2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Region-determined localization method for unmanned ground vehicle under pole-like feature environment

Yu-Hsiang Lai, Chia-Yun Chuang, Yu-Qiang Chen, and Feng-Li Lian, *Member, IEEE*

***Abstract*—In this paper, a region-determined localization method applied for unmanned ground vehicles (UGVs) is presented. The method aims to solve GNSS-denied localization problem using pole-like feature such as trees or street lights. The approach includes three parts: mapping, bounding, and localization. To map and reconstruct the environment, the hector mapping approach and circle-fitting method are adopted for the occupancy mapping and feature mapping. To bound out the available working region, we define the intersection area of features' enlarged radius and desired operating area as negative and positive virtual boundaries. While the robot is cruising, the likelihood detection method is adopted for obstacle searching and comparing. Using the detection's searching results as feedback reference, the Extended Kalman Filter (EKF) can modify the shifting between the GNSS signal and the true waypoints of the mowing robot. Three cruising demonstrations are presented to show the mapping and optimizing results. Different cases of demonstration represent different situations and potential issues.**

I. Introduction

In the past few years, UGVs are becoming more and more popular in various fields of applications. Maintenance tasks such as lawn mowing, environment monitoring, or harvesting are gradually replaced by human to robots. Most of those applications, especially in the outdoor environment, are time-consuming, heavy workload, and even hazardous for human.

The traditional way to do those tasks is using underground wires for limiting the working region. Robot will move randomly in the desired area. However, using the strategy, the robots' tracks are inefficient and may miss some desired regions while moving traversal. Therefore, a method that can automatically decide the virtual boundary and follow the desired waypoints should be developed.

To follow the desired waypoints, the localization is an important issue for outdoor maintenance tasks. The outdoor UGV robot often adopts an Inertial Measurement Unit (IMU), or adopt Global Navigation Satellite System (GNSS) for navigation. However, GNSS information and IMU signal are not always reliable. For outdoor localization, GNSS signals can provide high accuracy. Yet, the update frequency is slow, which may cause signal missing problem. Furthermore, if some buildings or shade are covering the top of the robot, the GNSS signal will also be blocked and provide an unreliable signal [1]. On the other hand, IMU signals' updated speed is faster than GNSS signals, but it provides low related accuracy signal and causes the error accumulation problem [2] [3].

To solve the mentioned problems, the combination of GNSS and IMU is a well-known strategy [4]. The method provides high accuracy with high update frequency. Nevertheless, it still does not solve the poor quality problem and blocking problem from GNSS.

Various feature-based localization methods are adopted to solve the problems. Some use low-level points [5] [6], corners, or lines [7], which are commonly appeared in the real world. However, it highly relies on environmental characteristics and is easily affected by noise. Middle-level features such as poles [8], curbs [9], or planes [10] are widely used in urban scenarios. Nevertheless, those methods often struggled in uniformly distributed scenes. The unique index of each feature should be established to classify each feature. And high-level features, such as object feature [11] [12] and semantic feature [13], are often detected by object detector. They are less affected by noise but require heavy computation loading.

In this paper, we present a region-determined localization method that combines RTK-GPS, IMU, and landmark maximum likelihood detection using pole-like features. The method can implement in different scenarios, including top-cover areas, familiar landmarks areas, and lack environmental features areas within the outdoor region. To decide the region, the Hector mapping [14] method is selected for 2D occupancy mapping. Moreover, the circle-fitting method and virtual boundary definition are also adopted for desired region determination. After determining the working region, we select the EKF for robot state estimation. Predict and update the state using the likelihood method results and feedback on the position error for better tracking results.

The contributions of this paper include:

- An approach using pole-like features for localization correcting within outdoor GNSS-denied scenario.
- A virtual boundary bounding method for working region determination is introduced, in which PVB is for the desired area selected, and NVB is for the obstacles area to avoid.
- A fusion of RTK-GPS, IMU, and depth camera, to implement in environment with various density, distribution, and numbers of pole-like features.

II. System Description

The flowchart of the proposed system is shown in Fig. 1. After preprocessing the depth data, we adopt the Hector mapping approach for environment reconstruction to build the occupancy map. Then, to build the feature map, the circle-fitting method is used for feature fitting due to the circle shape of the cylinder's cross-section. The actual feature distribution can be decided based on the fitted circle radius and each feature's position provided by the occupancy map.

In the parts of bounding, taking the feature map as the reference, the virtual boundary can be defined. The positive

*The authors are with National Taiwan University, Taipei, Taiwan. (e-mail: fengli@ntu.edu.tw).

virtual boundary (PVB) and the negative virtual boundary (NVB) are related to the desired area and the feature location we estimated. Those boundaries decide the limitation of cruising and avoiding collisions. In outdoor maintenance-task regions, pole-like obstacles are common and numerous. To make sure the pole-like features' index and robustness, the obstacles should be marked before the NVB built up. We will select the proper obstacles for previous landmark selection and remove the disturbance of dynamic objects. On the other hand, large-scale obstacles such as the pool, rocks, or flower garden can be considered as the excluded area, those areas can be ruled out while the desired area (PVB) determining.

To navigate the robot's location, the landmark perception can deal with the displacement between the GNSS signal and the real world. The obstacles-likelihood detection is adopted for comparing the landmarks and feedback on the position error. Updating the position information after the detection of each obstacle, the robot can track the provided path with better performance.

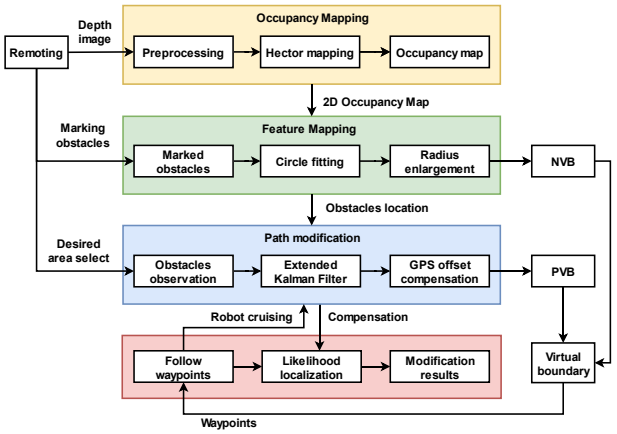


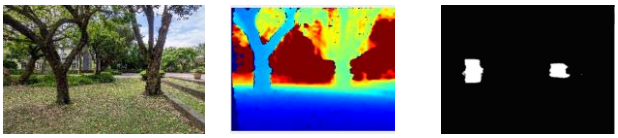
Figure 1. Flowchart of proposed system

III. MAPPING AND VIRTUAL BOUNDARY

A. Image mask

As the preprocessing of input data, an image mask is designed for depth image inputting. Due to the height of robot (0.5m), the mask includes the data within the height limitation at the same height of scanner and its 0.5m higher and lower region. The design can make sure the robot collects depth data from ground to 2-times height of robot. The region will be defined as a specific horizontal-slice layer. Taking those layers as scanning input, the process can ignore most of the scanning noise and reduce the calculation speed of the matching method. Furthermore, the mask only selects the core area of obstacles, e.g., trunk of trees. It can exclude the branches part and consider the trees as pole-like features.

Fig. 2 shows the input image and a horizontal slice through the resulting occupancy grid. Fig. 2(a) represents the scanning scenario scanned by RGB-D camera. Fig. 2(b) demonstrates the depth images with a depth-based color bar (cool colors are the closer region and vice versa). Fig. 2(c) is the image mask that based on the previous specific layer definition. The white region represents the included data within the layer, and the black part is the excluded data we remove. It shows the actual data we select is the most pole-like region of each obstacle.



(a) Scenario (b) Depth image (c) Image mask

Figure 2. The image mask for depth image.

B. 2D occupancy mapping

According to Kohlbrecher's Hector mapping method [14], the occupancy mapping process can be solved as a minimization problem with the scan endpoint s_i and the rigid transformation ξ. Scan endpoint s_i is the 2D local scanning data with size 640 x 1 pixels. ξ is the rigid transformation between robot locations and the original point (0, 0).

$$s_i = [s_{i,x}, s_{i,y}]^T \tag{1}$$

$$\xi = [p_x, p_y, \psi]^T \tag{2}$$

$[s_{i,x}, s_{i,y}]$ are the local coordinates on x-y plane of each endpoint, where i correspond to each incoming data index. $[p_x, p_y]$ represents the 2D position value, ψ represents the rotation value in degree. Combined the local coordinates s_i with the rigid transformation ξ, world coordinates $S_i(s_i, \xi)$ can be updated. Within occupancy map, the function M is defined as the occupied function which returns the map value in [0,1]. "1" is for the occupied data and "0" is the abandoned points. The cost function of occupancy map with n points is

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{n} [1 - M(S_i(s_i, \xi))]^2 \tag{3}$$

To solve the minimization problem, Gauss-Newton approach provides an intuitive solution. Rewrite the cost function as $F(x)$, where $F(x)$ and x represent the parameter to optimize:

$$F(x) = \frac{1}{2}|f(x)|^2 \tag{4}$$

$$\begin{aligned} \Delta x^* = & \underset{\Delta x}{\operatorname{argmin}}(F(x + \Delta x)) \\ = & \underset{\Delta x}{\operatorname{argmin}} \frac{1}{2}|f(x + \Delta x)|^2 \\ \approx & \underset{\Delta x}{\operatorname{argmin}} \frac{1}{2}|f(x) + J(x)\Delta x)|^2 \\ = & \underset{\Delta x}{\operatorname{argmin}} \frac{1}{2}(|f(x)|^2 + 2\Delta x^T J(x)^T f(x) \\ & + \Delta x^T J(x)^T J(x)\Delta x) \end{aligned} \tag{5}$$

where $J(x)$ is the Jacobian function. To minimize the error of Δx

$$J(x)^T f(x) + J(x)^T J(x)^T \Delta x = 0 \tag{6}$$

$$\Delta x = -(J(x)^T J(x))^{-1} J(x)^T f(x) \tag{7}$$

The first order Taylor expansion of endpoint registration is

$$1 - M(S_i(\xi + \Delta\xi)) \approx 1 - M(S_i(\xi)) - \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \Delta\xi \tag{8}$$

Compared with Eq. (5), function $J(x)$ is equal to $J(\xi)$ and corresponded with

$$J(\xi) = \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \tag{9}$$

According to the method [14], $\Delta\xi$ and $\nabla M(S_i(\xi))$ can be derived, the iteration of status ξ can be expressed in

$$\begin{aligned} \xi_{n+1} &= \xi_n + \Delta\xi \\ &= \xi_n - (J(\xi)^T J(\xi))^{-1} J(\xi)^T f(\xi) \end{aligned} \tag{10}$$

C. Feature fitting

After mapping the occupancy map, the locations of obstacles are sometimes unclear due to the scanning noise or registration displacement. Moreover, when the poles are too close, those poles should be defined as the same set of obstacles. Therefore, it is necessary to fit the useful data together to build up the feature map. The 2D projection of pole-like feature object is a horizontal circle slice. We consider the circle equation

$$(x_i - x_t)^2 + (y_i - y_t)^2 = r_t^2 \tag{11}$$

where x_i, y_i are the coefficients from the given data points. x_t, y_t, r_t are the part that needs to be solved. Rearrange the equation and rewrite to matrix form:

$$x_i(-2x_t) + y_i(-2y_t) + (x_t^2 + y_t^2 - r_t^2) = -(x_i^2 + y_i^2) \tag{12}$$

$$\begin{bmatrix} -2x_i & -2y_i & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ x_t^2 + y_t^2 - r_t^2 \end{bmatrix} = -x_i^2 - y_i^2 \tag{13}$$

$$\underbrace{\begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_t \\ y_t \\ x_t^2 + y_t^2 - r_t^2 \end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix} -x_1^2 - y_1^2 \\ -x_2^2 - y_2^2 \\ \vdots \end{bmatrix}}_{B} \tag{14}$$

The optimization problem can be solved by the least square method. The linear system of equations $A_{n\times 3} \cdot \mathbf{X}_{3\times 1} = B_{n\times 1}$. Solving the equation, the optimal results $\mathbf{X}^*$ are

$$\mathbf{X}^* = \begin{bmatrix} \mathbf{X}_1^* \\ \mathbf{X}_2^* \\ \mathbf{X}_3^* \end{bmatrix} = (A^T A)^{-1} A^T B \tag{15}$$

The desired trunk position and radius:

$$\begin{cases} x_t^* = \mathbf{X}_1^* \\ y_t^* = \mathbf{X}_2^* \\ r_t^* = \sqrt{(x_t^*)^2 + (y_t^*)^2 - \mathbf{X}_3^*} \end{cases} \tag{16}$$

D. Virtual boundary

According to our definition, virtual boundary can be clarified into PVB and NVB. PVB represents the desired area for applications. Which enclosures the boundary of robot cruising region through remoting or driving. NVB is the boundary that is decided by the feature obstacles. To avoid the collision caused by the robot size or the uneven-obstacles part such as tree roots, the obstacles radius enlargement should be considered. The enlarged radius will be defined as the summation of the obstacles' radius and the threshold value we set. Fig. 3 shows the forming process of NVB. Fig. 3(a) is the occupancy map mapped by Sec. 3. B. The white part represents the free area, black is the occupied area, and gray part shows the unknown region. Fig. 3(b) represents the feature map using the circle fitting approach mentioned in Sec. 3. C. The red circles are the developed fitted circle. Fig. 3(c) represents the NVB of map. The combination of the feature map and the enlargement form the NVB region. Red circles are the radius of the estimated obstacle, and blue-dotted lines represent the NVB.

(a) Occupancy map (b) Feature map (c) NVB

Figure 3. The definition of negative virtual boundary (NVB)

IV. Landmark-based GNSS shifting correction

A. Maximum Likelihood

When the feature object is observed, since there is no identification ID on the object, it is necessary to determine which object is seen. The position and the radius information are used for likelihood judgment. Defining the robot real pose $X = [x,\ y,\ \theta]^T$ and observes object's location $X_i = [x_i,\ y_i,\ \theta_i]^T$, where $(x,\ y)$ represent the position, θ is the rotate angle, and i is the index of incoming data. The observed obstacles information that includes position and radius value can be calculated by the circle fitting approach. Observed obstacle that using position landmark can be defined as z_{lm}:

$$z_{lm} = \begin{bmatrix} x \\ y \\ r \end{bmatrix} \tag{17}$$

where r is the radius of the feature. The ideal observed obstacle landmark information is defined as $\hat{z}_{lm}$ which can be derived by the rigid transformation between X and X_i:

$$\hat{z}_{lm}(X, X_i) = \begin{bmatrix} \sin(\theta) & -\cos(\theta) & 0 \\ \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i - x \\ y_i - y \\ r_i \end{bmatrix} \tag{18}$$

Inputting the (X, X_i) and corresponded z_{lm}, the likelihood function can be defined as $lh(z_{lm}, X, X_i))$:

$$lh(z_{lm}, X, X_{ti}) = e^{(-(\hat{z}_{lm} - z_{lm})^T W (\hat{z}_{lm} - z_{lm}))} \tag{19}$$

where W is the weighted 3x3 matrix that is designed for the sensitivity of detecting performance. The value of $lh(\cdot)$ will be larger when the positions are close, and vice versa. For each observed obstacles' z_{lm}, the likelihood function $lh(\cdot)$ can be calculated. The one t_i^* with the largest value can be found from map M, which means that it is the most likely obstacle.

$$t_i^* = \underset{x_i \in M}{\operatorname{argmax}}\ lh(z_{lm}, X,\ X_i) \tag{20}$$

B. State Estimation

The state consists with non-linear parts, therefore, EKF is adopted to solve the non-linear state estimation. The state of robot contains the position (x, y), the angle θ, and the estimator of the GNSS offset (x_{offset}, y_{offset}). The state transfer function can be written as a 5x1 matrix, where $\Delta X = [x, y, \theta, x_{offset}, y_{offset}]^T$:

$$X(n) = X(n-1) + \begin{bmatrix} -\frac{v}{\omega}\sin(\theta) + \frac{v}{\omega}\cos(\theta + \omega\Delta t) \\ \frac{v}{\omega}\cos(\theta) - \frac{v}{\omega}\cos(\theta + \omega\Delta t) \\ \omega\Delta t \\ 0 \\ 0 \end{bmatrix} \quad (21)$$

$u = [v, \omega]^T$ is the input signal. The input signal is the velocity command calculated by the controller, including linear velocity v and angular velocity ω.

V. Experimental Works and Results

A. Experimental setup

Figure. 4. The platform of robot.

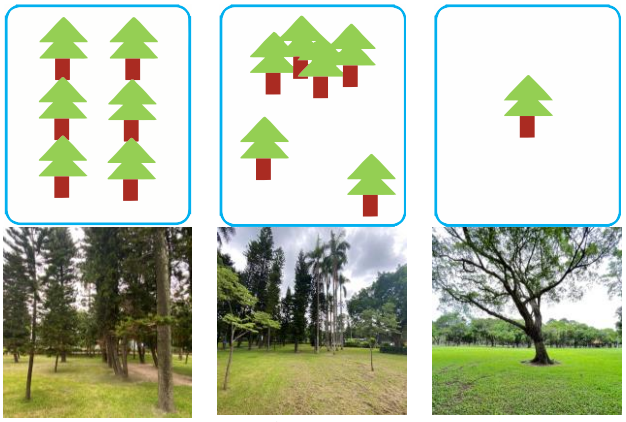
(1). Hardware description

To receive depth information and GNSS signal, an UGV which carries RGB-D camera and an RTK-GPS tracker is selected. The camera we select is Intel® RealSense™ D435, which provides 87° x 58° field of view (FOV) and 640 x 480 pixels resolution in the 6m scanning range. It can provide high data accuracy (within 3cm @ 4m) for occupancy mapping. Moreover, for a better environment mapping performance, the robot also provides an IMU, and wheel encoder odometry as the internal sensors. Jetson Nano, which provides high computation resources for depth processing compatible with RealSense™, is selected as the embedded Linux computer. The computer is connected to the Raspberry Pi module through the Wi-Fi dongle and provides a 15fps framerate in USB 2.0 for real-time implementations. Fig. 4 shows the platform of the robot.

(2). Scenario overview

There are three selected scenarios for the demonstration. The experiments took place in the Youth park, New Taipei city, Taiwan. Those cases represent the specific area combined with flat ground and straight obstacles such as lawns and trees. The scenarios describe different obstacles distributions under poor GNSS signal quality. For the GNSS quality, we clarify the it into three accuracy states, RTK-fixed, RTK-float, and differential GPS (DGPS), which are each corresponded to the accuracy within 2cm, 20cm-1m, and up to 1m [15]. For desired path planning, a boustrophedon decomposition method introduced by [16] is selected. The robot will follow the mentioned application process for mapping and tracking the desired path.

Three scenarios are demonstrated for different poor GNSS quality cruising situations. (1) Case 1: Enough tree with uniform tree distribution. (2) Case 2: Enough tree with non-uniform tree distribution. (3) Case 3: Non-enough tree. Fig. 5 shows the scenarios and schematic drawings of three scenarios. Fig. 5(a) represents the scenario description of case 1. In this case, the environmental characteristic is enough and uniform. Moreover, the tree distribution is sparse enough for the robot to go through. In the second case part, Fig. 5(b) also shows enough environmental feature. However, it describes a non-uniform tree distribution with different density tree areas. Fig. 5(c) represents the situation of case 3, which shows the lack of environmental characteristics.



(a) Case 1 (b) Case 2 (c) Case 3

Figure. 5. Three scenarios with different tree distribution.

B. Experimental results

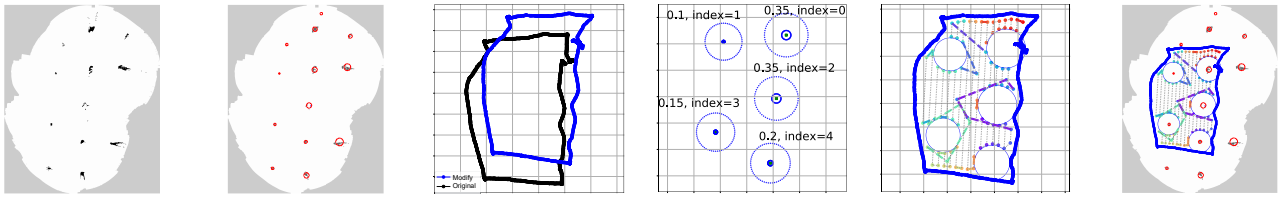
Each case's mapping, bounding, and path planning results can be shown in Figs. 6-8. Occupancy maps and feature maps of the environment can be described in Figs. 6(a)-(b), Figs. 7(a)-(b) and Figs. 8(a)-(b). The PVBs and NVBs are described in Figs. 6(c)-(d), Figs. 7(c)-(d) and Figs. 8(c)-(d).

In PVB part, we decide the desired area under the poor GNSS quality environment, therefore, the maximum likelihood detection (Sec. 4. A) is implemented to correct the shifting error. The black lines represent the original moving path, and the blue lines are the modified path. NVB is defined as the feature map and the obstacles' enlargement. In those cases, the specific enlarge value is selected as 1m of each obstacle radius. The blue dots are the obstacles with different indices, and blue circles represent the enlargement of different indices of obstacles' radius.

The path planning results and desired areas are described in Figs. 6(e)-(f), Figs. 7(e)-(f) and Figs. 8(e)-(f). The boundaries are based on the defined PVB and NVB, and color points represent the designed waypoints. From red to purple, the robot will track the waypoints in sort and cruising through the gray lines that are connected by waypoints. Eventually, those results lead to the desired cruising area which combines the mapping, bounding, and path planning results.

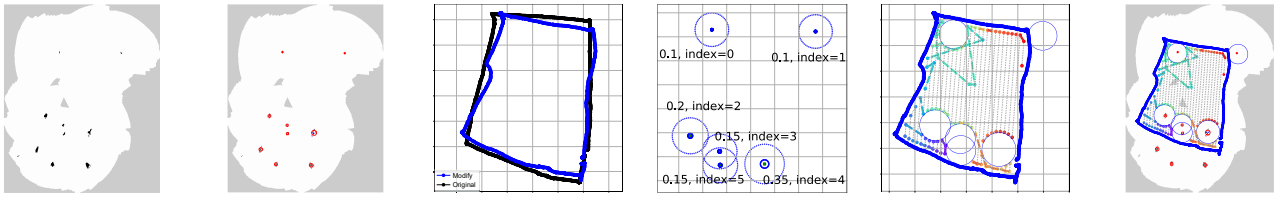
(1). Enough tree with uniform tree distribution

In case 1, the scenario describes enough environmental features and uniform tree distribution (Fig. 6), which



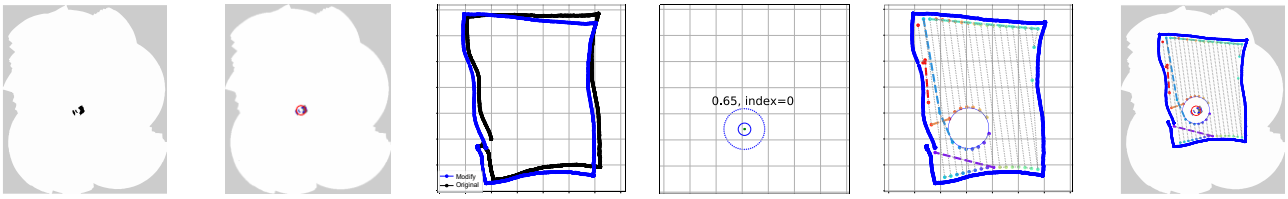
(a) Occupacy map (b) Feature map (c) PVB (d) NVB (e) Path planning (f) Desired area

Figure. 6. Mapping, bounding, and path planning results for case 1.



(a) Occupacy map (b) Feature map (c) PVB (d) NVB (e) Path planning (f) Desired area

Figure. 7. Mapping, bounding, and path planning results for case 2.



(a) Occupacy map (b) Feature map (c) PVB (d) NVB (e) Path planning (f) Desired area

Figure. 8. Mapping, bounding, and path planning results for case 3.

demonstrated the method's ability within severely poor GNSS quality scenarios. In Figs. 9(a)-(b), the boustrophedon decomposition results are demonstrated. Fig. 9(a), which represents the original path without correction, also shows the GNSS quality problem. The black line is the original moving path, and the red line represents the ideal expected waypoints. It can show that the displacement of GNSS shifting is very high during robot cruising.

However, the maximum likelihood detection can update the landmark obstacle for GNSS correction. Though the GNSS signal is worse, enough environmental characteristics provide the chance for the poses modification. In Fig. 9(b), the blue line is the modified moving path, and the red line represents the ideal expected waypoints. It shows the high overlapping of two lines, representing the successful tracking results.

During cruising, the GNSS signal quality is low due to the tree shade issue. Tree shades cover the top of the robot and block the signal receiving. In Fig. 6(c), the difference between the two paths is much higher compared to the other two cases (Fig. 7(c) and Fig. 8(c)). Figs. 9(g)-(h) shows the GNSS quality and the tree observation distribution. In Fig. 9(g), blue, red, and black dots represent the GNSS quality of RTK-fixed, RTK-float, and DGPS. The quality level remained in low-quality status, which are all in RTK-float status, and DGPS status. In Fig. 9(h), the tree observation distribution is described. The red line is the modified path described in Fig. 9(b), and the blue dots represent the tree-observation results during movement. Though the GNSS quality is poor and drifting, the modification result of robot location is in high overlapping due to enough tree observation.

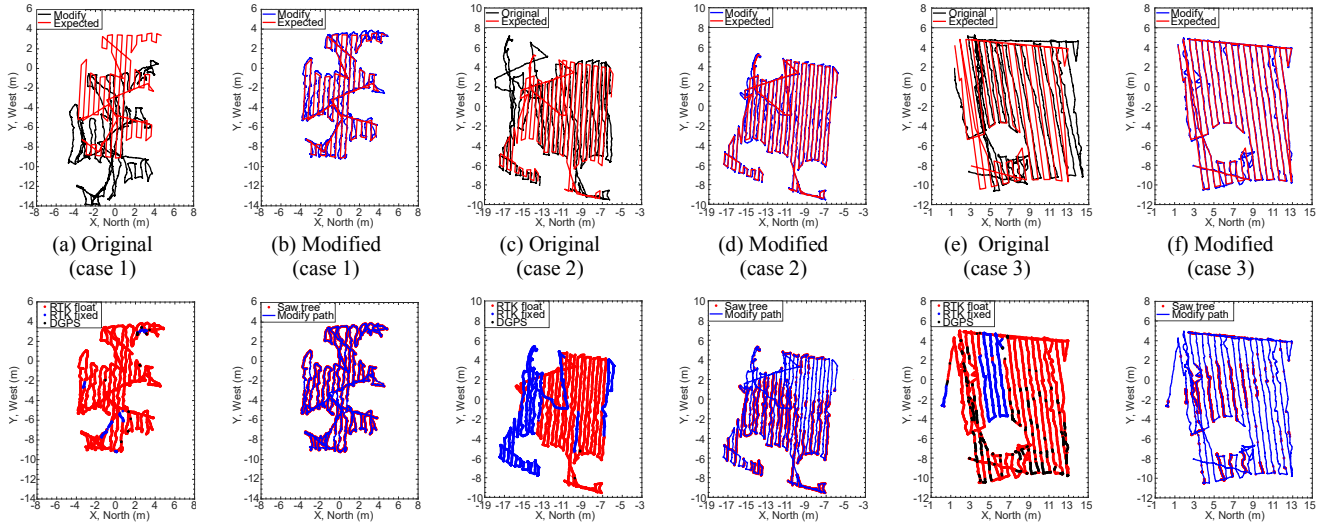
(2).Enough tree with non-uniform tree distribution

This scenario describes enough environmental features and non-uniform tree distribution (Fig. 7), which led to showing the method's ability within narrow and wide feature scenarios. Compared with case 1, the high-density area's trees, in this case, are much denser and won't able to let the robot go through. Therefore, the robot will only cruise through the uncovered flat ground. In Fig. 9(c), the original path in the black line shift away from the expected path (red), and Fig. 9(d) shows the correction of shifting (blue). Compared with the original path, the modified one corrects the displacement and shows great tracking performance. The familiar targets do not cause the error-modification of tracking. The maximum likelihood detection method is reliable in whether observation density case of area.

Figs. 9(i)-(j) shows the GNSS quality and the tree observation distribution of case 2. In Fig. 9(i), the quality status remains in RTK-float status and RTK-fixed status. In Fig. 9(j), there are only few regions without tree observing. As long as we observe the trees for modification, the position drifting would improve compared with non-modification cases.

(3). Non-enough tree

In case 3, the scenario shows non-enough environmental features (Fig. 8), and also implemented in poor GNSS areas due to leaves coverage. Compared with the other two cases, this case provides the lowest environmental features. There is only a single tree and flat ground in the desired area. The lack of environmental features will limit the chance of modifying



(a) Original (case 1) (b) Modified (case 1) (c) Original (case 2) (d) Modified (case 2) (e) Original (case 3) (f) Modified (case 3)

(g) GNSS quality (case 1) (h) Tree observation (case 1) (i) GNSS quality (case 2) (j) Tree observation (case 2) (k) GNSS quality (case 3) (l) Tree observation (case 3)

Figure. 9. Original path, modified path, GNSS quality, and tree observation distribution results for 3 cases.

the tracking path during cruising. Corresponding to the cruising results, in Fig. 9(e), the original path shift most in the right-side part. Fig. 9(f) shows the results of the modification. Using the method, the accumulated error is corrected.

In Fig. 8(e), the path planning result shows that the robot will cruise from left to right, which means the GNSS drifting problem will be highest on the right side of the figure under poor GNSS quality (Fig. 9(k)). However, in Fig. 9(l), we observe the tree and correct the localization position in the middle area. Therefore, in solving the error drifting problem, the modified path and the expected path highly overlap.

VI. Conclusion

This paper introduces a region-determined localization method including mapping, bounding, and GNSS-signal correcting techniques. The process is specific to the outdoor environment with pole-like obstacle features. It contains the Hector mapping approach and circle-fitting approach, Maximum likelihood method, and EKF filter for feedback tracking and error decreasing. Three demonstrations show the cruising results in different situations, representing different implementation purposes. The method can successfully modify the cruising path while tracking the desired path and following the desired path without collisions.

The process projects the pole-like features on the 2D plane. In the future, we would consider the 3D shapes of the object, the obstacle's boundaries would not be limited by circles but by arbitrary shapes. Nevertheless, exceptional cases, such as trees with the same size, direction, and distance at the same time, are not considered in this process.

ACKNOWLEDGEMENT

This research is supported by URSROBOT Systems and National Science and Technology Council, Taiwan, under Grants: MOST 111-2221-E-002-191 and NSTC 112-2221-E-002-192.

References

[1] M. Bilker, and H. Kaartinen. "The quality of real-time Kinematic (RTK) GPS positioning. " Rep. Finn. Geod. Inst. 2001, 1, 1-25.

[2] X. Liu, Q. Zhou, X. Chen, L. Fan and C. -T. Cheng, "Bias-Error Accumulation Analysis for Inertial Navigation Methods," in IEEE Signal Processing Letters, 2022, vol. 29, pp. 299-303

[3] H. Zhou, Y. Zhao, X. Xiong, and Y. Lou, "IMU Dead-Reckoning Localization with RNN-IEKF Algorithm," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2022, pp. 11382-11387

[4] A. D. Kulkarni, G. G. Narkhede, and S. N. Motade, "SENSOR FUSION: An Advance Inertial Navigation System using GPS and IMU," Int. Conf. on Computing, Communication, Control and Automation (ICCUBEA), 2022, pp. 1-5

[5] A. Ranganathan, S. Matsumoto, and D. Ilstrup, "Towards illumination invariance for visual localization," IEEE Int. Conf. on Robotics and Automation (ICRA), 2013, pp. 3791-3798

[6] C. Linegar, W. Churchill and P. Newman, "Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera," IEEE Int. Conf. on Robotics and Automation (ICRA), 2016, pp. 787-794

[7] J. H. Lee, S. Lee, G. Zhang, J. Lim, "Outdoor place recognition in urban environments using straight lines," IEEE Int. Conf. on Robotics and Automation (ICRA), 2014, pp. 5550-5557

[8] R. Spangenberg, D. Goehring and R. Rojas, "Pole-based localization for autonomous vehicles in urban scenarios, " IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2016, pp. 2161-2166

[9] H. Lee, J. Park and W. Chung, "Curb feature based localization of a mobile robot in urban road environments," IEEE Int. Conf. on Robotics and Automation (ICRA), 2015, pp. 2794-2799

[10] J. Kümmerle, M. Sons, F. Poggenhans, et al., "Accurate and Efficient Self-Localization on Roads using Basic Geometric Primitives," IEEE Int. Conf. on Robotics and Automation (ICRA), 2019, pp. 5965-5971

[11] J. Li, D. Meger and G. Dudek, "Semantic Mapping for View-Invariant Relocalization," IEEE Int. Conf. on Robotics and Automation (ICRA), 2019, pp. 7108-7115

[12] H. Wang, C. Xue, et al. "Visual Semantic Localization based on HD Map for Autonomous Vehicles in Urban Scenarios," IEEE Int. Conf. on Robotics and Automation (ICRA), 2021, pp. 11255-11261

[13] S. L. Bowman, N. Atanasov, K. Daniilidis and G. J. Pappas, "Probabilistic data association for semantic SLAM," IEEE Int. Conf. on Robotics and Automation (ICRA), 2017, pp. 1722-1729

[14] Kohlbrecher, Stefan, et al. "A flexible and scalable SLAM system with full 3D motion estimation." IEEE Int. symp. on safety, security, and rescue robotics, 2011.

[15] Y. Morales and T. Tsubouchi, "DGPS, RTK-GPS and StarFire DGPS Performance Under Tree Shading Environments," IEEE Int. Conf. on Integration Technology, 2007, pp. 519-524

[16] Huang, Kuo-Chun, et al. "A novel solution with rapid voronoi-based coverage path planning in irregular environment for robotic mowing systems." Int. Journ. of Intell. Robotics and Appl. , 2021, 5(4):558-5