PUTN: A Plane-fitting based Uneven Terrain Navigation Framework

Zhuozhu Jian†, Zihong Lu †, Xiao Zhou, Bin Lan, Anxing Xiao, Xueqian Wang*, Bin Liang

Abstract—Autonomous navigation of ground robots has been widely used in indoor structured 2D environments, but there are still many challenges in outdoor 3D unstructured environments, especially in rough, uneven terrains. This paper proposes a plane-fitting based uneven terrain navigation framework (PUTN) to solve this problem. The implementation of PUTN is divided into three steps. First, based on Rapidly-exploring Random Trees (RRT), an improved sample-based algorithm called Plane Fitting RRT* (PF-RRT*) is proposed to obtain a sparse trajectory. Each sampling point corresponds to a custom traversability index and a fitted plane on the point cloud. These planes are connected in series to form a traversable “strip”. Second, Gaussian Process Regression is used to generate traversability of the dense trajectory interpolated from the sparse trajectory, and the sampling tree is used as the training set. Finally, local planning is performed using nonlinear model predictive control (NMPC). By adding the traversability index and uncertainty to the cost function, and adding obstacles generated by the real-time point cloud to the constraint function, a safe motion planning algorithm with smooth speed and strong robustness is available. Experiments in real scenarios are conducted to verify the effectiveness of the method. The source code is released for the reference of the community†.

I. INTRODUCTION

With the development of simultaneous localization and mapping (SLAM) technology and the improvement of computer performance, autonomous navigation technology is widely applied to ground robots. At present, the technology of 2D indoor ground mobile robots is relatively mature. By improving the robustness and efficiency of existing 2D autonomous navigation solutions [1] [2] [3], the autonomous navigation of robots has been widely used in many fields [4] [5], but autonomy on unstructured and off-road terrain remains a challenge. The difficulties mainly include the following two aspects: 1) the representation and calculation of large-scale maps are time-consuming; 2) unstructured terrain and real-time obstacles affect path planning.

To address these issues, this study first presents a new autonomous navigation framework based on point cloud plane fitting. It contains three parts: 1) sparse global trajectory generation based on random sampling combined with plane fitting; 2) dense path generation based on gaussian process regression (GPR) and linear interpolation; 3) a nonlinear model predictive control (NMPC) planner that avoids dynamic obstacle and guarantees safety. Our work focuses on the surface of the point cloud map, extracts a set of point clouds around the sampling node, and then fits a local plane with the set. For this plane, we propose custom evaluation indicators including flatness, slope, and sparsity, which are integrated to evaluate the traversability. Based on the fitted plane, we combine sampling, GPR, NMPC methods for motion planning of the robot.

A. Related Work

Autonomous navigation based on unmanned ground vehicles (UGV) has been greatly developed in recent years. Based on 2D navigation, many frameworks for 3D navigation have been proposed. In [6], a multi-layer 2D map extracted from the 3D OctoMap is used to explore staircases and slopes. However, this approach limits the ability to explore more complex environments. In [7] [8] [9], global trajectory in more complex environmental scenarios is resolved, but their work all lack processing of real-time point clouds, which may lead to dangerous situations during the movement of the robot. Fan et al. [10] apply rapid uncertainty-aware mapping and traversability evaluation to the robot, and tail risk is assessed using the Conditional Value-at-Risk (CVaR). The most visible deficiency is that when working in large-
scale scenarios, the algorithm needs to analyze each point to generate a risk map, which consumes a large number of computing resources.

Map representation of space is essential for finding paths. The density of the map affects the accuracy and speed of planning [11]. Therefore, when planning requires greater accuracy, the RGBD camera is more commonly used [12] [13] [14]. In order to obtain a denser point cloud map with lidar, Shan et al. [15] apply Bayesian generalized kernel inference to terrain elevation and traversability inference. In this way, although theoretically higher accuracy can be obtained, it takes more time. In [16], the authors compute trajectories compliant with curvature and continuity constraints directly on unordered point cloud maps, omitting any kind of explicit surface reconstruction. But with this approach, the accuracy of the planning heavily depends on the density of the map.

B. Contributions

This work offers the following contributions:

1) Based on Rapidly-exploring Random Trees (RRT) [17], a new path planning algorithm named Plane Fitting RRT* (PF-RRT*) for uneven terrain integrating a new terrain assessment method is proposed.
2) A new Plane-fitting based Uneven Terrain Navigation framework (PUTN) for navigation of ground robots on uneven terrain is proposed.
3) Experiments in the real scenario are carried out to verify the real-time performance, effectiveness, and stability of the above algorithms.

II. OVERVIEW OF THE FRAMEWORK

A. Problem Statement

Define $X \subset \mathbb{R}^3$ as the workspace. Let $X_{surf} \subset X$ denote the areas close to the ground, $X_{obs} \subset X$ denote occupied area of space and $X_{free} = X \setminus X_{obs}$. Let $X_{trav} \subset X_{free} \cap X_{surf}$ denote the subspace that is traversable for ground robots.

The formal definition of the problem is as follows: Given the initial and target state $x_{start}, x_{goal} \in X_{trav}$, search a feasible control strategy $\pi^*$. Input $x_{start}$ to $\pi^*$ and it controls the robot to move from $x_{start}$ to $x_{goal}$. $\pi^*$ should satisfy: 1) all kinematic and dynamic constraints; 2) avoiding collision with obstacles along the way; 3) minimizing the time needed to move; 4) minimizing the risk of the robot being unable to maintain a stable posture.

B. System Framework

Fig. 2 shows the structure of the PUTN algorithm. The SLAM module can be implemented by A-LOAM [18], Lego-loam [19], etc. The Global Planning module and the Gaussian Process module generate the sparse path and the dense path, respectively. In addition to following the global path, considering that the loss of accuracy and the hysteresis exist in the two modules in real-time navigation, point cloud updated by lidar in real time will be combined in the Local Planning module to improve the responding ability to avoid the obstacles.

III. IMPLEMENTATION

A. Global Planning

1) Plane Fitting RRT* algorithm:

PF-RRT* is proposed to solve the problem of global path generation in uneven terrain. Its framework is based on informed-RRT* [20], and is presented in Alg.1. The following are the relevant definitions:

- $x \in \mathbb{R}^3$ represents a space point.
- $x \in \mathbb{R}^2$ represents a plane point.
\[ \mathbf{x} \in \mathbb{R}^3 \] represents a space point on the surface of the terrain.
\[ \hat{\mathbf{x}} \in \mathbb{R}^3 \] represents the position of the robot center on the terrain.
\[ S \] represents a path that is a sequence of states.

As shown in Fig.3, different from the traditional RRT, each element in the tree is represented as a node \( N_i = (T_{M_R}, \tau_i) \) instead of a position \( x_i \). \( T_{M_R} \) represents a local plane at the center of the robot footprint and \( \tau_i \in [0,1] \) represents the traversability. A higher value of \( \tau_i \) means harder to traverse. Here 0 represents absolutely traversable and 1 represents absolutely unable to traverse. The detailed calculation will be introduced in Section III-A.2.

Based on the definition above, given two nodes \( N_1, N_2 \), let \( l_{1,2} \) denote the Euclidean distance between them, and the cost function of the line connecting \( N_1 \) and \( N_2 \) is defined as follows:

\[
f(N_1, N_2) = \left( 1 + \omega \left( \frac{1}{1 - \tau_1} + \frac{1}{1 - \tau_2} - 2 \right) \right) \cdot l_{1,2} \tag{1}
\]

Where \( \omega \) is a penalty scale factor. In this way, the algorithm will aim to not only shorten the path but also avoid those paths that are hard to traverse. However, the traversability in this area is assumed to be uniform, which is a rough approach. Further processing is needed to evaluate the traversability of the path more accurately, and it will be described in Section III-B.

Some new subfunctions presented in Alg.1 are described as follows while subfunctions common to the informed-RRT* algorithm can be found in [17] [21] [20]:

- **Pos(\(N\)):** Given a node \( N \), the point stored in it is returned.
- **ProjectToPlane(\(x\)):** Given \( x = [x, y, z]^T \), it returns \( x = [x, y]^T \).
- **ProjectToSurface(\(x\)):** Given \( x = [x, y]^T \). Let \( res \) be the resolution of the grid map, and \( z_l \) be the lower limit of the height of the grid map. It returns \( \hat{x} = [x, y, z]^T \) where \( z \) satisfies

\[
\min_{z} \quad z \\
\text{s.t.} \quad \begin{cases} 
[x, y, z]^T \in X_{\text{obs}} \\
[x, y, z + res]^T \in X_{\text{free}} \\
z - z_l = k \cdot \text{res}, k \in \mathbb{N}
\end{cases}
\tag{2}
\]

- **FitPlane(\(x\)):** Given a point \( \hat{x} \) on the surface, it fits a plane centered on \( \hat{x} \) by combining the information of the global map. This process will be described in detail in Section III-A.2.

Before global planning, the space is downsampled and represented as a spatial grid map as shown in Fig.2. This shortens the time to find the path, but also reduces the accuracy of the path. The problem will be partially solved in III-B.

As illustrated in Alg.1, the PF-RRT* algorithm repetitively adds random samples to search and optimize the solution. Specifically, the PF-RRT* algorithm samples and expands with 2D methods to get a new plane point \( \hat{x}_{\text{new}} \). Then, \( \hat{x}_{\text{new}} \) is first projected to a surface point \( \hat{x}_{\text{new}} \). After that, a new local plane centered on \( \hat{x}_{\text{new}} \) will be fitted. Based on the analysis of the local plane, a new node \( N_{\text{new}} \) is generated, storing a new state \( \hat{x}_{\text{new}} \) and its corresponding traversability. Next, the PF-RRT* algorithm will connect \( N_{\text{new}} \) to the tree and execute the optimization with 3D methods if the validity of \( N_{\text{new}} \) is verified by collision-checking. The PF-RRT* algorithm repeats the operations above until it iterates to the specified maximum iterations, and an optimal solution \( S^* \) will be returned.

Furthermore, the algorithm uses the previous path as a heuristic which ensures the stability of the posture of the robot. In addition to inheriting the fast search and convergence speed of the informed-RRT* algorithm, the PF-RRT* algorithm has the following advantages:

First, many traditional algorithms analyze the entire map before planning. However, as the environment enlarges and updates, the process of analysis will tend to be highly time-consuming, which is lethal in the field of real-time planning. To address this issue, the PF-RRT* algorithm carries out terrain analysis during the expansion of the random sampling tree to avoid useless analysis, which reduces calculation workload and speeds up the response of the algorithm.

Second, while many traditional algorithms adopt a 2D grid map or 2.5D elevation map [4] [5] [12], in which the height value is used as an evaluation index for obstacles or costs, the PF-RRT* algorithm directly adopts 3D map to represent the environment. This enables PF-RRT* to work in more complex 3D environments.
means that the ground is completely unsuitable for vehicles. These three indicators are calculated as follows:

\[
s = k_s \arcsin z_{e^x}
\]

\[
f = \frac{\sum_{j=1}^{N} (e^z \cdot x^j)^4}{N}
\]

\[
\lambda = \begin{cases} 
\frac{1}{r-r_{\min}} & r > r_{\max} \\
0 & r \in [r_{\min}, r_{\max}] \land \text{tr} (\Sigma^T \Sigma) < t_{\text{trace}} \\
\text{otherwise} & 
\end{cases}
\]

where \(k_s\) and \(k_f\) are constant coefficients. \(z_{e^x}\) represents the projection of the plane normal vector on the \(Z\)-axis of the world coordinate system \(M\). \(r\) represents the proportion of vacant parts of the plane. \(r_{\min}\) and \(r_{\max}\) represent the maximum and minimum acceptable vacancy ratio respectively. When the ratio is between the maximum value and the minimum value, whether the vacant points correspond to different situations. When the empty defects are concentrated, it indicates that there may be pits and depressions on the ground.

As shown in Fig. 4, compared with the direct analysis of the landing point, the analysis method of a local plane will take more into account the whole ground situation of the terrain. Since traversability is added to the RRT tree as cost, the RRT tree will automatically stop expanding in difficult-to-pass areas, which makes PF-RRT\(^*\) more efficient than traditional sampling-based terrain analysis.

In order to ensure a smooth transition between adjacent nodes in the trajectory, it is necessary to ensure that the step length \(l_{i,i+1}\) between two nodes satisfies

\[
l_{i,i+1} < \frac{l}{2}
\]
B. Gaussian Process Regression Prediction

After global trajectory generation, a trajectory set $S$ and a random sampling tree $T$ are obtained by the PF-RRT* algorithm. Since in each local plane associated with each node, the traversability is assumed to be a constant value, and the sampling step size and planning time limit the density of generated trajectory points. In order to get a denser path, GPR and linear interpolation are used to achieve this. Dense trajectory points are defined as

$$ S^I = \{ (\xi^l, \tau^l) : i \in \gamma_I \} $$

where, $\{\xi^l : i \in \gamma_I\}$ is obtained by linear interpolation from waypoint set $\{\xi_i : i \in \gamma\}$. Here, the method of Gaussian process regression is proposed to predict the traversability $\{\tau^l : i \in \gamma_I\}$ of interpolation points. The tree node set $D = \{(\xi_i, \tau_i)_{i=1:N}\}$ is used as the training set for Gaussian process regression, and $S^I$ is the test set. The classical radial basis function is adopted as the covariance function

$$ k(\xi, \xi') = \sigma_f^2 \exp \left(-\frac{(\xi - \xi')^2}{2l^2}\right) $$

Using a simple differential-drive model dynamics to track the given path while considering the traversability and obstacles avoidance. The optimization problem has $N$ nodes and spans over 1 second. To simplify the problem, we only consider the 2D planning problem in aerial view. The local terrain is described as the normal vector $n_i$ of the local plane with obstacles $\{x_i^{ob} = (x_i^{ob}, y_i^{ob}) : i = 0, 1, \ldots, N_{ob}\}$. We adopt the differential-drive model on local planes as the system model:

$$ \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} (\|n \times e_x\|) \cos \theta_k \\ (\|n \times e_y\|) \sin \theta_k \end{bmatrix} u_k \Delta t $$

The optimization formulation has the following form:

$$ \begin{align*}
\min_{\{x_k, u_k\}} & \quad \sum_{k=0}^{N-1} \left( \|x_k - x^d_k\|^2 + \lambda \|u_k\|^2 \right) \\
\text{s.t.} & \quad x_{k+1} = f(x_k, u_k) \\
& \quad x_0 = x_{\text{start}} \\
& \quad x_k \in X, u_k \in U \\
& \quad \|x_k - x^d_k\| \geq d_{\text{safe}}
\end{align*} $$

Where $x^d_k$ is two-dimensional vector representing the target points, obtained by $S_d$, $\|x\|_A := \frac{1}{2} \sqrt{x^T Ax}$, and the two positive definite matrices $Q$ and $R$ are respectively coefficients measuring terminal costs and control costs. $\lambda = ((1 - t_{\text{mean}})(1 - \sigma_{\text{mean}}))^{-2}$ is the coefficient of control cost, $t_{\text{mean}}$ and $\sigma_{\text{mean}}$ are obtained by averaging $\tau$ and $\sigma$ in the current local plane, respectively. This coefficient can help the robot slow down when the average traversability is bad or the estimation confidence is low.

IV. EXPERIMENTS

As shown in Fig.7, Scout2.0, a four-wheel-drive mobile robot is used for the experiment. The lidar sensor is OS0-128, and the computational hardware is Intel NUC with an i5 2.4GHz CPU and 16GB memory.

Fig.6 shows the experiment on arch bridge terrain. CasADi is used to solve the NMPC problem, and a C++ library is used to implement GPR, which takes less time than Python library. As is shown in Fig.8, after traversability and the corresponding uncertainty are added to NMPC, the linear velocity of the vehicle decrease where traversability increase. Through this method, the vehicle travels more smoothly in potentially dangerous areas such as undulating ground and the edge of river. In addition to the arch bridge scenario, we also conduct experiments in the forest, flat bridge, steep slope scenarios, respectively, which are shown in Fig.9.

This study also conducted experiments to demonstrate the advantage of the PF-RRT* algorithm in response speed. We compare PF-RRT* with RRT* on normal grid map and grid map analyzed in advance, respectively. The point cloud is analyzed using the same method, which is mentioned in SectionIII-A.2. The point cloud data are collected from real scenarios, shown in Fig.1. The same starting point and goal point are selected for each algorithm. Two indicators are used to estimate the performance of each algorithm:
Fig. 10 shows the result. RRT* on the map analyzed in advance takes the least amount of time. Since there is no terrain assessment, it represents the minimum time for planning using sampling methods. After adding terrain assessment, RRT* obviously increases a lot of time consuming, while PF-RRT* does not. That’s because PF-RRT* only focuses on the terrain where the robot may pass by and completes the task with less terrain analysis. In different scenarios, the accuracy and convergence speed of the algorithm need to be considered comprehensively. PF-RRT* will be faster if the grid map resolution is set to a larger value.

V. CONCLUSION

This study proposes the PUTN algorithm, which is designed for a ground robot to safely and effectively navigate in environments with uneven terrain. A new terrain assessment method is proposed, which integrates different indicators including slope, flatness, and sparsity to evaluate the traversability of terrain. Combined with the informed-RRT* algorithm and this terrain assessment method, a new path planning algorithm, PF-RRT*, is proposed to obtain a sparse global path. By using GPR to regress the RRT tree generated by PF-RRT*, the traversability of the dense
path is obtained. PUTN combines PF-RRT*, GPR, NMPC to complete the fast, stable and safe motion planning of the robot. Experiments are conducted in several typical real-world scenarios. The results verify the advantages of the PF-RRT* algorithm and the practicability of PUTN.

REFERENCES

[1] S. Pütz, J. S. Simón, and J. Hertzberg, “Move base flex a highly flexible navigation framework for mobile robots,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 3416–3421.

[2] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, “Combining optimal control and learning for visual navigation in novel environments,” in Conference on Robot Learning. PMLR, 2020, pp. 420–429.

[3] V. S. Kalogeiton, K. Ioannidis, G. C. Sirakoulis, and E. B. Kosmatopoulos, “Real-time active slam and obstacle avoidance for an autonomous robot based on stereo vision,” Cybernetics and Systems, vol. 50, no. 3, pp. 239–260, 2019.

[4] W. Yuan, Z. Li, and C.-Y. Su, “Multisensor-based navigation and control of a mobile service robot,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 4, pp. 2624–2634, 2019.

[5] A. Xiao, H. Luan, Z. Zhao, Y. Hong, J. Zhao, J. Wang, and M. Q.-H. Meng, “Robotic autonomous trolley collection with progressive perception and nonlinear model predictive control,” arXiv preprint arXiv:2110.06648, 2021.

[6] C. Wang, J. Wang, C. Li, D. Ho, J. Cheng, T. Yan, L. Meng, and M. Q.-H. Meng, “Safe and robust mobile robot navigation in uneven indoor environments,” Sensors, vol. 19, no. 13, p. 2993, 2019.

[7] P. Kim, J. Chen, J. Kim, and Y. K. Cho, “Slam-driven intelligent autonomous mobile robot navigation for construction applications,” in Workshop of the European Group for Intelligent Computing in Engineering. Springer, 2018, pp. 254–269.

[8] I. Jeong, Y. Jang, J. Park, and Y. K. Cho, “Motion planning of mobile robots for autonomous navigation on uneven ground surfaces,” Journal of Computing in Civil Engineering, vol. 35, no. 3, p. 04021001, 2021.

[9] X. Xiao, J. Biswas, and P. Stone, “Learning inverse kinodynamics for accurate high-speed off-road navigation on unstructured terrain,” IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 6054–6060, 2021.

[10] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi, “Step: Stochastic traversability evaluation and planning for safe off-road navigation,” arXiv preprint arXiv:2103.02828, 2021.

[11] P. Papadakis, M. Gianni, M. Pizzoli, and F. Pirri, “Constraint-free topological mapping and path planning by maxima detection of the kernel spatial clearance density,” in International Conference on Pattern Recognition Application and Methods, 2012.

[12] P. Fankhauser, M. Bloesch, and M. Hutter, “Probabilistic terrain mapping for mobile robots with uncertain localization,” IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3019–3026, 2018.

[13] L. Campos-Macias, R. Aldana-López, R. de la Guardia, J. J. Parra-Vilchis, and D. Gómez-Gutiérrez, “Autonomous navigation of mavs in unknown cluttered environments,” Journal of Field Robotics, vol. 38, no. 2, pp. 307–326, 2021.

[14] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, “Robust and efficient quadrotor trajectory generation for fast autonomous flight,” IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 3529–3536, 2019.

[15] T. Shan, J. Wang, B. Englot, and K. Doherty, “Bayesian generalized kernel inference for terrain traversability mapping,” in Conference on Robot Learning. PMLR, 2018, pp. 829–838.

[16] P. Krüsi, P. Furgale, M. Bosse, and R. Siegwart, “Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments,” Journal of Field Robotics, vol. 34, no. 5, pp. 940–984, 2017.

[17] S. M. LaValle and J. J. Kuffner Jr, “Randomized motion planning,” The international journal of robotics research, vol. 20, no. 5, pp. 378–400, 2001.

[18] J. Zhang and S. Singh, “Loam: Lidar odometry and mapping in real-time,” in Robotics: Science and Systems, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[19] T. Shan and B. Englot, “Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 4758–4765.

[20] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014, pp. 3097–3104.

[21] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” The international journal of robotics research, vol. 30, no. 7, pp. 846–894, 2011.

[22] A. Chilian and H. Hirschmüller, “Stereo camera based navigation of mobile robots on rough terrain,” in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009, pp. 4571–4576.