Efficient Pedestrian Following by Quadruped Robots

Guangyao Zhai, Zhen Zhang, Xin Kong, Yong Liu

Abstract— **Legged robots have superior terrain adaptability and flexible movement capabilities than traditional wheeled robots. In this work, we use a quadruped robot as an example of legged robots to complete a pedestrian-following task in challenging scenarios. The whole system consists of two modules: the perception and planning module, relying on the various onboard sensors.**

I. Introduction

Sometimes robots are required to assist humans in complicated environments, like narrow corridor scenes. Under these circumstances, space is always confined and narrow, where the limited mobility of the wheeled robot restricts its performance. Compared with traditional wheeled robots, quadruped robots represent high mobility and dynamic motion capability as a representative of legged robots. There are tons of tasks for the quadruped robots to complete. One of them is the pedestrian following. It can be widely used in agriculture, industry, military, and other diverse domains.

Pedestrian following can be challenging for a quadruped robot to complete, especially in constraint scenarios. To begin with, due to the lack of a prior map, the robot needs to update its surrounding information and the dynamic location of the followed pedestrian online by onboard sensors. Both the non-stationary goal and the dynamic environment require real-time perception feedback and trajectory generation. Besides, although quadruped robots have similar locomotion abilities as the normal omnidirectional robots do, the kinodynamics model of quadruped robots is more complicated because of the limits existing in the actuators. Therefore, the design of path planning algorithms should be exclusive.

We introduce an efficient mobile system that can complete the following task in challenging scenarios in this work. The whole system can be divided into two modules, which are a real-time perception module and an efficient path planning module. The perception module consists of two blocks, including a Single-Object Tracking (SOT) block and an obstacle detection block. The main contributions of this system are:

- Propose a real-time and robust LiDAR-IMU SOT method which can provide an accurate location of the followed object insensitive to occlusion.
- Present a novel two-step, optimization-based, receding horizon method for quadruped robots to follow the person in confined environments.

The authors are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, P. R. China.

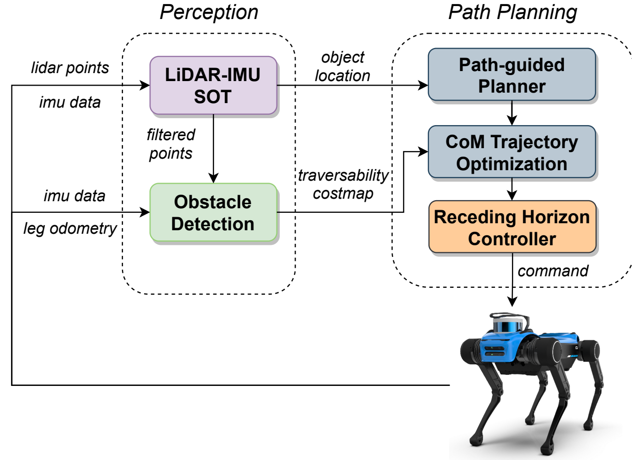

Fig. 1: The overview of our scheme for efficient navigation of quadruped robots.

II. Methodology

As shown in Fig. 1, our system includes a perception module and a path planning module. All the data required by the perception module are collected by sensors mounted on the robot body.

A. Perception

The perception module realizes two functions, providing the pedestrian's location and a traversability costmap. Both of them are then sent to the path planning module.

LiDAR-IMU SOT. Sequenced localization is a one-shot tracking problem in nature. We solve this through two parts: *1) Initialization.* This part is used to specify the initial Region of Interest (RoI). For a specific category like humans, its shape and size remain relatively constant at different moments. Therefore, we can initialize a point along with a tracking bounding box centered on it which can cover the pedestrian.
2) Segmentation and iterative tracking. As the core of the proposed SOT algorithm, this part achieves high real-time performance through a fast iterative process. After the initialization, we use min-cut algorithm belonging to the Graph Cut field to deal with the foreground and background segmentation in the RoI. Then, we can obtain the tracked pedestrian by comparing the size of the foreground and background points. Regularly, some inside points will move out of the bounding box due to the movement of the pedestrian, causing a shift between the center of the bounding box and the pedestrian center. We can make two centers align with each other again by iteratively min-cut and update the

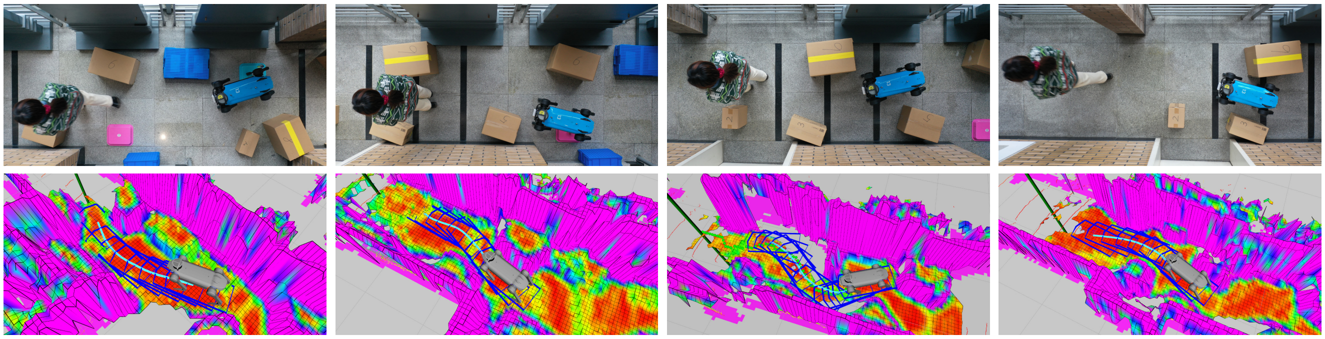

Fig. 2: Image flows of following a pedestrian in the confined corridor (top) and the corresponding traversability costmaps (bottom). Areas with purple-like color are treated as obstacles with a high probability. The planned local trajectory is shown as a sequence of blue polygon boxes on the bottom.

position of the bounding box. Due to the rotation disturbance of the used quadruped robot platform, we compensate the center point by integrating the IMU measurements between two LiDAR timestamps.

Obstacle Detection. This block is responsible for local mapping and walkable area extraction.

1) Local mapping. In order to save the computing resources of the robot platform, we chose FLOAM, a lightweight version of LOAM [1], to form this part, as it reduces the computational cost by three times compared with LOAM. We filter out the tracked points to build the local map so that the tracked pedestrian is not considered for obstacle extraction.

2) Walkable area extraction. This part provides a traversability costmap to the path planning module based on the prior local map. The costmap is built on the universal grid map library [2]. We create a 2.5D grid map by estimating terrain traversability by three characteristics in each 2D BEV grid: the roughness r, the slope s, and the step height h, The traversability cost C_t of each cell is formulated as follows:

$$C_t = \alpha \frac{r}{r_{\text{thre}}} + \beta \frac{s}{s_{\text{thre}}} + \gamma \frac{h}{h_{\text{thre}}} \tag{1}$$

r_{thre}, s_{thre}, and h_{thre}, are the thresholds of each characteristics. α, β, and γ are weights, which add up to 1. The roughness is the height variance in the grid, and the slope is the sine value of the angle between the maximum plane normal vector of the grid and the horizontal plane normal vector. As C_t increases, the probability of this grid being treated as an obstacle is higher.

B. Path Planning

Our planning module receives the location of the followed pedestrian as a local goal then firstly employs a geometric path planner to find a coarse path. After the path generation, we perform an optimization procedure iteratively refines the Center of Mass (CoM) trajectory of the robot. The planning and the optimization mentioned above are repeated once the costmap updates, which allows avoiding collision and improves the movement robustness.

1) Path-Guided Planner. We first adopt a variant A* algorithm [3] to search for a rough path guiding to the local goal in the 2D costmap because of its search efficiency.

2) Trajectory Generation. We treat the quadruped robot as a rigid body and project it onto the 2D plane as a rectangular footprint. This method fully considers the robot's dynamic motion capability so that the robot can flexibly adjust its orientation to pass through narrow passages. We define trajectory generation as an optimization problem taking the kinodynamic constraints into account and solve it using nonlinear programming. The objective functions include minimizes the time and minimizes the sum of accelerations of the trajectory. In addition, the kinodynamics constraints of the robot as well as the obstacles constraints in the surrounding environment are considered, and they are added to the objective function as additional penalty terms to improve the efficiency of fast online solvers. Thus it can be formulated as multiple objective functions, and each item is given a weight:

$$\mathcal{B}^* = \min_{\mathcal{B}} \{ \sum_{i=0}^{n-1} (\mathbf{a}^\top \mathbf{W_1} \mathbf{a} + \mathbf{b}^\top \mathbf{W_2} \mathbf{b} + \mathbf{c}^\top \mathbf{W_3} \mathbf{c}) \} \tag{2}$$

where, $\mathbf{a} = [\Delta T_i, \|a_i\|]^\top$, $\mathbf{b} = [\sum_{k=1}^{P} F_1(d_k)]^\top$, $\mathbf{c} = [F_2(v_i), F_2(\omega_i), F_2(f_i)]^\top$. $\mathcal{B}^*$ denotes the optimal path solution, $\mathbf{a}$ is the the associated term for optimizing the time and acceleration of the trajectory, $\mathbf{b}$ and $\mathbf{c}$ represent the penalty functions of obstacles constraints and kinodynamic constraints. $\mathbf{W_1}$,$\mathbf{W_2}$ and $\mathbf{W_3}$ are symmetric and positive definite matrices to weight and balance these factors. Finally, we generate a set of steering commands from $\mathcal{B}^*$ by utilizing a receding horizon controller.

III. Experiments

The scheme has been experimentally validated in a real scene: the narrow corridor with cluttered blocks. Fig. 2 shows the whole process of the robot Jueying[1] performing pedestrian-following task, clearly showing that the robot successfully follows a pedestrian and adjusts its body to avoid the collision.

References

[1] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, 2014.

[2] P. Fankhauser and M. Hutter, "A universal grid map library: Implementation and use case for rough terrain navigation," in *Robot Operating System (ROS)*, 2016.

[3] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[1] http://www.deeprobotics.cn