

Representation of 3D Structure for Path Planning with Biped Wall-Climbing Robots

Guozhen Cai, Weilin Lin, Shangbiao Wei, Shichao Gu, Haifei Zhu* and Yisheng Guan

Biomimetic and Intelligent Robotics Lab (BIRL)
School of Electromechanical Engineering, Guangdong University of Technology
Guangzhou, Guangdong Province, China, 510006

Abstract—For biped wall-climbing robots, mapping and extraction of the parametric expression map denote challenging tasks, especially for the environments where lidar measurements degrade. To build a representation of the environment for global path planning, a three-dimensional (3D) planar structure modelling and parameterization system is proposed. The system includes the two following tasks: 1) mapping by a lidar-IMU fusion method in which point cloud registration algorithm tracks the lidar poses with the initial transformation matrix provided by IMU integration; 2) extracting the vertexes of planes that represent the parametric expression of the 3D planar structure. Experiments showed that our mapping algorithm was more robust than others in the lidar-degraded environment. Besides, the precision analysis verified that the parametric expression map was accurate enough to be used for path planning of biped wall-climbing robots.

Index Terms—lidar-degraded, lidar-IMU fusion, parametric expression map

I. INTRODUCTION

Given an accurate map, a wall-climbing robot can perform tasks in a 3D planar structure such as inspection and maintenance. A 3D planar structure refers specifically the steel bridge archways, carriages of high-speed rails and ship cabins. These environments are characterized by repetitive geometry features, a lack of visual features and limited space. A biped wall-climbing robot [1] has been developed in our lab, as shown in Fig. 1. This robot has an excellent mobility on smooth walls. Besides, it can correct the pose error caused by the accumulation of map error and motion error [2]. As long as the map error is less than a specified threshold, the robot can use it to implement autonomous climbing.

The Simultaneous Localization and Mapping(SLAM) has been widely used for mapping in 2D, 2.5D, or 3D environment using a mobile robot equipped with sensors such as lidars and stereo cameras. However, the focus of the above work is on the localization rather than on constructing highly-accurate 3D maps [3]. When a wall-climbing robot performs mapping using SLAM, sensors can be placed on or outside the climbing robot. In [4], a multi-agent system is introduced in mapping. The ground robots were equipped with RGB cameras and acted as mobile observers with wide coverage to detect and estimate the structure to be

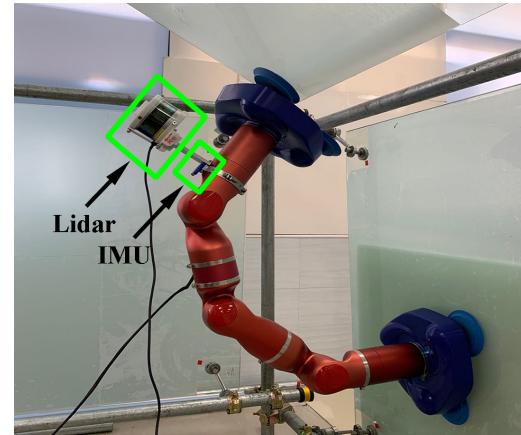


Fig. 1: The biped wall-climbing robot equipped with a Velodyne VLP-16 and LPMS IMU while it is climbing on the wall.

climbed on. However, this method complicated the problem and increased the hardware cost. In [5], a wall-climbing robot equipped with a camera performed mapping in the steel bridge archways. The camera poses could be computed using forward kinematics so that it avoided the problem of a lack of visual and geometry environmental features in registration. In [6], the climbing robot used the RTAB-map to construct a point cloud map of a truss-style structure. The experimental environments presented in [6] were rich in visual features. To the best of our knowledge, rare researches similar to ours were conducted, in which a climbing robot equipped with the sensor performed mapping without using kinematic model in an environment that lacked visual features.

The lidar observes only a few usable points on a 3D planar structure, which means that lidar degrades. The observed points are mainly from the side walls, and only a small portion of points are observed from the ceiling and floor. In this case, the matched lidar features can easily lead to the ill-constrained pose estimation. As stated in [7], an IMU-Kinect integrated sensor system was prone to fall into degenerate states owing to its narrow field-of-view. In [8], given the prior from the lidar-IMU odometry, a rotational constrained refinement method was proposed to further optimize the final sensor poses and the generated point cloud map. However, this method required to excite the IMU sufficiently to ini-

* Corresponding author (hfzhu@gdut.edu.cn). This work is partially supported by the Natural Science Foundation of China (Grant No. 51605096, 51705086), and the Frontier and Key Technology Innovation Funds of Guangdong Province (Grant No. 2017B050506008 and 2019A050503011).

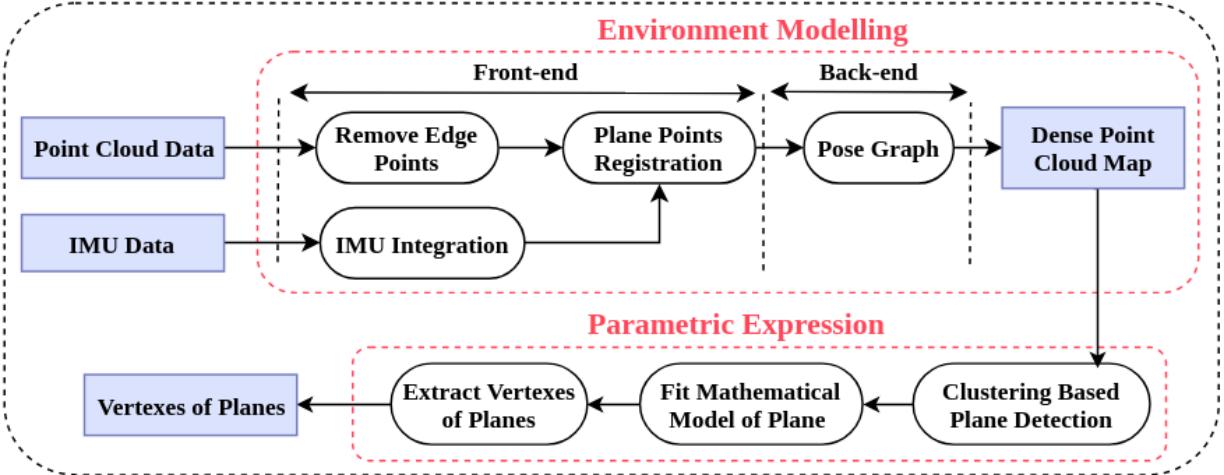


Fig. 2: The proposed environment modelling and parameterization framework. The blue blocks indicate data flow. The red dotted frames indicate the environment modelling and extraction of parametric expression map.

tialize it online, and this could last for a long time since our robot moved slowly.

For a biped wall-climbing robot, expressing an unknown 3D planar structure in a form that is convenient for path planning is a key problem. Parametric expression of the environment was known to provide a memory-efficient expression [9]. Also, it was an ideal alternative for path planning [10]. Therefore, in this work, we extract the parametric expression from the point cloud map.

In this paper, we present an environment modelling and parameterization system to build a representation of a 3D planar structure used for path planning of a biped wall-climbing robot. In our experimental environment, ill-constrained poses estimation often occur for lack of constraint in feature points matching. To overcome this problem, we use the Normal Distributions Transform (NDT) algorithm to register raw point cloud instead of feature points. The NDT takes local surface structures into account rather than matching individual points. Using the initial transformation matrix obtained by the IMU integration, the NDT can register point cloud with high precision. After constructing a dense point cloud map, the planes are detected, and then vertexes of the planes are extracted.

II. PROBLEM STATEMENT

The framework of our system is presented in Fig. 2. The system can be divided into two steps, environment modelling and parametric expression. In addition, the extrinsic calibration parameters T_B^L of a lidar-IMU system need to be recovered using the method proposed in [11].

In the first step, a lidar-IMU fusion method is used to track the sensor pose to overcome the lidar-degraded problem. Assumes that $P_t = \{p_1, p_2, \dots, p_n\}$ is the point cloud acquired at time t , where p_i is the i^{th} point in P_t . The IMU raw data $I_{i,j}$ are collected from the last timestamp i to the current timestamp j . A pose graph is constructed as the back-end to optimize the sensor poses.

In the second step, the key problem of how to express a 3D planar structure is solved efficiently. Without the loss of generality, we consider that planes are polygonal and can be parameterized by their vertexes. Let w_i represents a plane, and E_w represents a 3D planar structure, then they can be respectively expressed by:

$$w_i = \{p_j | j = 1, 2, \dots, N, N \geq 3\}, \quad (1)$$

$$E_w = \{w_i | i = 1, 2, \dots, N_w, N_w > 0\}, \quad (2)$$

where P_j denotes the vertexes of a plane, N denotes the number of vertexes of the plane, N_w denotes the number of planes in a 3D planar structure. The three steps, including plane segmentation, plane mathematical model fitting and vertexes extraction, are implemented to obtain the parametric expression map from the point cloud map.

III. IMPLEMENTATION

A. Environment Modelling

1) Before P_j arrives, IMU states are updated via Euler integration iteratively to obtain the initial transformation matrix.

2) When the latest point cloud P_j is received, an edge feature extraction step is applied to remove the edge points.

3) The latest point cloud is registered to the target point cloud. The pose graph is constructed using constraints between the key point cloud frames and solved by g2o to optimize the results.

Details of the three steps are presented in the following.

The Euler integration of IMU data between P_i and P_j can update the previous IMU state $X_{B_i}^W$ to the current IMU state $X_{B_j}^W$ in the world frame, which can provide the initial transformation matrix for registration algorithms. The IMU state is given by:

$$X_{B_i}^W = [p_{B_i}^{WT} \ v_{B_i}^{WT} \ q_{B_i}^{WT}]^T, \quad (3)$$

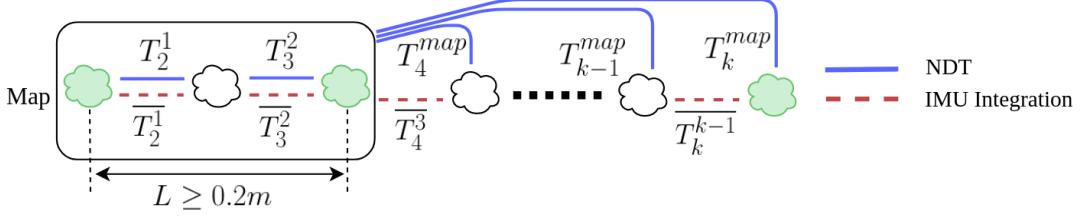


Fig. 3: The front-end of lidar-IMU odometry. The red dashed line represents the IMU integration that provides an initial transformation matrix T_k^{k-1} between two point clouds. The blue line denotes the NDT that computes the refined transformation matrix T_k^{k-1} or T_k^{map} using the initial transformation matrix. The white cloud and green cloud represent the consecutive point cloud frame and key point cloud frame, respectively. The distance between two key point cloud frames must be larger than 0.2 m. First, the latest point cloud is registered with the previous point cloud. The latest point cloud will be registered with the map when the map contains sufficient key point cloud frames.

where $p_{B_i}^W$, $v_{B_i}^W$ and $q_{B_i}^W$ are the position, velocity and orientation of the body frame with respect to the world frame respectively. Let $T_j^W \in SE3$ denotes the IMU pose at timestamp j in the world frame. The initial transformation matrix in the IMU frame can be obtained by:

$$T_j^i = T_i^W T_j^W, \quad (4)$$

The edge feature points are extracted and removed. 3D planar structures contain some non-planar structures such as trusses, stiffeners, and rivets. These structures are used to connect and support multiple planes. However, in the point cloud map, these points can be noisy points when planes point cloud are segmented. To this end, we remove these non-planar points from the raw point cloud. The method of distinguishing the non-planar points in the raw point cloud is similar to the method used in [12]. Namely, the raw point cloud P_t is first projected onto a range image. The resolution of the projected range image is 1800×16 . Each valid point p_i in P_t is now represented by a unique pixel in the range image. The range value r_i that is associated to p_i represents the Euclidean distance from the corresponding point p_i to the sensor. Let S denotes the set of adjacent points of p_i from the same row of the range image. Half of the points in S are on each side of p_i . We set $|S|$ to 10 and evaluate the roughness c of point p_i in S by:

$$c = \frac{1}{|S| \cdot \|r_i\|} \left\| \sum_{j \in S, j \neq i} (r_j - r_i) \right\|, \quad (5)$$

Points in a row are sorted based on the value of c , and points whose roughness is larger than the threshold c_{th} are labeled as the non-planar points and removed from P_t . Let \bar{P}_t denotes the point cloud after the non-planar points are removed. A visualization of the point cloud before and after removing non-planar points is shown in Fig. 4.

The registration algorithm is used to track the lidar poses, and a pose-graph is constructed to optimize the lidar poses. The main idea of the NDT is to represent the point cloud as a set of Gaussian probability distributions. Thus, this is a more memory-efficient representation and can provide more accurate results compared to the modern implementation of the Iterative Closest Point (ICP) method when faced with the

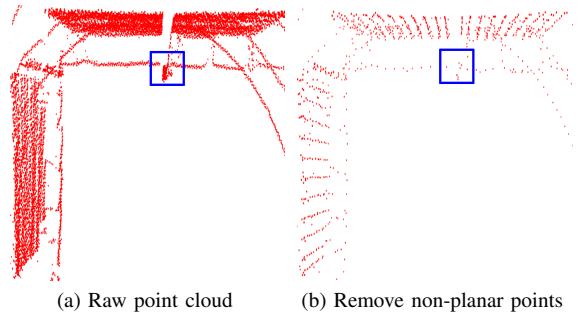


Fig. 4: The point cloud before and after removing the non-planar points. The blue box indicates the non-planar points.

point cloud that contains a little overlap [13]. Details of the tracking process of lidar pose using the NDT are provided in the following.

First, the latest point cloud, called the source cloud, is registered to the last point cloud, called the target cloud, and the IMU integration is used to provide the initial transformation matrix. The source cloud is downsampled and transformed to the coordinate of the target cloud using the NDT result, and it is stored as a map in the memory. The registration between two consecutive frames of the point cloud can lead to error accumulation. To avoid error accumulations, when the map contains at least three key point cloud frames, it is used as the target cloud, and the latest point cloud is registered to the map. The first point cloud is denoted as the first key point cloud frame. The latest point cloud is denoted as the key point cloud frame when the distance L between it and the previous key point cloud frame is more than 0.2 m. A visualization of the front-end is shown in Fig. 3. In the back-end, the constraints between the key point cloud frames are used to construct a pose-graph. The graph is optimized using General Graphic Optimization(g2o).

B. Extraction of Parametric Expression Map

The environment modelling outputs dense point cloud maps shown in Figs. 7(b) and 7(d). In the extraction of parametric expression map, planes in the point cloud are detected, and the vertexes of planes are extracted. The

parametric expression map improves the space-saving to a great extent compared with the point cloud expression.

The Point Cloud Library (PCL) is used to implement the preprocessing operations. First, a voxel grid filter and a statistical outlier removal filter are used to downsample and remove noisy points, respectively. Then, the normal based region growing segmentation algorithm is used to detect planes in the point cloud map. This algorithm merges points that are close enough in terms of the smoothness constraint. Thereby, the output of this algorithm represents a set of clusters, where each cluster denotes a set of points that are considered to be a part of the same smooth surface. The result of the plane segmentation process is shown in Fig. 5. Each color represents a cluster, wherein each color represents a cluster, and red points represent noisy points that do not belong to any clusters.

The extraction of vertexes of planes is described in detail in the following. First, a RANSAC algorithm is used to fit the plane model to the plane point cloud. Next, the extracted plane point cloud is projected to the plane generated by its mathematical model to filter the error along the normal direction. Then, a concave hull algorithm is used to obtain plane contour. Finally, the Douglas Peucker algorithm is used to decrease the number of points on the contour to get an approximate polygon. The obtained parametric expression map is shown in Fig. 6.

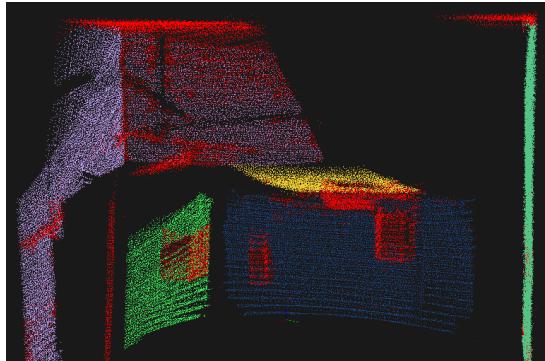


Fig. 5: The plane detection result.

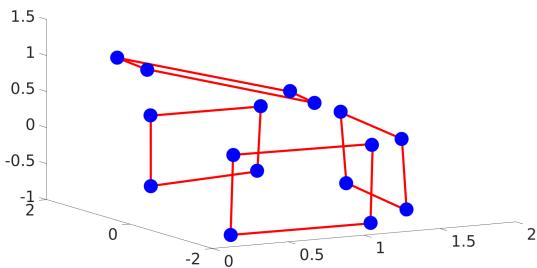


Fig. 6: The parametric expression map. Red polygons represent a 3D planar structure. Blue points represent vertexes of planes, which are the parametric expression of the 3D planar structure.

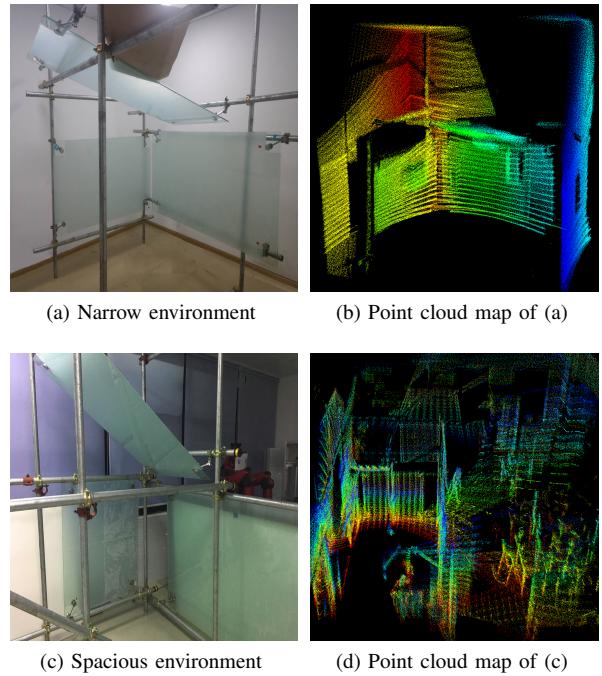


Fig. 7: Two different experimental environments. (a)(c) Images of two environments. (b) The point cloud map of (a). (d) The point cloud map of (c).

IV. EXPERIMENTS

Two experiments were conducted, one to verify the ability of environment modelling in a lidar-degraded case and other common cases, and another to evaluate the accuracy of the parametric expression of a 3D planar structure.

A. Mapping Test

Two different environments in which a robot could perform the tasks were used in the experiments, as shown in Figs. 7(a) and 7(c). The first environment was a narrow room with a size of $3\text{ m} \times 3\text{ m} \times 3\text{ m}$. This environment was characterized by repetitive geometry features, a lack of visual features, and limited space, like a bridge archway tunnel. In this environment, almost all points that lidar has observed are on vertical walls, and only a few points are on the ceiling and ground. The registration of these point clouds is prone to fall into ill-constrained result. The second environment was a spacious indoor environment with many manufacture structures, which provided many geometry and visual features. This environment is common in SLAM for mobile robots.

We collected ten datasets for each environment and compared our algorithm with the other lidar-based SLAM methods, including the LIO-mapping, ALOAM, and Lego-LOAM. The two point cloud maps generated by our methods are shown in Figs. 7(b) and 7(d). The success rate of the mapping test is given in Table I. The reason for the failure of our method was that in some cases, the robot changed motion direction rapidly, which had a bad effect on the IMU integration. The LIO-mapping required to initialize the

IMU online, which required sufficient 3D motion to excite the IMU. However, our robot moved slowly, so it took a long time to complete the IMU initialization. The ALOAM and Lego-LOAM extracted edge points and matched them to register the point cloud. However, in the lidar-degraded cases, the edge points always laid on the intersection lines of the wall. These points denoted unstable points since they lacked the constraints along the z -axis, around the x -axis and y -axis. In addition, the Lego-LOAM was unsuited to the 3D mapping since it assumed ground points exist in every raw point cloud. It should be noted that our algorithm had a longer point cloud registration time than the other two methods because our algorithm used more points to register the point cloud.

TABLE I: Mapping success rates in two environments

Environment type	Our method	LIO-Mapping	ALOAM	Lego-LOAM
Narrow	60%	40%	0%	-
Spacious	80%	50%	100%	-

B. Accuracy of Parametric Expression Map

In this experiment, three evaluation indexes were proposed to assess the accuracy of parametric expression maps, and a precision analysis comparing the parametric expression of the 3D planar structure with the ground truth(GT) was conducted. The values of the three evaluation indexes had to be lower than the thresholds so that the autonomous aligning system of the biped wall-climbing robot could correct the pose error to align the sucker with the wall.

The three evaluation indexes were: 1) distance dc_i - this was a distance between the central point of the GT plane and the central point of the estimated plane; dc_{thres} represented the threshold having the value of 0.2 m; 2) the maximum distance dv_i - this was a distance between the GT plane and the estimated plane; dv_{thres} represented the threshold, and its value was also 0.2 m; 3) angle a_i - this was an angle between the normal to the GT plane and the normal to the estimated plane; a_{thres} represented the threshold, and its value was 15° . Fig. 8 provides a diagram of the three evaluation indexes.

For the first evaluation index, the Euler distance dc_i between the GT central point c_i^{gt} and the estimated central point c_i is computed by:

$$c_i = \frac{1}{N} \sum_{j=1}^N p_j. \quad (6)$$

For the second evaluation index, distance d_j between each vertex of an estimated plane to the GT plane was calculated by (7).

$$(p_j - c_i^{gt}) n_i^{gt} + d_j = 0, \quad (7)$$

where c_i^{gt} and n_i^{gt} denoted the central point and normal of a GT plane, respectively. The maximum distance between the estimated plane and the GT plane dv_i was the maximum d_j .

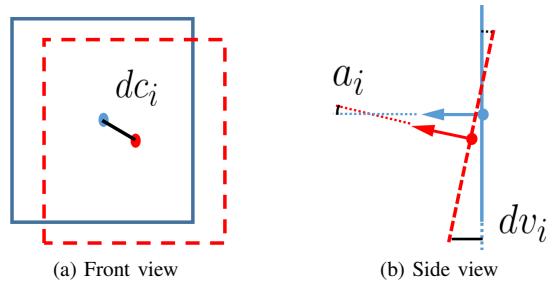


Fig. 8: Three evaluation indexes. The red-dotted frame denotes the estimated plane, and the blue-line frame denotes the GT plane. dc_i is the distance between the two central points. dv_i is the maximum distance between the two planes and a_i is the angle between the two normals.

For the third evaluation index, angle a_i between the normal to the estimated plane and the normal to the GT plane was obtained by:

$$a_i = \arccos(n_i \cdot n_i^{gt}). \quad (8)$$

Five parametric expression maps that have three planes were generated for two environments, respectively. The three evaluation indexes were calculated for each planes, and the results were represented as the boxplot. A boxplot showed values of an evaluation index calculated by our algorithm and others. The precision analysis results of the parametric expression maps of the narrow environment using our method and LIO-mapping are presented in Fig. 9. The precision analysis results of the parametric expression maps of the spacious environment using our method, LIO-mapping, and ALOAM are shown in Fig. 10. For both experimental environments, the boxplots show that the maximal error of the three evaluation indexes are smaller than the three thresholds that are 0.2 m, 0.2 m and 15° , respectively. Besides, the median of three evaluation indexes show that our method achieve higher accuracy in environment modelling than the others. The precision analysis results show that the parametric expression maps are accurate enough to be used for path planning.

V. CONCLUSIONS

In this paper, an environment modelling and parameterization expression system is proposed to build a representation of 3D planar structures. In the environment modelling process, the lidar-IMU fusion odometry is used to estimate sensor poses. The fusion odometry performs well, even in a lidar-degraded environment. In the plane detection and parametric expression processes, the plane point cloud is segmented from the dense point cloud map, and vertexes of the plane are extracted to represent a 3D planar structure. The parametric expression of the environment is known to provide a memory-efficient expression, and is also an ideal alternative to path planning. Experimental results show that the precision of the parametric expression of a 3D planar structure meets the requirements proposed in our previous work. However, our algorithm is very time-consuming because the feature extraction is omitted in the point cloud

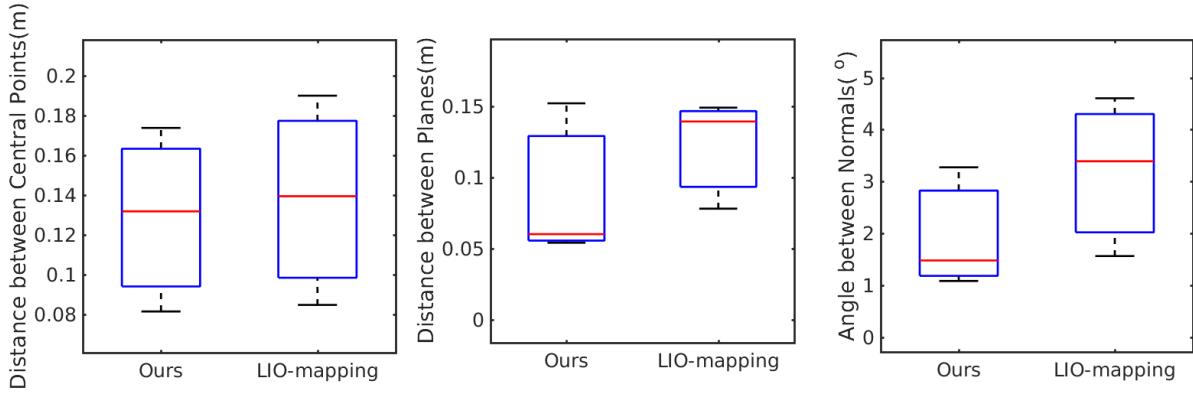


Fig. 9: The error of parametric expression map in narrow environment.

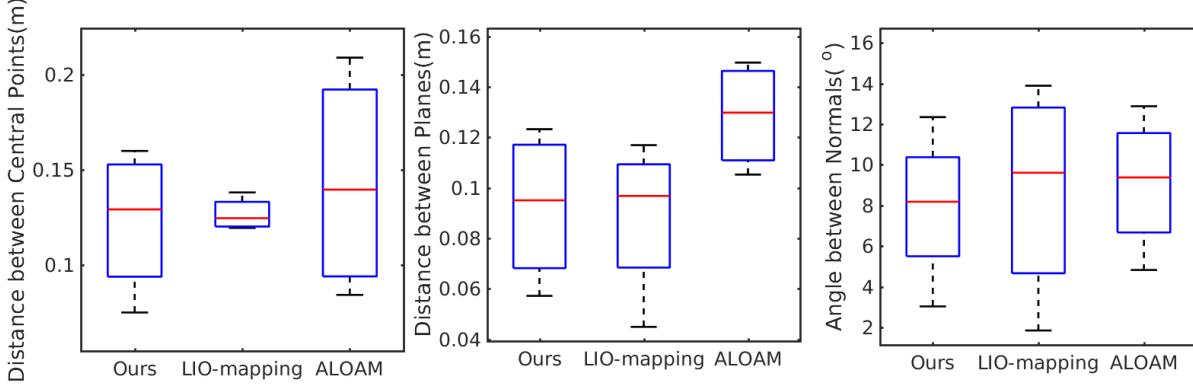


Fig. 10: The error of parametric expression map in spacious environment.

registration. In our future work, we will study the stable features in a lidar-degraded environment.

REFERENCES

- [1] Yisheng Guan, Haifei Zhu, Wenqiang Wu, Xuefeng Zhou, Li Jiang, Chuanwu Cai, Lianmeng Zhang, and Hong Zhang. A modular biped wall-climbing robot with high mobility and manipulating function. *IEEE/ASME transactions on mechatronics*, 18(6):1787–1798, 2012.
- [2] Haifei Zhu, Yisheng Guan, Wenqiang Wu, Lianmeng Zhang, Xuefeng Zhou, and Hong Zhang. Autonomous pose detection and alignment of suction modules of a biped wall-climbing robot. *IEEE/ASME transactions on mechatronics*, 20(2):653–662, 2014.
- [3] Gavin Paul, Shuyuan Mao, Liyang Liu, and Rong Xiong. Mapping repetitive structural tunnel environments for a biologically-inspired climbing robot. In *ASSISTIVE ROBOTICS: Proceedings of the 18th International Conference on CLAWAR 2015*, pages 325–333. World Scientific, 2016.
- [4] Mahmoud Tavakoli, Gonçlo Cabrita, Ricardo Faria, Lino Marques, and Anibal T de Almeida. Cooperative multi-agent mapping of three-dimensional structures for pipeline inspection applications. *The International Journal of Robotics Research*, 31(12):1489–1503, 2012.
- [5] Gavin Paul, Phillip Quin, Andrew Wing Keung To, and Dikai Liu. A sliding window approach to exploration for 3d map building using a biologically inspired bridge inspection robot. In *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 1097–1102. IEEE, 2015.
- [6] Weinan Chen, Shichao Gu, Lei Zhu, Hong Zhang, Haifei Zhu, and Yisheng Guan. Representation of truss-style structures for autonomous climbing of biped pole-climbing robots. *Robotics and Autonomous Systems*, 101:126–137, 2018.
- [7] HyunGi Cho, Suyong Yeon, Hyunga Choi, and Nakju Doh. Detection and compensation of degeneracy cases for imu-kinect integrated continuous slam with plane features. *Sensors*, 18(4):935, 2018.
- [8] Haoyang Ye, Yuying Chen, and Ming Liu. Tightly coupled 3d lidar inertial odometry and mapping. *arXiv preprint arXiv:1904.06993*, 2019.
- [9] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.
- [10] Haifei Zhu, Yisheng Guan, Shengjun Chen, Manjia Su, and Hong Zhang. Single-step collision-free trajectory planning of biped climbing robots in spatial trusses. *Robotics and biomimetics*, 3(1):1, 2016.
- [11] C. Le Gentil, T. Vidal-Calleja, and S. Huang. 3d lidar-imu calibration based on upsampled preintegrated measurements for motion distortion correction. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2149–2155, May 2018.
- [12] 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)*, pages 4758–4765, Oct 2018.
- [13] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk. Beyond points: Evaluating recent 3d scan-matching algorithms. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3631–3637, May 2015.