LiDAR-Inertial SLAM with DEM-driven Global Constraints for Planetary Rover Exploration

Xusheng Zhang, Yuan Li, Zeyuan Cao, Junying Lv, Zefeng Huang, Wuming Zhang

School of Geospatial Engineering and Science, Sun Yat-Sen University, Zhuhai 519082, China

Keywords: Planetary Rover, SLAM, Monte Carlo Localization, DEM, Graph Optimization.

Abstract

The positioning capability of a rover is a critical factor that determines the efficiency of a planetary exploration missions. Most rovers primarily rely on visual odometry for trajectory estimation and relative pose determination. However, the distinctive characteristics of planetary environment, particularly limited visibility, can significantly compromise the accuracy of positioning. To address the limitations of visual localization techniques, this study introduces a novel LiDAR-Inertial SLAM framework integrating satellite Digital Elevation Model (DEM) data. A prior map is developed using DEM data through surface fitting after an offline process, followed by three online operational subparts: LiDAR odometry for estimate pose, 3D-Monte Carlo Localization (3D-MCL) adjust the estimate pose using the prior map, and graph optimization for final pose estimation. Experimental results indicate an average absolute trajectory error of approximately 0.6 meters, demonstrating the framework's effectiveness for long-distance navigation.

1. Introduction

Rovers play a pivotal role in planetary exploration. To ensure effective exploration, precise mapping and real-time positioning are critical. Existing positioning methods for rovers primarily include image matching (Di et al., 2011), celestial navigation (Ma et al., 2015), Very Long Baseline Interferometry (VLBI) (Brisken et al., 2002), and dead reckoning (Nister et al., 2004). Among these, dead reckoning offers real-time and relatively precise pose estimates that meet the demands of effective exploration. The rapidly advancing Simultaneous Localization and Mapping (SLAM) technology, which evolves from dead reckoning, enables the construction of high-precision environmental maps while simultaneously obtaining real-time pose estimates, thus presenting a novel approach for rover positioning and mapping.

However, due to the sparse features and significant lighting variations on planetary surfaces, the robustness of existing SLAM frameworks is often inadequate (Lambert et al., 2012). For instance, in lunar south pole exploration, numerous permanently shadowed regions hinder the effectiveness of vision-based SLAM frameworks (Carle et al., 2010). LiDAR-based SLAM frameworks also encounter challenges in these areas due to a lack of robust feature points, resulting in error accumulation over prolonged operations without conditions for loop closure (Lin et al., 2021). Currently, mainstream LiDAR SLAM solutions such as LOAM (Zhang et al., 2014), FAST-LIO (Xu et al., 2021), and LIO-SAM (Shan et al., 2020) primarily utilize relative positioning frameworks. Absolute positioning can only be achieved if the rover's initial pose is accurately provided.

Recently, positioning methods that leverage prior information (such as High Definition (HD) map or high-precision point cloud maps) have been developed. These methods can be categorized into feature matching and particle filtering techniques. Feature matching involves extracting features from both prior and currently acquired information, such as utility poles (Chen et al., 2021), curbs (Wang et al., 2017), corners of buildings and lane markings (Im et al., 2016; Kim et al., 2020). These current features are then matched with those from the prior information using

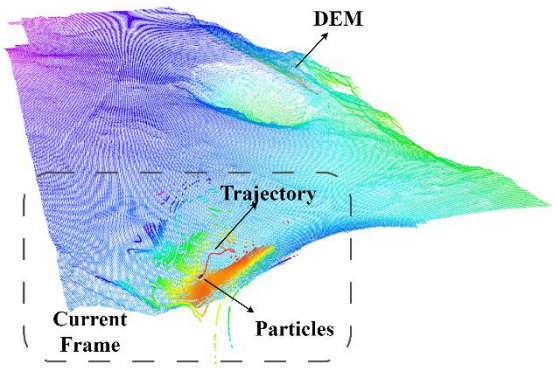

Figure 1. DEM-driven LiDAR-Inertial SLAM. The background features a point cloud interpolated from DEM data. The current frame point cloud is matched in real-time with the prior map to obtain accurate poses and trajectories, enabling the construction of a detailed local point cloud map.

registration techniques to calculate global localization (Magnusson et al., 2007). Another method, known as particle filtering, is a recursive state estimation algorithm based on Bayesian filtering, often referred to as Monte Carlo Localization (MCL) in the positioning field (Dellaert et al., 1999). This technique uses a set of random particles to represent and estimate the state distribution of a dynamic system, which is typically employed in 2D environments, especially indoors (Hess et al., 2016). For outdoor environments, sufficient features like lane markings, poles, curbs, and walls are required to project onto a 2D plane to construct a prior raster map for global localization (Chalvatzaras et al., 2022; Sefati et al., 2017). While some researchers have extended MCL-based methods to 3D, these frameworks assume the availability of high-precision point cloud data obtained using the same sensors as the prior map. Notable frameworks in this domain include HDL-Localization (Koide et al., 2019) , where the global positioning algorithm employs Normal Distributions Transform (NDT) or Iterative Closest Point (ICP) to match the current frame's point cloud with the prior point

cloud map in real time. Additionally, NDT-MCL (Saarinen et al., 2013), initially proposed for precise indoor robot localization, constructs an NDT prior map by scanning the indoor environment to create a detailed 3D representation, thereby matching the current frame's point cloud with the NDT prior map for localization in real time.

However, planetary scenes often lack HD maps and high-precision point cloud maps. Given the weak environmental conditions and the scarcity of man-made landmarks, achieving real-time positioning and mapping in planetary environments necessitates more effective informational resources. Currently, many satellites used for planetary exploration have collected vast amounts of data. Among these, DEM data derived from laser altimetry, although low in resolution, provides wide coverage and extensive topographical features that are beneficial for rover positioning. Therefore, this paper proposes using DEM data as a global constraint for real-time positioning and mapping of rovers. Additionally, considering that visual sensors may be limited by lighting conditions in certain planetary environments, and with deep-space missions increasingly focusing on establishing long-term planetary bases (Pei et al., 2020), this paper assumes that, provided there is sufficient energy from these bases, Light Detection and Ranging (LiDAR) will be selected as the primary sensor for data collection. The working effect of the proposed framework is shown in Figure 1.

The remainder of this paper is organized as follows: The next section provides an overview of the proposed framework. Section 3 describes the application of MCL, the creation of prior maps, and the construction of the laser odometry. Section 4 details the factor graph construction. Section 5 introduces the dataset, experimental process, and results. Finally, Section 6 presents the conclusion.

2. System Overview

In this work, a tailored SLAM framework specifically designed for planetary rovers. It integrates 3D-MCL using LiDAR, IMU, and DEM data. The pipeline of the proposed framework is illustrated in Figure 2.

The overall process is divided into two parts: the offline component and the online component. The primary purpose of the offline part is to produce a prior map. Although there are no HD maps or high-precision point cloud maps available for planetary scenes, DEM data can provide essential topographic information. Therefore, this paper focuses on generating a prior map utilizing DEM data. Rather than simply converting the DEM data into a point cloud, we first interpolate the DEM data into a point cloud format. Next, voxelization is been performed on the interpolated point cloud. Finally, surface fitting is conducted on the voxels containing points to generate the prior map.

The online part mainly consists of three subparts. In the first subpart, a curvature-based method is used to extract feature points of the current LiDAR frame. These points are then matched between consecutive frames to derive the rover's odometry information, providing a reliable initial estimate of rover’s motion continuously. In the second subpart, the MCL subscribes point cloud of current frame, the estimate poses from odometry, and the prior map with its range. At start-up, the MCL generates an initial particle set based on the range and predefined bias values in four dimensions (x, y, z, yaw). As rover moves, the MCL uses odometry information to update the particle set, with the centre of the new particle set indicates the rover's pose currently, this phase is called prediction. Following this, the point

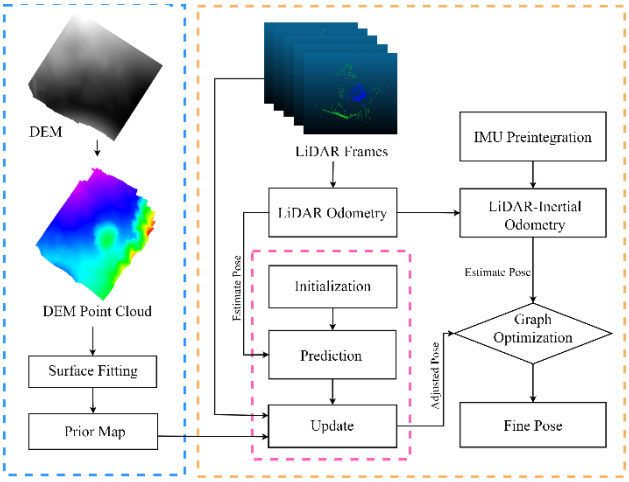

Figure 2. Workflow of the proposed framework. Offline part is in blue box, while yellow box contains the online part of the framework, pink box circles the MCL process.

cloud from the current frame is aligned with the pose of each particle and compared against the prior map. This comparison assesses the quality of alignment, particles are then weighted based on this quality, with those particles with higher weights are retained and replicated, while those with lower weights are discarded. This update phase is crucial for refining the particle set for subsequent prediction. The outputted poses from the MCL consistently incorporate adjustments based on the global constraint provide by DEM. Finally, the third step integrates the results from the previous two subparts, and produce the final pose estimation of the rover using graph optimization. Concurrently, a local map is constructed using these optimized poses.

3. Monte Carlo Localization

3.1 Prior Map

DEM data is typically represented in a geographic coordinate system, which must first be projected to an appropriate coordinate system for general utilization. Additionally, due to the significant storage requirements associated with DEM data stored in TIFF format, this paper employs bicubic interpolation to derive point clouds from DEM data. This approach reduces storage needs while maintaining efficiency in data processing and analysis.

Common forms of prior maps include occupancy grid maps and NDT maps. An occupancy grid map is a raster map used for environmental modelling and representation. The entire mapped area is divided into uniform grids, with each cell representing a small region of the environment. The value of each cell typically indicates occupancy; a value of 0 signifies that the region is empty, while a value of 1 indicates that it is completely occupied by an obstacle. However, the high resolution of DEM data often results in large grid cells, which can lead to reduced precision when utilizing occupancy grids. Similarly, NDT-based prior maps face limitations due to the insufficient density and low resolution of DEM point clouds. It is sensitive to the quality and quantity of point clouds, and its effectiveness may be compromised when applied to the low-resolution point clouds generated by DEM.

To address these challenges, a virtual surface is constructed as a prior map based on second-order surface fitting, leveraging the characteristics of DEM data. This method aims to provide a more accurate and efficient representation for prior mapping purposes,

thereby enhancing both storage efficiency and analytical capabilities. This paper constructs prior maps using NDT and second-order surface fitting for subsequent experimental comparisons.

3.1.1 NDT Prior Map. The NDT is a method used for map representation and matching, initially introduced by Biber and Strasser (Biber et al., 2003) for 2D scan matching and later extended to 3D applications. NDT effectively transforms measurement data into parameters of a normal distribution, allowing for accurate modelling and representation of the environment.

The core concept of NDT involves partitioning space into fixed-size grid cells and employing normal distributions to characterize the point cloud data within each cell. The process begins by subdividing the space into a regular grid. For each cell containing at least three points, the mean vector q of the points is calculated using the formula (1):

$$q = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

where n represents the number of points in the grid cell, and x_i is the coordinate of the i-th point. Following this, the covariance matrix Σ for the points within the cell is computed using formula (2):

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(x_i - q)(x_i - q)^T \tag{2}$$

The point cloud data within each grid cell is then represented by a normal distribution $N(q, \Sigma)$, where q is the mean vector and Σ is the covariance matrix. The probability density of any point x within the grid cell is described by the normal distribution as shown in formular (3):

$$p(x) \sim exp\left(-\frac{1}{2}(x-q)^T\Sigma^{-1}(x-q)\right) \tag{3}$$

In this context, x denotes the coordinate of the point, the normal distribution $N(q, \Sigma)$ characterizes the distribution of points within the cell.

3.1.2 Surface Fitting. Given the differences from urban and indoor scene point cloud data, the point cloud generated from DEM data has a lower resolution and lacks distinct features, with only noticeable terrain undulations. Therefore, this paper first constructs a voxel grid based on the range of the point cloud and counts the number of points within each voxel. A surface fitting method is then used to obtain a virtual surface within each voxel. In this paper, a second-order surface is used for the construction, as described by the following formula (4):

$$\begin{bmatrix} x_1^1 & y_1^2 & x_1y_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m^2 & y_m^2 & x_my_m & x_m & y_m & 1 \end{bmatrix} \begin{bmatrix} a_{20} \\ a_{02} \\ a_{11} \\ a_{10} \\ a_{01} \\ a_{00} \end{bmatrix} = \begin{bmatrix} f(x_1, y_1) \\ \vdots \\ f(x_m, y_m) \end{bmatrix} \tag{4}$$

The left matrix is an $m \times 6$ matrix, where each row corresponds to a data point (x_i, y_i) with the elements $i_2^x, i_2^y, x_iy_i, x_i, y_i$, and 1. The middle column vector contains the coefficients $a_{20}, a_{02}, a_{11}, a_{10}, a_{01}$, and a_{00} of the second-order surface equation. The right column vector contains the function values $f(x_i, y_i)$ at each data point.

Due to the need for solving six parameters in second-order surface fitting, at least six points must be located within the same unit. The coefficients of the second-order polynomial for units meeting these conditions are solved and saved as a prior map. Essentially, this process preserves the terrain undulation characteristics within each unit cell.

3.2 Monte Carlo Localization in Three-Dimension

MCL is a probabilistic method used for robot localization in an unknown environment. It estimates the robot's position and orientation using a set of random samples, known as particles. MCL employs recursive Bayesian estimation to iteratively refine the distribution of these particles based on sensor data and motion commands, thereby estimating the robot's true position. Below is a detailed description of the MCL approach and its four main phases.

Initialization: At the beginning of the localization process, MCL initializes a set of particles, each representing a possible initial state of the robot, including its position and orientation (x, y, z, yaw). These particles are distributed randomly within the environment, typically based on prior knowledge or assumptions about the robot's initial location and the uncertainty in its starting point.

Prediction: MCL predicts the new state of each particle based on the robot's motion model. This involves updating the position and orientation of each particle according to the distance and yaw angle of the robot's movement, which can be modelled as follows:

$$p_k^X = p_{k-1}^X + dX + w \tag{5}$$

Where p_k^X represents the state (x, y, z, yaw) of the p-th particle at the k-th time step, dX is the distance and yaw angle inferred from the LiDAR odometry, and w is Gaussian white noise, representing the error in the odometry measurement. Thus, the new state of particles can be obtained.

Update: The update phase involves incorporating sensor data to refine the estimates of the robot's position. This typically includes comparing the sensor readings, such as distances or angles to environmental features, with the predicted positions of the particles.

$$t_i^w \propto p\left(z_t | t_i^X\right) \tag{6}$$

Where t_i^w denotes the weight of the i-th particle, and $p\left(z_t | t_i^X\right)$ represents the likelihood of observing the sensor data z_t given the i-th particle's position t_i^X.

The weight update can incorporate a calculation based on the distance of particles to a hypothetical surface: For a particle p_i in state (x, y, z), the distance to the hypothetical surface can be represented as follow:

$$d_{j,i} = \frac{|f(x_j, y_j, z_j)|}{\|\nabla f(x_j, y_j, z_j)\|} \tag{7}$$

where $\nabla f\left(x_j, y_j, z_j\right)$ is the normal direction of the surface. The weight of the particle can then be calculated as the average distance using formular (8) as below:

$$w_i = \frac{1}{N}\sum_{j=1}^{N} d_{j,i} \tag{8}$$

Combining the weight update equation, formular (9) can be obtained:

$$w_i = \frac{1}{N}\sum_{j=1}^{N}\frac{|f(x_j,y_j,z_j)|}{\|\nabla f(x_j,y_j,z_j)\|} \quad (9)$$

Through this equation, the particle weights are effectively updated, furthermore, the state of current rover can be obtained by formular (10) as follow:

$$\hat{x}_t = \sum_{i=1}^{M} t_i^x \cdot t_i^w \quad (10)$$

Where $\hat{x}_t$ represents the predicted state of the robot, t_i^x denotes the state of the i-th particle, t_i^w is the weight of the i-th particle, and M is the total number of particles.

Resampling: Due to the varying weights of particles, some particles may become overrepresented while others may be underrepresented, leading to a loss of diversity in the sample set. To maintain a representative distribution, MCL performs a resampling step where a new set of particles is generated based on the weights of the current particles.

$$\sum_{i=1}^{N} t_i^w = 1 \quad (11)$$

Particles with higher weights are more likely to be selected, ensuring that the sample set reflects the current belief about the robot's state.

Through these phases, MCL effectively estimates the robot's position and orientation. The MCL module in this paper is an extension of the AMCL3D framework (catec/amcl3d: Adaptive Monte Carlo Localization (AMCL) in 3D. (github.com)), We utilize the laser odometry mentioned in LIO-SAM as the motion model for the MCL module. For the detailed construction method, refer to (Shan et al., 2020). Initially, feature points and plane points are extracted based on different point cloud curvatures. Feature points are matched using the point-to-line distance method, while plane points are matched using the point-to-plane distance method, as shown in the following formulas:

$$d_{point-line} = \frac{|Ax+By+C|}{\sqrt{A^2+B^2}} \quad (12)$$

Where $Ax + By + C = 0$ represents the line equation, and (x, y) is the point's coordinates. This metric helps to minimize the distance between scanned points and the reference line, facilitating robust feature association. On the other hand, the point-to-plane distance is given by the formula (13):

$$d_{point-plane} = \frac{|Ax+By+Cz+D|}{\sqrt{A^2+B^2+C^2}} \quad (13)$$

Where $Ax + By + Cz + D = 0$ represents the plane equation, and (x, y) is the point's coordinates.

Based on the two aforementioned equations, the inter-frame matching relationship is derived. By applying iterative optimization using the Levenberg-Marquardt (L-M) algorithm, the motion transformation for each frame is obtained, which serves as the motion model for the prediction phase of MCL.

Considering that long-distance operation of laser-based Simultaneous Localization and Mapping (SALM) can result in trajectory drift, where the current frame's point cloud has a reduced match with the prior map, this paper also includes a feature to adaptively update the particle generation range. Once a decrease in matching accuracy is detected, the range for generating particles in the next frame is updated based on the product of the pose transformation and an adaptive coefficient.

Following these steps, by using a laser odometer as the motion model and the fit of a second-order surface as the observation model, and applying Sequential Importance Resampling (SIR), the real-time pose of the rover corrected by DEM data can be obtained.

4. Graph Optimization

Since the laser odometry obtained based on LiDAR point cloud frame-to-frame matching only contains relative position information and faces trajectory drift issues during long-distance movement, while the MCL correction results contain absolute position information but are prone to sudden changes, this paper fuses the two by constructing a factor graph. Considering the low frequency of point cloud frame-to-frame matching, IMU pre-integration factors are also introduced. The use of each factor is as follows:

LiDAR odometry factor: The lidar odometry factor utilizes point-to-line and point-to-plane matching techniques, as implemented in LIO-SAM (Shan et al., 2020). This factor generates constraints based on the alignment of laser scan points with geometric featur-

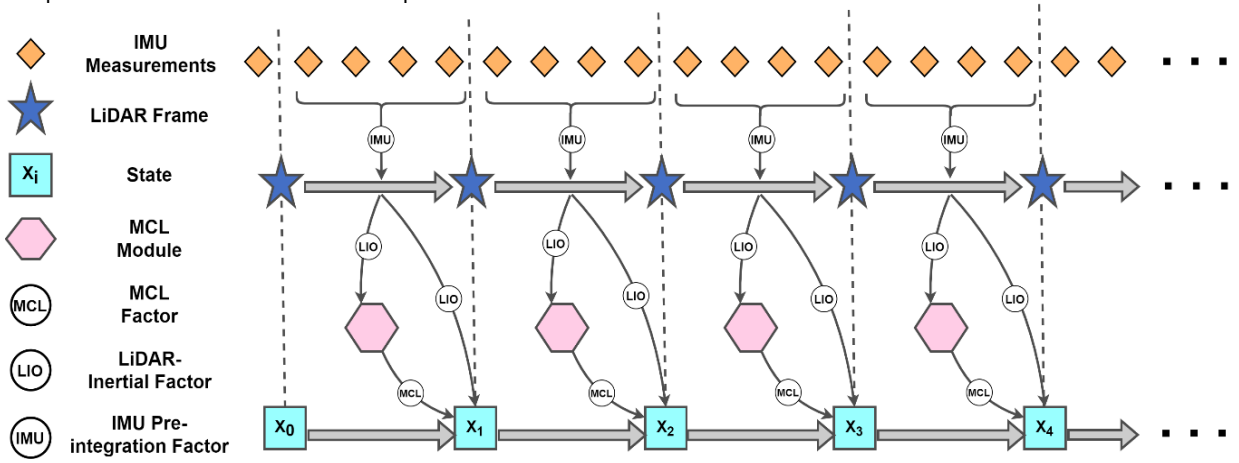

Figure 3. The yellow squares represent IMU messages, the blue stars represent LiDAR frames, the pink hexagons represent the MCL module, and the cyan squares represent the current state of the rover.

es in the environment, such as edges and planes. By minimizing the distance between observed points and their corresponding model projections, this factor improves the accuracy of relative motion estimates. In the factor graph, each lidar measurement introduces constraints that refine the trajectory by effectively capturing the robot's movement over consecutive frames, accounting for uncertainties in the measurements.

MCL factor: It not only provides absolute positional information but also corrects the LIO pose. By combining laser measurements with particle filtering, MCL can identify the robot's position within the environment in real time. This mechanism introduces absolute constraints in the factor graph that enhance the stability and accuracy of pose estimation, especially in complex environments.

IMU pre-integration factor: The IMU pre-integration factor leverages inertial measurements to build a dynamic model of motion between keyframes. By integrating angular velocity and linear acceleration over specific time intervals, this factor provides robust motion estimates, which are especially valuable during periods with poor visual observation. The integration is performed using a formulated state transition model, resulting in a compact representation that is incorporated into the factor graph. This ensures continuity in state estimation, thus aiding in reducing drift.

In summary, these factors are integrated into the factor graph to optimize the overall state estimation process, the overall factor graph is illustrating in Figure 3.

5. Experiment

5.1 Dataset

The ARCHES Mount Etna Dataset (AMEDS) was collected during a test campaign conducted on the slopes of Mount Etna in the summer of 2022 by ESA’s Human Robotic Interaction lab (HRI) (Suter et al., 2023), The experimental setup is shown in the figure 4. The data acquisition involved the HRI’s Interact rover, equipped with advanced navigation's spatial dual inertial navigation system (GNSS/INS), which combines a real-time kinematic compatible GNSS receiver and an IMU for high-precision pose measurement, a schematic diagram of the rover is shown in the Figure 4. Additionally, a velodyne VLP-32C LiDAR is mounted on the rover to acquire a large field-of-view data. The rover was tasked with performing nine traverses and two rock picking experiments, aimed at generating a robotic dataset representative of a lunar analogue environment. The experiments of the proposed framework and the subsequent analyses are based on this dataset. A bird's-eye view of the overall experiment area and the DEM data are illustrated in Figure 5.

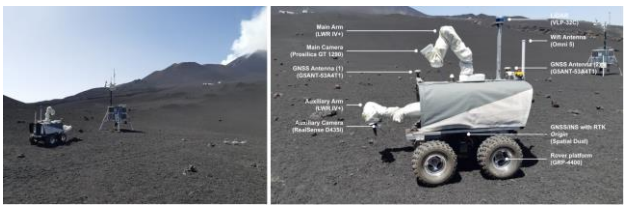

Figure 4. Environments and patrol devices employed in the dataset used in the experiment.

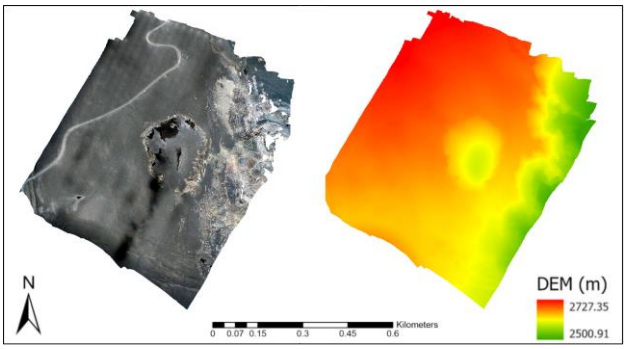

Figure 5. Overview of the experimental area from a bird's eye perspective and utilization of DEM Data in the proposed framework.

5.2 Results

Due to data loss encountered during each individual experiment in the AMEDS data collection, some of the ground truth data is inaccurate. Consequently, experiments were conducted using traverse4 and traverse6 to verify the absolute positioning capability of the proposed framework. We designated a 20m × 20m area as the starting region without providing an accurate initial pose. The operational process of the framework is illustrated in Figure 6.

It is evident that even without an accurate position and orientation, the proposed framework successfully achieves real-time absolute positioning and mapping. To further validate the real-time absol-

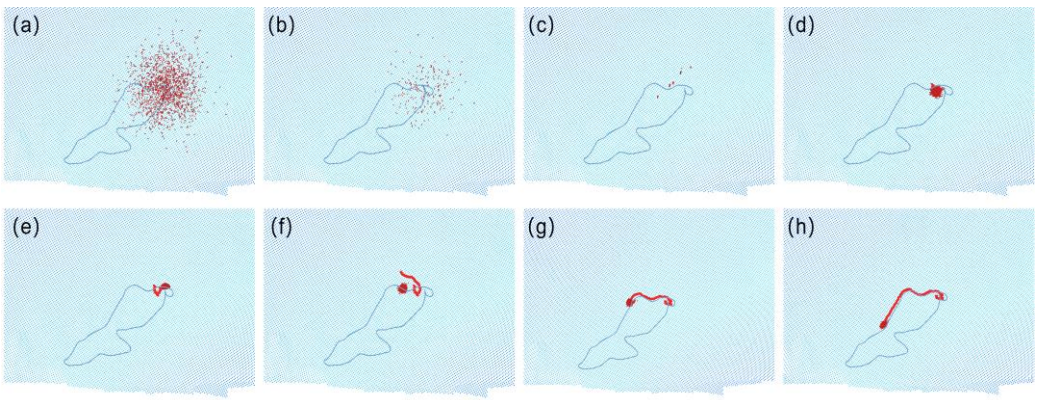

Figure 6. The operating process of the framework proposed in this paper. (a) the initial range of particles is about 400 m^2. (b)(c)(d) the particles have successfully gathered at the correct position due to the matching. (e)(f) the trajectory got a deviated direction and absolute position. (g)(h) As the MCL factor continuously checks the poses, the trajectory and moving direction can be accurately adjusted based on the MCL factor while ensuring the success of the localization task.

ute positioning capability of the proposed framework, we compared it with the open-source frameworks HDL-Localization, LeGO-LOAM (Shan et al., 2018), and LIO-SAM.

Due to the relatively flat terrain and the absence of significant obstacles, LeGO-LOAM failed to generate maps in tests conducted on traverse4 and traverse6. Additionally, the resolution of the DEM point cloud was too low for the HDL-Localization framework, resulting in localization failure as well. Therefore, we focused our comparison on LIO-SAM and the proposed framework. The red line in the Figure 7 represents the relative positioning results from LIO-SAM, while the green line illustrates the real-time absolute positioning results based on the proposed framework within a 400 square meter area.

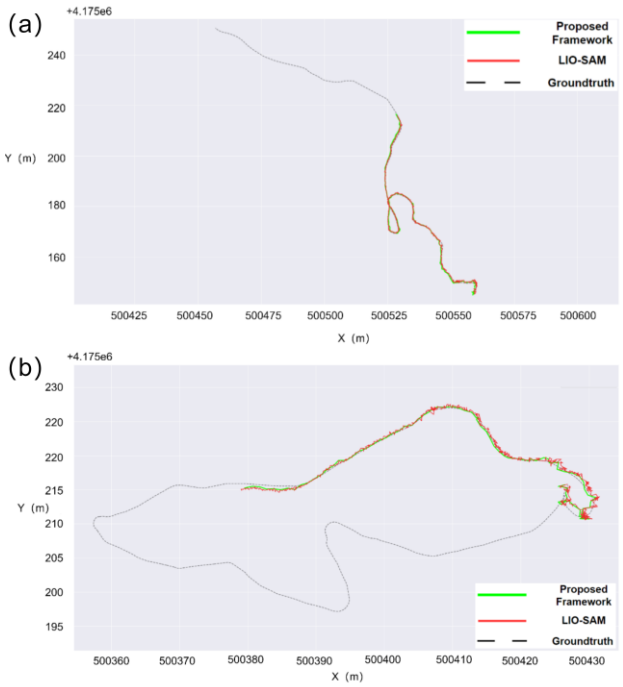

Figure 7. Comparison of trajectories between LIO-SAM and the proposed framework. (a) Comparison experiment on traverse4. (b) Comparison experiment on traverse6. The trajectories from both experiments were aligned using the EVO tool for comparison. The actual trajectory obtained by LIO-SAM deviates from the ground truth, whereas the proposed framework more accurately captures the rover's true absolute position.

Observing the ATE distribution of the positioning and mapping trajectory from the proposed framework (see Figure 8), it is evident that the localization error is primarily concentrated during the initial stage. Due to only providing a rough range (400 square meters), the ATE is relatively large at the beginning. However, as the rover continues to move, the localization error gradually decreases and stabilizes, as shown in the Figure 8. Qualitative comparison results of several frameworks are presented in the Table 1 and Table 2.

Considering the characteristics of NDT and second-order surface fitting, it is evident that voxel size significantly impacts the overall performance of the framework. Second-order surface fitting requires the estimation of six parameters, necessitating at least six points. If higher-order surface fitting is employed, even more points are needed. In this study, the DEM data used in the

	Absolute Positioning	Initial Range (m)	ATE (m)	SSE (m^2)	RMSE (m)
LeGO-LOAM	NO	—	—	—	—
LIO-SAM	NO	0	**0.581**	595.099	**0.650**
HDL-Localization	YES	—	—	—	—
Proposed Framework (NDT)	YES	—	—	—	—
Proposed Framework (surface fitting)	YES	400	0.612	**70.976**	0.686

Table 1. Trajectory comparison results of traverse4, where “--” means, the framework is failed using the DEM prior map.

	Absolute Positioning	Initial Range (m)	ATE (m)	SSE (m^2)	RMSE (m)
LeGO-LOAM	NO	—	—	—	—
LIO-SAM	NO	0	0.824	945.909	0.886
HDL-Localization	YES	—	—	—	—
Proposed Framework (NDT)	YES	—	—	—	—
Proposed Framework (surface fitting)	YES	400	**0.601**	**40.313**	**0.677**

Table 2. Trajectory comparison results of traverse6, where “--” means, the framework is failed using the DEM prior map.

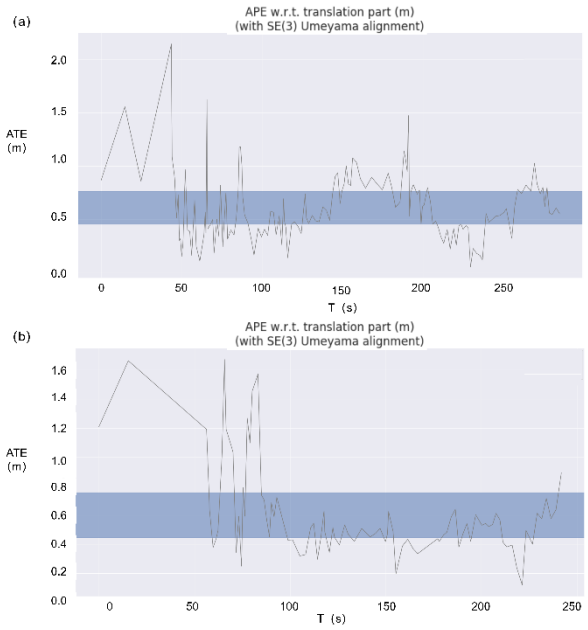

Figure 8. The Absolute Trajectory Error (ATE) varies over time. It can be observed that the periods with larger ATE are primarily at the initial stages, and as the rover continues to move, the ATE gradually stabilizes.

experimental dataset has a resolution of 1 meter, and the sampled point cloud also has a 1-meter resolution, which sets a voxel size of 3 meters as the baseline for the experiments. To investigate the effect of voxel size on the experimental results, tests were conducted using voxel sizes of 3, 5, 7, and 9 meters.

Voxel Size (m)	ATE (m)	RMSE (m)
3	0.636	0.742
5	0.591	0.709
7	0.618	0.696
9	0.569	0.642

Table 3. Results comparison with different voxel size.

The results are presented in Table 3, which compares the performance across different voxel sizes. As shown in the table, the 3-meter voxel size exhibits relatively lower accuracy compared to larger voxel sizes, primarily due to the decreased number of points it contains. Hence, selecting an appropriate voxel size is crucial for optimizing accuracy in the framework.

6. Conclusions

This paper addresses the current needs of rovers in deep space exploration missions by conducting preliminary research on synchronous real-time absolute positioning and mapping based on LiDAR technologies. Given the limitations of existing positioning methods, which are characterized by inadequate real-time responsiveness and trajectory drift issues in SLAM frameworks over long distances, a novel approach is proposed that integrates DEM data for real-time localization based on 3D-MCL. Experimental results demonstrate that, even within a 400 m² area, the proposed framework achieves relatively accurate absolute localization, with an average Absolute Trajectory Error (ATE) of approximately 0.6 meters. This research offers significant insights into real-time absolute positioning for rover operations.

References

Biber, P., Strasser, W. 2003: The normal distributions transform: a new approach to laser scan matching, paper presented at Proc eedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Cat. No.03CH37453), 27-31 Oct. 2003. doi.org/10.1109/IROS.2003.1249285.

Brisken, W. F., Benson, J. M., Goss, W. M., Thorsett, S. 2002: Very long baseline array measurement of nine pulsar parallaxes, *The Astrophysical Journal*, *571*(2), 906.

Carle, P. J. F., Furgale, P. T., Barfoot, T. D. 2010: Long-Range Rover Localization by Matching LIDAR Scans to Orbital Elevation Maps, *Journal of Field Robotics*, *27*(3), 344-370. doi.org/10.1002/rob.20336.

Chalvatzaras, A., Pratikakis, I., Amanatiadis, A. A. 2022: A survey on map-based localization techniques for autonomous vehicles, *IEEE Transactions on Intelligent Vehicles*, *8*(2), 1574-1596.

Chen, G., Lu, F., Li, Z., Liu, Y., Dong, J., Zhao, J., Yu, J., Knoll, A.. 2021: Pole-curb fusion based robust and efficient autonomous vehicle localization system with branch-and-bound global optimization and local grid map method, *IEEE Transactions on Vehicular Technology*, *70*(11), 11283-11294.

Dellaert, F., Fox, D., Burgard, W., Thrun, S. 1999: Monte Carlo localization for mobile robots, paper presented at 1999 IEEE International Conference on Robotics and Automation (ICRA), 10-15 May 1999. doi.org/10.1109/ROBOT.1999.772544.

Di, K., Liu, Z., Yue, Z. 2011: Mars Rover Localization based on Feature Matching between Ground and Orbital Imagery, *Photogrammetric Engineering & Remote Sensing*, *77*(8), 781-791. doi.org/10.14358/pers.77.8.781.

Hess, W., Kohler, D., Rapp, H., Andor, D. 2016: Real-time loop closure in 2D LIDAR SLAM, paper presented at 2016 IEEE International Conference on Robotics and Automation (ICRA), 16-21 May. doi.org/10.1109/ICRA.2016.7487258.

Im, J.-H., Im, S.-H., Jee, G.-I. 2016: Vertical corner feature based precise vehicle localization using 3D LIDAR in urban area, *Sensors*, *16*(8), 1268.

Kim, K.W. Jee, G.I. 2020: Free-resolution probability distributions map-based precise vehicle localization in urban areas, *Sensors*, *20*(4), 1220.

Koide, K., Miura, J., Menegatti, E. 2019: A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement, *International Journal of Advanced Robotic Systems*, *16*.

Lambert, A., Furgale, P., Barfoot, T. D., Enright, J. 2012: Field testing of visual odometry aided by a sun sensor and inclinometer, *Journal of Field Robotics*, *29*(3), 426-444.

Lin, X., Wang, F., Yang, B., Zhang, W. 2021: Autonomous Vehicle Localization with Prior Visual Point Cloud Map Constraints in GNSS-Challenged Environments, *Remote Sensing*, *13*(3), 506.

Ma, X., Fang, J., Ning, X., Liu, G., Ge, S. S. 2015: Autonomous celestial navigation for a deep space probe approaching a target planet based on ephemeris correction, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *229*(14), 2681-2699. doi.org/10.1177/0954410015586841.

Magnusson, M., Lilienthal, A., Duckett, T. 2007: Scan registration for autonomous mining vehicles using 3D-NDT, *Journal of Field Robotics*, *24*(10), 803-827. doi.org/10.1002/rob.20204.

Nister, D., Naroditsky, O., Bergen, J. 2004: Visual odometry, paper presented at 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 27 June-2 July 2004. doi.org/10.1109/CVPR.2004.1315094.

Pei, Z., Liu, J., Wang, Q., Kang, Y., Zou, Y., Zhang, H. 2020: Overview of Lunar Exploration and International Lunar Research Station, *Chinese Science Bulletin*, *65*(24), 2577-2586. doi.org/10.1360/tb-2020-0582.

Saarinen, J., Andreasson, H., Stoyanov, T., Lilienthal, A. J. 2013: Normal distributions transform Monte-Carlo localization (NDT-MCL), paper presented at 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 3-7 Nov. 2013. doi.org/10.1109/IROS.2013.6696380.

Sefati, M., Daum, M., Sondermann, B., Kreisköther, K. D., Kampker, A. 2017: Improving vehicle localization using semantic and pole-like landmarks, paper presented at 2017 IEEE Intelligent Vehicles Symposium (IV), 11-14 June 2017. doi.org/doi.org/10.1109/IVS.2017.7995692.

Shan, T. Englot, B. 2018: LeGO-LOAM_Lightweight and Ground Optimized Lidar Odometry and Mapping on Variable Terrain, paper presented at 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1-5 Oct.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D. 2020: LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping, in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, edited, pp. 5135-5142. doi.org/10.1109/iros45743.2020.9341176.

Suter, W., Fornarelli, L., Exter, E. D., Pereira, A., Putzeys, B., Ferreira, E., *et al.* 2023. The ARCHES Mount Etna DataSet (AMEDS): A planetary rover data collection in a lunar analogue environment., In 17th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA).

Wang, L., Zhang, Y., Wang, J. 2017: Map-based localization method for autonomous vehicles using 3D-LIDAR, *IFAC-PapersOnLine*, *50*(1), 276-281.

Xu, W. Zhang, F. 2021: FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter, *IEEE Robotics and Automation Letters*, *6*(2), 3317-3324. doi.org/10.1109/LRA.2021.3064227.

Zhang, J. Singh, S. 2014: LOAM: Lidar Odometry and Mapping in Real-time, *Robotics: Science and Systems*.