depends on the accumulation of each reconstruction from the initial map to the current update. In addition, the time for map extension-based planning is also reduced by 31% compared with the full-map planning. This improvement is mainly due to the use of the roadmap's local characteristics in path searching, which avoids invalid searches in the other added submaps.

Despite the great improvement in the autonomy and planning efficiency, there are still some limitations in our method. Firstly, for the map extension strategy, the special case where a common part exists between the initial map and the submap has not been considered. In this case, the submap needs to be divided to ensure that the added submap is completely new concerning the initial map. Secondly, our method is applied in a two-dimensional environment, and we hope to extend it to a three-dimensional case with appropriate improvements. These two topics will be our future concerns.

6. Conclusions

In this paper, we proposed an autonomous task planning method for the lunar rover in a complex environment. Considering the equipment requirements and terrain features, a strategy for autonomous selection of detecting points is designed, where the safe zone for the rover's motion is defined, and within it the detecting points can be selected. According to the characteristics and distribution of obstacles in the environment, the safe zone is divided into two kinds of area, the open area and the dense obstacle area. Then the points are sampled in the dense obstacle area to construct the local roadmap. A feasible path from the current position to the detecting point can be planned by combining the linear path with local paths searched within the roadmap. In addition, taking into consideration the change of the unstructured lunar environment and the enlargement of the detecting area, a map extension strategy is proposed to further improve our path planning method in a real environment. The contributions of this paper mainly include: (1) The autonomous detecting points selection strategy is proposed, with terrain and instrument requirements taken into consideration. (2) The sampling-based path planning method under linear guidance can reduce the time for roadmap construction and the planning process, and so the planning efficiency can be greatly improved with high connectivity and high planning success rate. (3) The map extension strategy can deal with the expansions and changes of the map, further improving the planning capacity and efficiency. The insights from this study may contribute to the task planning of surface rovers for extraterrestrial objects.

Author Contributions: Conceptualization, G.C. and H.Y.; methodology, Z.H.; software, H.Y.; validation, Z.H., J.F. and Y.W.; formal analysis, J.F.; investigation, H.Y.; resources, G.C.; data curation, Z.H.; writing—original draft preparation, H.Y.; writing—review and editing, G.C.; visualization, Z.H.; supervision, Y.W.; project administration, H.Y.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was co-supported by the Research Fund of the Manned Space Engineering (No. 18051030101), the National Natural Science Foundation of China (No. 62173044), State Administration of Science Technology and Industry for National Defense (No. HTKJ2019KL502012), the Science and Technology Foundation of State Key Laboratory (No. 19NY1208), and BUPT Excellent Ph.D. Students Foundation (CX2021119).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the reviewer for their valuable comments and suggestions that greatly improve our paper. Special thanks to Yu Cheng for the careful revision in the writing.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, M.C.; Song, Z.M.; Dai, G.M.; Peng, L.; Zheng, C. Asteroids Exploration Trajectory Optimal Design with Differential Evolution Based on Mixed Coding. *Int. J. Distrib. Sens. Netw.* **2015**, *2015*, 1–8. [[CrossRef](#)]
2. Tom, E.; Pierre, L.; Yin, H.R. Dynamic Anchoring in Soft Regolith: Testing and Prediction. *J. Aerosp. Eng.* **2018**, *31*, 04017087.
3. Burns, J.O.; Benjamin, M.; Matthew, S. Science on the lunar surface facilitated by low latency telerobotics from a Lunar Orbital Platform-Gateway. *Acta Astronaut.* **2019**, *154*, 195–203. [[CrossRef](#)]
4. Xing, Y.; Teng, B.Y.; Liu, X. Guidance, Navigation and Control Technology for Lunar Surface Exploration. *Chin. J. Space Sci.* **2016**, *36*, 196–201.
5. Zhao, N.; Zhu, P.M.; Yuan, Y.F. The shallow subsurface structures of Chang'E-3 landing site based on the wavefield characteristics of LPR Channel-2B data. *Adv. Space Res.* **2018**, *62*, 884–889. [[CrossRef](#)]
6. Wang, L.W.; Guo, D.W.; Zhang, Z.; Lyu, Z.H.; Zhao, M.; Liu, Y. Introduction to NASA's Artemis Lunar Exploration Program. *Spacecr. Recovery Remote Sens.* **2020**, *41*, 12.
7. Dai, Z.P. *Small-Scale Complex Environment Perception and Terrain Modeling for Patrol and Exploration Missions*; Hubei University of Technology: Wuhan, China, 2020.
8. Di, K.C.; Wang, J.; Xing, Y.; Liu, Z.; Wan, W.; Peng, M.; Wang, Y.; Liu, B.; Yu, T.; Li, L.; et al. Progresses and prospects of environment perception and navigation for deep space exploration rovers. *Acta Geod. Cartogr. Sin.* **2021**, *50*, 12.
9. Gao, W.; Cai, D.B.; Zhou, J.P.; Gao, Y.; Yin, M. Activity planning method for Chang'E-3 "Jade Rabbit" rover. *J. Beijing Univ. Aeronaut. Astronaut.* **2017**, *43*, 277–284.
10. Li, L.H.; Lian, J.; Huang, H.Y. Traversability based obstacle avoidance path-planning and path-following control for lunar rover. *J. Intell. Fuzzy Syst. Appl. Eng. Technol.* **2015**, *28*, 547–559. [[CrossRef](#)]
11. Garrido, S.; Martin, F.; Alvarez, D.; Moreno, L. Towards an Anisotropic Fast Marching Method applied to the Path Planning Task for Mars Rovers. In Proceedings of the 6th International Conference on Space Mission Challenges for Information Technology (SMC-IT), Madrid, Spain, 27–29 September 2017; pp. 13–19.
12. Zhou, L.F.; Yang, L.N.; Tang, H.W. Research on Path Planning Algorithm and Its Application Based on Terrain Slope for Slipping Prediction in Complex Terrain Environment. In Proceedings of the 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Shenzhen, China, 15–17 December 2017; pp. 224–227.
13. Kosek, F.; Culka, A.; Drahota, P. Applying portable Raman spectrometers for field discrimination of sulfates: Training for successful extraterrestrial detection. *J. Raman Spectrosc.* **2017**, *48*, 1085–1093. [[CrossRef](#)]
14. Hewitt, R.A.; Ellery, A.; Ruiter, D.A. Training a terrain traversability classifier for a planetary rover through simulation. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1–14. [[CrossRef](#)]
15. Orger, N.C.; Yin, Y.; Chen, Z. Dynamic analysis for peg-in-hole assembly with contact deformation. *Int. J. Adv. Manuf. Technol.* **2006**, *30*, 118–128.
16. Wu, W.R.; Zhou, J.L. Key Technologies in Teleoperation of the "Yutu" lunar rover. *Sci. Sin. Inf.* **2014**, *44*, 425–440.
17. Wang, K.W.; Peng, S. Study on Path Planning of Lunar Rover Based on A* Algorithm Optimization. *Spacecr. Eng.* **2019**, *28*, 19–26.
18. Seo, M.G.; Hong, S.M.; Tahk, M.J. Lunar Lander Landing Site Decision in Low-Fuel Case. *MATEC Web Conf.* **2016**, *54*, 9003. [[CrossRef](#)]
19. Saito, M.; Nonaka, K.; Sekiguchi, K. Optimal Path Planning utilizing Dissipation Function based on Terrain Elevation Map of lunar rovers. In Proceedings of the 2017 11th Asian Control Conference (ASCC), Gold Coast, Australia, 17–20 December 2017; pp. 394–399.
20. Amato, N.M.; Bayazit, O.B.; Dale, L.K. OBPRM: An Obstacle-Based PRM for 3D Workspaces. In Proceedings of the 3rd Workshop on the Algorithmic Foundations of Robotics on Robotics, Houston, TX, USA, 5–7 March 1998; pp. 155–168.
21. David, H.; Jiang, T.T.; John, R. The bridge test for sampling narrow passages with probabilistic roadmap planners. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), Taipei, China, 14–19 September 2003; Volume 3, pp. 4420–4426.
22. Fareh, R.; Baziyad, M.; Rabie, T. Enhancing Path Quality of Real-Time Path Planning Algorithms for Mobile Robots: A Sequential Linear Paths Approach. *IEEE Access* **2020**, *8*, 167090–167104. [[CrossRef](#)]
23. Speyerer, E.J.; Lawrence, J.D.; Stopar, J.D. Optimized traverse planning for future polar prospectors based on lunar topography. *Icarus* **2017**, *273*, 337–345. [[CrossRef](#)]
24. Li, Q.H.; Pan, F.; Zhao, Z.G. Process Modeling and Monitoring with Incomplete Data Based on Robust Probabilistic Partial Least Square Method. *IEEE Access* **2018**, *6*, 10160–10168. [[CrossRef](#)]
25. Ryu, H.; Chung, W.K. Local map-based exploration using a breadth-first search algorithm for mobile robots. *Int. J. Precis. Eng. Manuf.* **2015**, *16*, 2073–2080. [[CrossRef](#)]
26. Zhang, T.; Xu, H.X. Path Planning for Lunar Rover Parking in Place to Probe. *Comput. Simul.* **2017**, *34*, 110–114.
27. Zhang, J.T.; Yang, F.X.; Weng, X. An Evolutionary Scatter Search Particle Swarm Optimization Algorithm for the Vehicle Routing Problem with Time Windows. *IEEE Access* **2018**, *6*, 63468–63485. [[CrossRef](#)]
28. Lozano-Perez, T.; Wesley, M.A. An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles. *Commun. ACM* **1979**, *22*, 560–570. [[CrossRef](#)]