

A Robust 2D-SLAM Technology With Environmental Variation Adaptability

Li-Hsin Chen and Chao-Chung Peng

Abstract—Simultaneous localization and mapping (SLAM) in complicated indoor/outdoor unknown environments is challenging. With a demand on high mobility and high integrity intelligent robotics, it is desired that the SLAM system should be portable and possibly standalone. To carry out the pose estimation as well as the mapping without relying on the information from other sensors, such as image, inertial measurement unit, rotary encoder of ground vehicle and so on, a single 2D light detection and ranging (LiDAR) is considered in this paper. In order to fulfill a robust 2D SLAM technology in unknown environments, the principal component analysis (PCA) is utilized to evaluate LiDAR scan contours and to carry out a corridor detector. The corridor detector is further extended to achieve adaptive unstable points removal, mapping probability adjustment as well as loop closure. Based on an adaptive grid map segmentation scheme, the cumulative mapping errors can obviously be reduced and a precise 2D map can be eventually carried out. Many experiments are conducted to verify the proposed method. Finally, for comparison, this paper utilizes the scan data and ground truth provided by the Computer Science and Artificial Intelligence Laboratory (CSAIL) at the Massachusetts Institute of Technology (MIT), to verify the localization precision of the proposed algorithm. Experiment shows that from the scan data in the route up to about 350 m, the maximum error can be as low as about 20 cm.

Index Terms—Simultaneous localization and mapping (SLAM); light detection and ranging (LiDAR); iterative closest point (ICP); occupancy grid map; loop closure.

I. INTRODUCTION

WE are entering an era where vehicles and robots need to be equipped with cognitive capability so that they will be able to explore the external environment and operate autonomously. Simultaneous localization and mapping (SLAM) is the core technology responsible for making this the implementation possible. SLAM uses sensors to carry out a spatial scan, obtain the geometric outline of the environment and build a map for the robot to explore the environment. Furthermore, the sensor localization can be carried out based on the built map, and the map can be updated according to the localization results and the scan data from the sensor. Accordingly, the spatial cognition capability

Manuscript received May 31, 2019; revised July 11, 2019; accepted July 11, 2019. Date of publication July 26, 2019; date of current version November 13, 2019. This work was supported by the Ministry of Science and Technology under Grant MOST 107-2221-E-006-114-MY3 and Grant MOST 108-2923-E-006-005-MY3. The associate editor coordinating the review of this article and approving it for publication was Dr. Ashish Pandharipande. (Corresponding author: Chao-Chung Peng.)

The authors are with the Department of Aeronautics and Astronautics, National Cheng Kung University, Tainan 701, Taiwan (e-mail: ccpeng@mail.ncku.edu.tw).

Digital Object Identifier 10.1109/JSEN.2019.2931368

is significantly important in environmental cognition. Hence, the ranging sensor is most suitable for the implementation. Particularly, LiDAR has a higher accuracy in measuring distances, which is much more beneficial for precision map building and self-localization.

In recent years, 2D SLAM relevant technologies have been widely applied in autonomous ground vehicles in the pursuit of smart factories. Based on the applications, many issues are raised and discussed accordingly. As mentioned by the outstanding SLAM review paper [1], for many applications and environments, numerous major challenges and important questions remain unsolved. One of the most important issues is the adaptability of the 2D SLAM for the environment variations. As a result, it is worthy to pursue a robust perception and navigation system.

Since 1987, ranging sensors such as Sound Navigation And Ranging (Sonar) and LiDAR have been developed as vehicle sensors, and the occupancy grid mapping approach has been utilized to build obstacle mapping for navigation [2], [3]. In 1991, the extended Kalman filter (EKF) was implemented in the sonar's measurement model and vehicle's model for the first time to locate the vehicle in a known environment [4]. After that, the EKF was utilized to estimate the location of landmark and the pose of the vehicle, and at the same time the vehicle localization and the landmark map building are carried out, which can be referred to as EKF-SLAM [5]–[11].

During the development of an occupancy grid map, the binary Bayes filter was first used in order to update the occupancy grid map [2], [3]; then, the map updating approach is simplified by using the binary Bayes filter with the odds ratio of occupied and unoccupied grids [12]. Based on the above research, the mechanism of updating the grid map with the scanning characteristics of the sensor can be referred to as “inverse sensor model”, and the approach of updating the grid map by using the “forward model” is also proposed [13]. The occupancy grid map is widely used on the robot systems [14]–[20].

However, most of the SLAM algorithms rely on the information from the rotary encoder of ground vehicle, hence the localization and map building cannot be carried out independently. In addition, significant cumulative errors occur due to the long-term operation of SLAM. A large-scale (to suburban area) SLAM, as well as the SLAM and Moving Objects Tracking (SLAMMOT) were carried out in 2007 [21]. The implementation actively reduces the impact of moving objects on a static map. Also, in the entire architecture, the regional submaps are built, and the pose of the vehicle in the global

map can be obtained by the loop closure from submaps to the global map. In [22], a large-scale outdoor SLAM is carried out with the data association of outdoor scan, and the approach of calculating histogram correlation is used to determine the loop closure.

As a well-known SLAM algorithm, GMapping [23] is based on Rao-Blackwellized particle filters relied on wheel odometry. Different from GMapping, Hector SLAM [24] and Cartographer [25] use an odometry denied scan matcher, which solves the optimal pose to maximize the occupancy probability of the laser scan on grid map. Moreover, Cartographer deals with the loop closure by applying Sparse Pose Adjustment [26] to solve the pose graph optimization. Afterward, comparisons and evaluations among these three algorithms are made by [27], [28].

Recently, various methods and approaches in SLAM have been presented. The study [29] analyzes the contour, or shape, of LiDAR scan to approach a LiDAR odometry. The paper [30] proposes a forward compensation of pose estimation by applying fast Fourier transform to LiDAR scan image. In [31], correntropy, similar to Gaussian weight, is introduced to compute rigid registration of scan matching with high precision and robustness. In [32], the pose graph optimization is speeded up by sampling the usable pose nodes according to the designed score. Additionally, several researches [33]–[35] intend to extract line feature from LiDAR scan for line-based mapping and localization-aiding. In [36], line features of LiDAR scan is used to form its orientation histogram for absolute orientation sensing (relative to the building North) and loop closure. Another feature extraction algorithm [37] uses corner as well as line to recognize proper correspondences so as to enhance localization speed. However, line feature extraction might be unsuitable for some complex environments.

Robots and unmanned vehicles can be implemented in a wider range of applications, to fulfill the highly commercialized demand for them. In view of this, the robustness and usability of the SLAM algorithm will become significantly important when facing a more varied environment. Therefore, the research goal of this study is to develop a two-dimensional LiDAR algorithm for localization and mapping, so that the localization and operation can be carried out independently without relying on other sensors. Moreover, the loop closure will be utilized for the error correction, to create a comprehensive and low-cost localization and mapping system.

II. POSE ESTIMATION

On the basis of LiDAR, the pose estimation is to match and align the scan points with obstacle points in the map, and then go back to calculate the pose of LiDAR in the map, to achieve the purpose of pose estimation.

Let ${}^B\mathbf{p}_i^{(k)}$ denote the returned k^{th} scan point from the i^{th} scan in the 2-D LiDAR sensor coordinate system, and let ${}^E\mathbf{q}_i^{(k)}$ denote the obstacle point in the mapping coordinate system matching to ${}^B\mathbf{p}_i^{(k)}$, then the geometric relationship between them is as follows:

$${}^E\mathbf{q}_i^{(k)} = \mathbf{R}^{(k)} {}^B\mathbf{p}_i^{(k)} + \mathbf{t}^{(k)} \quad (1)$$

where $\mathbf{R}^{(k)}$ represents the sensor's rotation matrix of yawing angle of the k^{th} scan in the map, and $\mathbf{t}^{(k)}$ represents the sensor's translation vector in the map. The LiDAR pose estimation resolves the following question:

$$\begin{aligned} (\hat{\mathbf{R}}^{(k)}, \hat{\mathbf{t}}^{(k)}) &= \arg \min_{\mathbf{R}, \mathbf{t}} \sum_i^N w_i \|(\mathbf{R} {}^B\mathbf{p}_i^{(k)} + \mathbf{t}) - {}^E\mathbf{q}_i^{(k)}\|^2 \\ \text{subject to } \mathbf{R}^T \mathbf{R} &= \mathbf{I} \end{aligned} \quad (2)$$

where $\hat{\mathbf{R}}^{(k)}$ and $\hat{\mathbf{t}}^{(k)}$ is the pose estimation for the k^{th} scan, and w_i is the weight factor of each scan point in cost function. Selection of the weight factor will be addressed in the following section. To enhance the computation efficiency for 2D SLAM case, an analytic solution is derived without the use of a singular value decomposition (SVD). Please refer to Appendix A for its analytical solution. However, ${}^E\mathbf{q}_i^{(k)}$ is also an unknown parameter in implementation. To deal with that, ICP, which is an algorithm using the iterative approach to search ${}^E\mathbf{q}_i^{(k)}$, is introduced.

Iterative Closest Point (ICP) [38] is a point-alignment algorithm with simpler structure, and it is able to calculate the rotation and translation between two sets of unmatched points. Its algorithm disassembles (2) into the following recursion:

$$\begin{aligned} ({}^\ell \Delta \mathbf{R}, {}^\ell \Delta \mathbf{t}) &= \arg \min_{\mathbf{R}, \mathbf{t}} \cdot \sum_i^N w_i \|(\mathbf{R} {}^\ell \mathbf{p}_i + \mathbf{t}) - {}^\ell \mathbf{q}_i\|^2 \\ \text{subject to } \mathbf{R}^T \mathbf{R} &= \mathbf{I} \end{aligned} \quad (3)$$

$${}^\ell \mathbf{q}_i = \arg \min_{\mathbf{q}_j \in Q} \| {}^\ell \mathbf{p}_i - \mathbf{q}_j \| \quad (4)$$

$$\begin{cases} {}^{\ell+1} \mathbf{R} = {}^\ell \Delta \mathbf{R} {}^\ell \mathbf{R} \\ {}^{\ell+1} \mathbf{t} = {}^\ell \Delta \mathbf{R} {}^\ell \mathbf{t} + {}^\ell \Delta \mathbf{t} \\ {}^{\ell+1} \mathbf{p}_i = {}^\ell \Delta \mathbf{R} {}^\ell \mathbf{p}_i + {}^\ell \Delta \mathbf{t} \end{cases} \quad (5)$$

where the left superscript of the variables represent the loop index of the ICP. In the ℓ^{th} iteration, the closest point to ${}^\ell \mathbf{p}_i$ is regarded as the corresponding matching point ${}^\ell \mathbf{q}_i$ in (4), and it will be taken to (3) for rigid alignment to calculate the pose increment, ${}^\ell \Delta \mathbf{R}$ and ${}^\ell \Delta \mathbf{t}$. After that, ${}^\ell \mathbf{R}$, ${}^\ell \mathbf{t}$ and the scan points ${}^\ell \mathbf{p}_i$ can be updated through (5) to search for the new matching point ${}^{\ell+1} \mathbf{q}_i$. Follow the repeating procedures to update the points continuously, the correct matching point can then be found. When iteration converges, ${}^\ell \mathbf{R}$ and ${}^\ell \mathbf{t}$ will be the desired pose estimation, $\hat{\mathbf{R}}^{(k)}$ and $\hat{\mathbf{t}}^{(k)}$.

However, if ICP is utilized to carry out the LiDAR pose estimation by aligning the scan points with obstacle points in the map, some details and process should be taken into consideration, including initial pose compensation, outlier removal and weight design in rigid alignment. Therefore, the overall LiDAR estimation process based on ICP is as shown in Fig. 1, and the details will be described in the subsequent section.

A. Initial Pose Compensation

The procedure of “Closest Point Matching” in ICP will likely search the wrong matching points, especially when the initial poses between two sets of points have a significant difference. It will make the solution converge to the local optimum, and the points set will converge to the wrong pose, as shown in Fig. 2. Therefore, initial pose compensation for

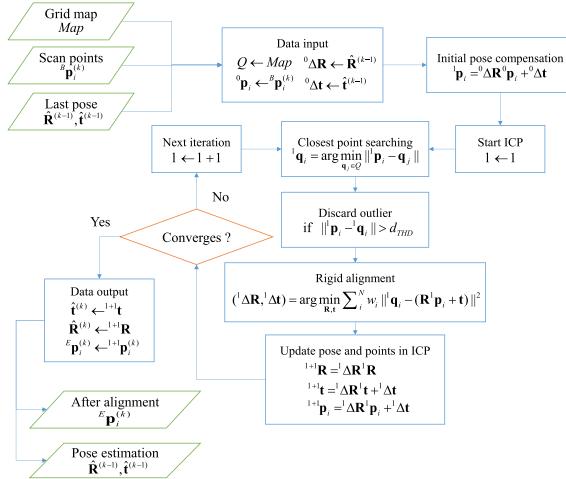


Fig. 1. The flowchart of LiDAR pose estimation by using the ICP.

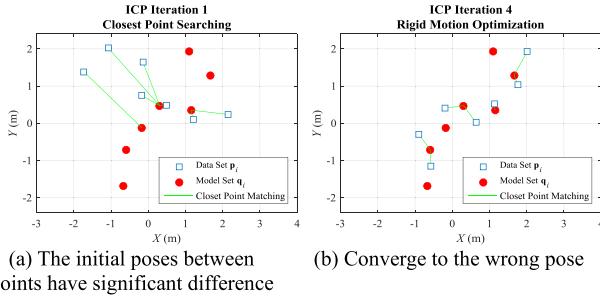


Fig. 2. The points will converge to the wrong pose if the initial poses between points have a significant difference.

the points in ICP (i.e., ${}^0\Delta\mathbf{R}$ and ${}^0\Delta\mathbf{t}$) is very important. In order to reduce the pose difference between initial points ${}^0\mathbf{p}_i$ (i.e., the scan data ${}^B\mathbf{p}_i^{(k)}$) and target matching points ${}^E\mathbf{q}_i^{(k)}$ in ICP when solving $\hat{\mathbf{R}}^{(k)}$ and $\hat{\mathbf{t}}^{(k)}$, the sensor pose at the k^{th} scan, the pose estimation $\hat{\mathbf{R}}^{(k-1)}$ and $\hat{\mathbf{t}}^{(k-1)}$ from the last scan can be used as the initial pose compensation, ${}^0\Delta\mathbf{R}$ and ${}^0\Delta\mathbf{t}$, for the current ICP.

B. Outlier Removal

The closest point matching in ICP also includes the wrong matching points for the calculation of rigid alignment, which results in pose miscalculation for the points, as shown in Fig. 3. Therefore, when processing the closest point matching in ICP, the matched outlier should be discarded. The removal can be determined by the distance between the matching points. The matched point that is too far apart will be regarded as the matched outlier. It will be discarded and will not be included in the calculation of rigid alignment, which is:

$$\begin{aligned} &\text{Discard the matching pair } (\ell \mathbf{p}_i, \ell \mathbf{q}_i) \\ &\text{if } \|\ell \mathbf{p}_i - \ell \mathbf{q}_i\| > d_{THD} \end{aligned} \quad (6)$$

where d_{THD} is the distance threshold of matching points, and its value can be determined by the scan frequency and the moving speed of the sensor. Under scan frequency of 40 Hz and sensor moving speed of 3 m/sec without sharp rotation, the sensor displacement will be about 75 mm, and the distance

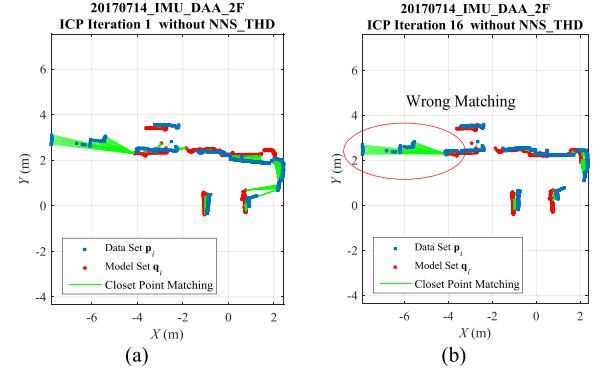


Fig. 3. The closest point matching of outlier will cause pose estimation error in ICP: (a) Matching of initial points. (b) Iterative convergence is influenced by the closest point matching of outlier.

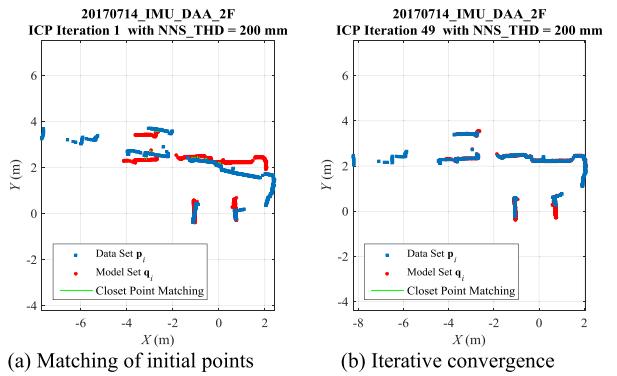


Fig. 4. The ICP results after the removal of the matched outlier.

between the corresponding matching points in LiDAR scan will not exceed this limit. Hence, d_{THD} can be set to 100 mm to 200 mm, and the matched outlier can be discarded, as shown in Fig. 4.

C. Weight Design in Rigid Alignment

In the weight design of ICP rigid alignment in (3), the distribution density of scan points ${}^B\mathbf{p}_i^{(k)}$ is inversely proportional to the scan distance $r_i^{(k)}$ due to the radial scan of LiDAR. For example, the surface of an object far from LiDAR will be sketched by more scan points than the one closer to LiDAR, as shown in Fig. 5. In order to balance this situation, the weight w_i in (3) is designed to be:

$$w_i = \begin{cases} r_i, & r_i > \bar{r} \\ 0, & \text{otherwise} \end{cases},$$

where

$$\bar{r} = \frac{1}{N} \sum_i^N r_i \quad (7)$$

The distribution of the distance weight (7) is determined by the scan distance r_i , to compensate for the distribution difference of scan point density at different scan distances, and to balance the inverse relationship between the scan density and the scan distance with the radial scan. In addition, (7) sets the weight of scan points that are close to the average scan

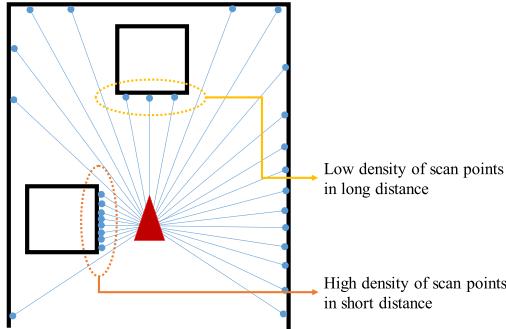


Fig. 5. Distribution density of scan points of LiDAR is influenced by the scan distance.

distance \bar{r} to zero, to reduce the impact from the scan points in short distance on the pose estimation. Hence, the scan points in long distance can align to their matching points in the map, and the accuracy of the sensor's pose estimation will be enhanced. Similarly, the inverse weight can be designed as the following:

$$I_{w_i} = \begin{cases} 1/r_{\min} - 1/r_i, & 1/r_i < 1/\bar{r} \\ 0, & \text{otherwise} \end{cases}$$

where

$$\frac{1}{\bar{r}} = \frac{1}{N} \sum_i^N \frac{1}{r_i} \quad (8)$$

In comparison with (7), the weight distribution of inverse weight III is more concentrated on the scan points far from LiDAR, so that the sensor pose estimation is able to overcome sharper rotation and movement.

III. MAP CONSTRUCTION

In SLAM, map construction is an important part. The built map can be used as a system output, as well as a feedback for the localization of pose estimation algorithm, to enhance accuracy and robustness of the localization.

A. Occupancy Grid Map

The coordinates of scan points obtained by LiDAR can be confirmed after the localization. The coordinates indicate the position of obstacles, which represents the obstacle map as shown in Fig. 6 (a). The occupancy grid map can divide the continuous spaces into finite grids, to compute the countless scan points and determines whether the grid is occupied by an obstacle, as shown in Fig. 6 (b).

B. Binary Bayes Filter

The binary Bayes filter is utilized to update the occupancy grid map based on the derivation of probability theory [12], [39]. The set of observation of LiDAR from the 1st scan to the k^{th} scan), denoted by $z^{(1:k)}$, was taken into consideration (the superscript colon of z “:” represents “continuous intersection”) to calculate the probability $p(m_\ell|z^{(1:k)})$ of the ℓ^{th} grid point m_ℓ that is occupied by the obstacle in the map. The probability is indicated by the color of the grids. $p(m_\ell|z^{(1:k)}) = 0$ for white grids, and $p(m_\ell|z^{(1:k)}) = 1$ for black ones, as shown in Fig. 7 (a).

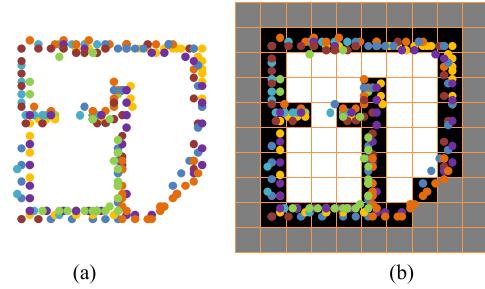


Fig. 6. LiDAR scan points and mapping: (a) The scan points of obstacle will overlap and accumulate over the scan time. (b) The occupancy grid map can divide spaces into grids and compute the countless scan points with finite grids, to complete the map construction.

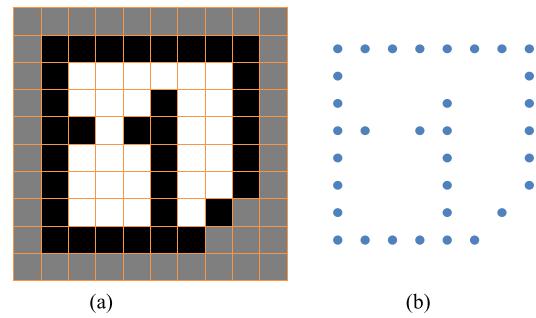


Fig. 7. The occupancy grid map and the extracted scan points on obstacle map: (a) The color of grid map indicates the probability of grid in occupancy. (b) The grids with high probability will be extracted to be the obstacle points in the map.

In sensor pose estimation, the black grids will be extracted from the grid map to be the obstacle points in the map, and they will be aligned with the scan points of the sensor, as shown in Fig. 7 (b). In this approach of extracting obstacle points, the density of points will be fully depended on the resolution of grid in the map. In addition, the obstacle points need to be matched and aligned with the scan points in the pose estimation. According to the alignment strategy in (7), the focus of the alignment of points will be at the scan points in long distance. Thus, the resolution of the grid map must not be less than those scan points so that the grid map is able to provide sufficient and detailed geometric information of the obstacle. For example, the longest scan distance of LiDAR will be limited to about 5 m in an ordinary indoor space. As for the LiDAR with 0.25 degrees of angular resolution, the spacing between the scan points is about 20 mm in a scan distance of 5 m, so it is more appropriate to set the grid size to 20 mm or smaller. If the longest scan distance of LiDAR at a broader space is kept to about 10 m, then the interval of scan points will be wider; hence, the grid resolution of the map can be relaxed to about 50 mm based on the same calculation.

The updated formula for the probability of the binary Bayes filter would be:

$$\frac{p(m_\ell|z^{(1:k)})}{1 - p(m_\ell|z^{(1:k)})} = \frac{p(m_\ell|z^{(k)})}{1 - p(m_\ell|z^{(k)})} \frac{p(m_\ell|z^{(1:k-1)})}{1 - p(m_\ell|z^{(1:k-1)})} \frac{1 - p(m_\ell)}{p(m_\ell)} \quad (9)$$

where $p(m_\ell)$ is the priori probability of the grid m_ℓ of being occupied by obstacles. However, there is no information regarding to the occupancy status of grid before mapping, hence $p(m_\ell)$ will be set to a fixed value of 0.5. In order to enhance the numerical accuracy and calculation speed in (9), its logarithm is obtained by:

$$l_\ell^{(1:k)} = l_\ell^{(k)} + l_\ell^{(1:k-1)} - l_\ell \quad (10)$$

where l represents the logarithm of the odds ratio $p/(1-p)$:

$$\begin{aligned} l_\ell^{(1:k)} &\equiv \log \frac{p(m_\ell|z^{(1:k)})}{1 - p(m_\ell|z^{(1:k)})}, \\ l_\ell^{(k)} &\equiv \log \frac{p(m_\ell|z^{(k)})}{1 - p(m_\ell|z^{(k)})}, \quad l_\ell \equiv \log \frac{p(m_\ell)}{1 - p(m_\ell)} \end{aligned} \quad (11)$$

In (9) and (10), the inverse sensor model $p(m_\ell|z^{(k)})$ is the key to correlating $p(m_\ell|z^{(1:k-1)})$ with $p(m_\ell|z^{(1:k)})$, which indicates the confidence level of the grid m_ℓ being occupied under the current sensor observation, $z^{(k)}$. The occupancy grid map can be updated according to (9) with the decision of the inverse sensor model.

C. Inverse Sensor Model

The inverse sensor model describes the confidence level of the grid being occupied under the sensor measurement. In the case of LiDAR, it scans the environment with several laser beams in the space. The returning point of the laser beam is where the obstacle is located, and the space penetrated by the laser beam before returning is the area that is not occupied by the obstacle. For those grids on the grid map that are penetrated by the laser beam, the confidence level of the grids to be occupied by obstacle is a function between “the scan distance of LiDAR” and “the distance between the grid and the position of LiDAR”. On the other hand, the grid that is not penetrated by the laser cannot be determined whether it is occupied by the obstacle, so its confidence level is 0.5. The following formula is based on the above discussion:

$$p(m_\ell|z^{(k)}) = \begin{cases} f(d_\ell^{(k)}, r_i^{(k)}), & \text{if } m_\ell \text{ is on the } i\text{-th laser beam.} \\ 1/2, & \text{otherwise} \end{cases} \quad (12)$$

where $p(m_\ell|z^{(k)})$ represents the confidence level of the grid m_ℓ being occupied under the observation of $z^{(k)}$ at the k^{th} scan. If m_ℓ is located on the i^{th} laser beam, then $d_\ell^{(k)}$ is the distance between m_ℓ and the position of LiDAR, and $r_i^{(k)}$ is the scan distance of the i^{th} laser beam, as shown in Fig. 8. The confidence level of the grid m_ℓ in occupancy is the function between $d_\ell^{(k)}$ and $r_i^{(k)}$, and this function should describe the confidence level of the grid with the distribution of grids $d_\ell^{(k)}$ at the LiDAR scan distance $r_i^{(k)}$.

In this study, two LiDAR inverse sensor models are designed, one is the Gaussian distribution model (13), and

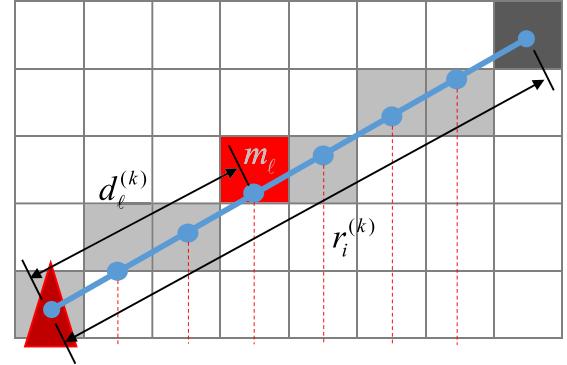


Fig. 8. The geometric relationship between the observed grids in the map and the LiDAR laser beam.

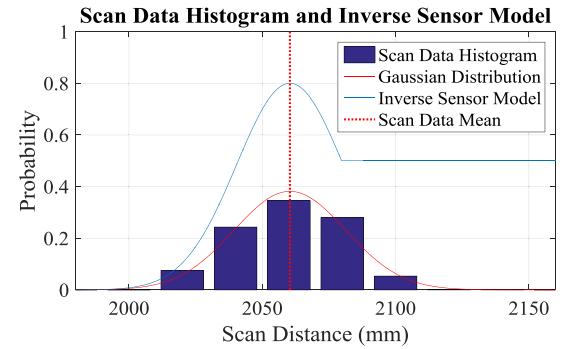


Fig. 9. The Gaussian distribution model with $C = 1.0$ and the distribution of laser beam on grids with scan distance of 20 mm in static LiDAR.

the other is Quadratic distribution model (14):

$$p_G(d, r) = \begin{cases} \frac{1+C}{2} \exp\left(-\frac{(d-r)^2}{2\sigma^2}\right), & d \leq r + \sigma\sqrt{2\ln(1+C)} \\ \frac{1}{2}, & d > r + \sigma\sqrt{2\ln(1+C)} \end{cases} \quad (13)$$

$$p_Q(d, r) = \begin{cases} \frac{1}{2} \frac{1}{(r-\sigma)^2} d^2, & d \in (0, r-\sigma] \\ \frac{1}{2} \left[1 + C \left(1 - \frac{1}{\sigma^2} (d-r)^2 \right) \right], & d \in [r-\sigma, r+\sigma] \\ \frac{1}{2}, & d > r+\sigma \end{cases} \quad (14)$$

where σ is the standard deviation of Gaussian distribution, and $C \in (0, 1)$ is the scaling factor used for adjusting the peak value of inverse sensor model, to determine the cognitive speed of the occupied grid. The design of the Gaussian distribution model (13) is based on the noise property of Gaussian distribution in the LiDAR laser ranging, as shown in Fig. 9. The grids that are located outside of the scan distance r cannot be penetrated by the laser beam and cannot be determined whether they are occupied or not; hence, their confidence level is 0.5.

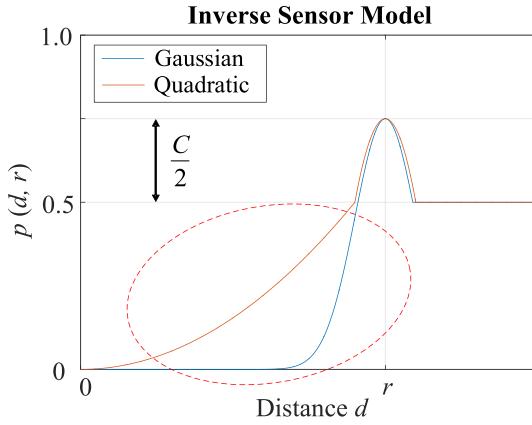


Fig. 10. Comparison of the quadratic distribution model and Gaussian distribution model.

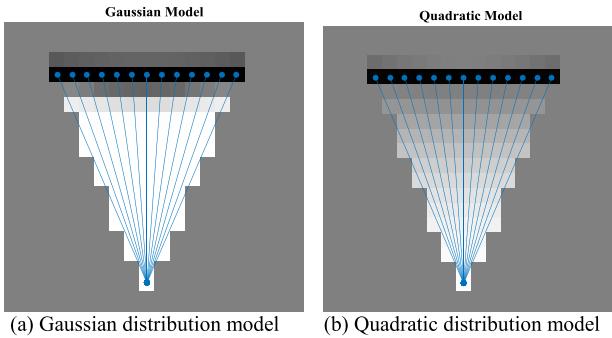


Fig. 11. Comparison of Gaussian distribution model and quadratic distribution model on the grid map.

The Quadratic distribution model (14) refers to the concept of [2] and (13), being that the main difference from (13) is that the calculation of the Quadratic distribution model is simpler than the Gaussian distribution model. According to the characteristics of Quadratic distribution model, its cognition to the unoccupied grids in the area within the scan distance r is relatively conservative to the Gaussian distribution model. As shown by the red circle part in Fig. 10, since the probability of quadratic model (orange line) is higher than the probability of Gaussian model (blue line), the update rate for the unoccupied grid by using the Quadratic model is slower than the one obtained by the Gaussian model. The associated simulation comparison can be seen in Fig. 11. It shows that the unoccupied grid will be recognized slower in the Quadratic model, but its reliability is relatively higher than the Gaussian distribution model once the grid is confirmed as an unoccupied space. However, due to its slower update rate, it may lead to an unstable SLAM when a robot suddenly goes into a corridor. To enhance its adaptability against to the change of environment, in the next section we are going to present an environment detection algorithm such that the C value is going to be updated accordingly.

IV. PREPROCESSING OF SCAN POINTS

In the algorithm of localization and mapping, the pose of the sensor is estimated from the matching and alignment in

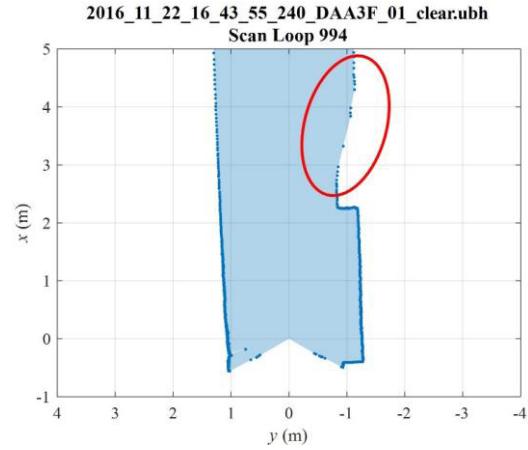


Fig. 12. Unstable scan points. (the scan area is located on the 3rd floor of the Department of Aeronautics and Astronautics, DAA).

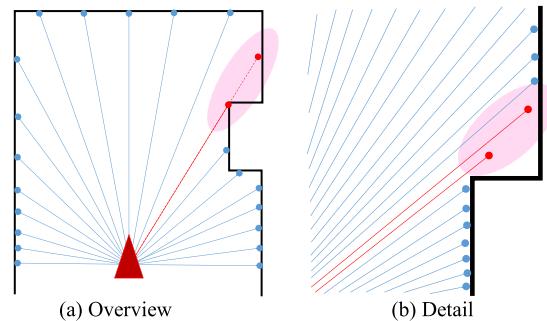


Fig. 13. The area where unstable scan points occur.

ICP between the points on the map and the LiDAR scan points; hence, the scan quality of LiDAR will affect the result of the points alignment. However, the influence on the pose estimation of the sensor will be reduced if these unstable scan points can be discarded before the ICP process.

A. Unstable Points Removal

The scan quality of LiDAR on the scan points depends on the scan of the laser beam's reflection. In the radial scan of laser beam, the scan boundary that is blocked by obstacles has a gap, and the scan quality of laser beam in this area is less stable: the laser beam may hit the boundary of the obstacle in front or in contrast, the background in the back. The received data from the scan distance varies between the two scan boundaries. In other words, the scan points in this region will move along the radial direction, as demonstrated by the pink oval area in Fig. 12 and Fig. 13.

The connection direction of scan points in the area where these unstable scan points occur is similar to the emission direction of the laser. Therefore, the angle between the two directions needs to be calculated in order to search for the scan points in this area. On the basis of Fig. 14, there are two connection directions for the i^{th} scan point ${}^B\mathbf{p}_i$: the first being that it can be connected with the previous scan point ${}^B\mathbf{p}_{i-1}$ or with the next scan point ${}^B\mathbf{p}_{i+1}$; and the second one

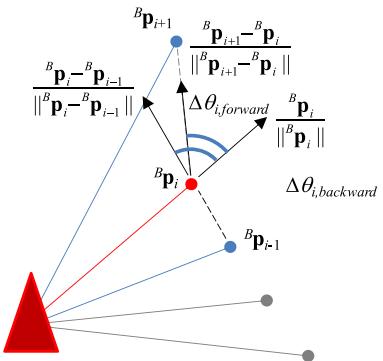


Fig. 14. Geometric diagram of the scan angles.

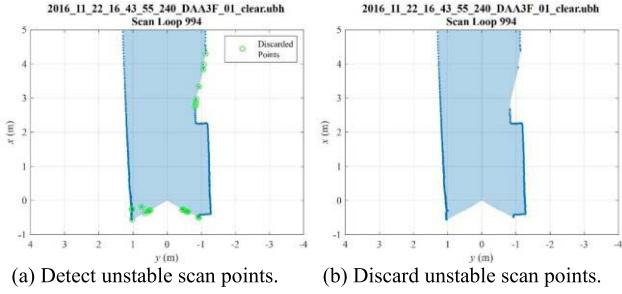


Fig. 15. Use the mechanism of detecting small scan angle to discard unstable scan points.

is that the emission direction of laser beam for ${}^B\mathbf{p}_i$, is parallel to its position vector in the sensor coordinate.

On the basis of the scan angles in the geometric diagram of Fig. 14, the forward and backward scan angles can be calculated from the coordinates of scan points ${}^B\mathbf{p}_i$, ${}^B\mathbf{p}_{i+1}$ and ${}^B\mathbf{p}_{i-1}$:

$$\begin{cases} \Delta\theta_{i,forward} = \cos^{-1} \left| \frac{{}^B\mathbf{p}_i^T}{\|{}^B\mathbf{p}_i\|} \frac{{}^B\mathbf{p}_{i+1}-{}^B\mathbf{p}_i}{\|{}^B\mathbf{p}_{i+1}-{}^B\mathbf{p}_i\|} \right| \\ \Delta\theta_{i,backward} = \cos^{-1} \left| \frac{{}^B\mathbf{p}_i^T}{\|{}^B\mathbf{p}_i\|} \frac{{}^B\mathbf{p}_i-{}^B\mathbf{p}_{i-1}}{\|{}^B\mathbf{p}_i-{}^B\mathbf{p}_{i-1}\|} \right| \end{cases} \quad (15)$$

where the absolute value of arccosine is for obtaining the acute angle of scan angle. If the forward and backward scan angles $\Delta\theta_{i,forward}$ and $\Delta\theta_{i,backward}$ are too small for the i^{th} scan point ${}^B\mathbf{p}_i$, it will be regarded as an unstable scan point and will be discarded:

Discard ${}^B\mathbf{p}_i$ if

$$\Delta\theta_{i,forward} < \Delta\theta_{Thd} \text{ or } \Delta\theta_{i,backward} < \Delta\theta_{Thd} \quad (16)$$

where $\Delta\theta_{Thd}$ is the threshold of the scan angle. More scan points will be discarded as the value of $\Delta\theta_{Thd}$ increases. With the application of (15) and IV-B, these unstable scan points can be found and discarded. This approach has excellent judging results especially when in an environment formed of structures of beams and columns, as shown in Fig. 15.

B. Corridor Detector

As pointed out by [1], environment recognition is also one of the challenging points for the robust SLAM. The features

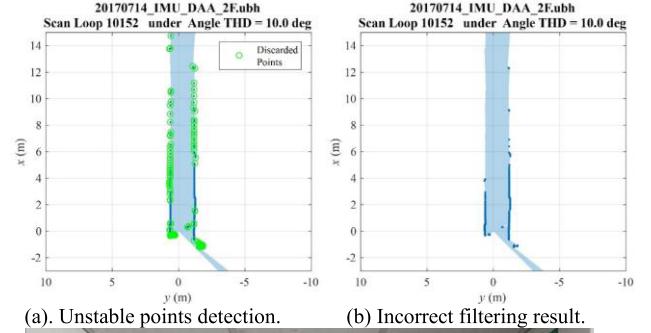


Fig. 16. The mechanism of unstable point detection causes excessive removal of scan points in the distance of corridor.

of the point cloud distribution in a room or in a corridor are different. Usually, a sophisticated model or a feature identifier is needed to figure out if a robot is moving in a corridor or elsewhere. Moreover, there is no strong link between the parameters adaption and the corridor detection. Therefore, in the following section, a corridor detector is proposed and the link for unstable point removal, mapping probability on-line adjustment as well as loop closure are made.

Based on the aforementioned section, it has been shown that the mechanism of detecting and discarding scan points with small scan angle in (15) and IV-B may cause incorrect outlier filtering results in the corridor environment, as shown in Fig. 16. The cause is that the geometric characteristics of far scan points in the corridor are similar to the ones of unstable scan points: the angle between the wall direction in a corridor and the emission direction of laser will reduce with the increasing of the scan distance.

As shown in Fig 17, if LiDAR is located in the middle of a corridor with a width of $2d$, then the distance between LiDAR and the wall in corridor will be d . For the scan point that is in the front of corridor with a distance of $d \times \cot(\Delta\theta_{Thd})$, its scan angle is equal to $\Delta\theta_{Thd}$, as shown in Fig 17(b). For the scan point that is farther than the distance of $d \times \cot(\Delta\theta_{Thd})$, its scan angle will be less than $\Delta\theta_{Thd}$. For example, if LiDAR is located in the middle of a corridor with a width of 2 meters, the scan point on the wall that is in the front of corridor with a distance of 20 meters will be discarded under 3 degrees of $\Delta\theta_{Thd}$. If $\Delta\theta_{Thd}$ is set to 10 degrees, the scan point on the

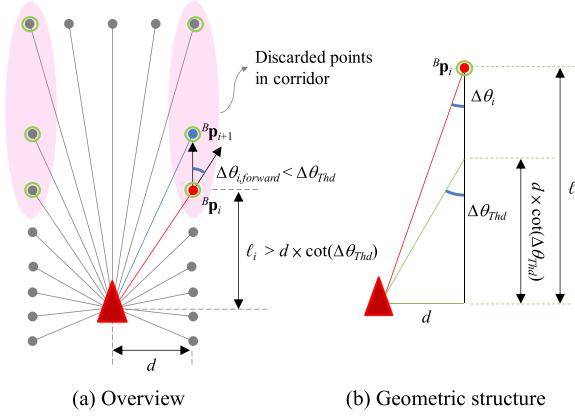


Fig. 17. The geometric relationship between the mechanism of discarded scan points and corridor.

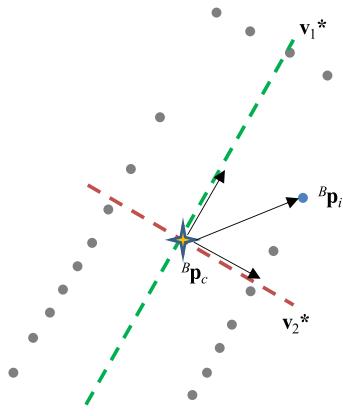


Fig. 18. A diagram for the corridor detection.

wall only in 6 meters will be reserved, as shown in Fig. 16. Excessive removal of scan points in the corridor will give rise to incorrect pose estimation.

In order to avoid the situation occurred in Fig. 16, it is necessary to determine whether LiDAR is located in the corridor based on the distribution shape of the current scan point, and then adjust the threshold $\Delta\theta_{Thd}$ in IV-B.

If the point distribution is in an elongated shape, the variation of the maximum of varied direction v_1^* of the points will be much greater than the variation of the minimum varied direction v_2^* , as shown in Fig. 18. The projection amount on the axis that go through the center of the points \mathbf{p}_c along the direction v_1^* , is much more than the projection amount on the v_2^* axis. The ratio between the two indicates the elongated degree of the distribution shape of points. On the basis of above discussion, we can define the indicator parameter of corridor $\eta_{Corridor}$ in (17):

$$\begin{aligned}\eta_{Corridor} &\equiv \frac{\max_{\|\mathbf{v}\|=1} \sum_i^N [\mathbf{v}^T ({}^B\mathbf{p}_i - \mathbf{p}_c)]^2}{\min_{\|\mathbf{v}\|=1} \sum_i^N [\mathbf{v}^T ({}^B\mathbf{p}_i - \mathbf{p}_c)]^2} \\ &= \frac{\sum_i^N [v_1^{*T} ({}^B\mathbf{p}_i - \mathbf{p}_c)]^2}{\sum_i^N [v_2^{*T} ({}^B\mathbf{p}_i - \mathbf{p}_c)]^2}\end{aligned}\quad (17)$$

where \mathbf{p}_c is the center of points:

$$\mathbf{p}_c = \bar{\mathbf{p}} \equiv \frac{1}{N} \sum_i^N {}^B\mathbf{p}_i \quad (18)$$

The solution of (17) is computed from the PCA. The directions \mathbf{v}_1 and \mathbf{v}_2 of the principal component can be obtained by calculating the covariance matrix Σ with diagonal decomposition:

$$\Sigma \equiv \sum_i^N ({}^B\mathbf{p}_i - \bar{\mathbf{p}}) ({}^B\mathbf{p}_i - \bar{\mathbf{p}})^T > 0 \quad (\Sigma = \Sigma^T) \quad (19)$$

which can be re-represented by

$$\Sigma = \mathbf{Q} \Lambda \mathbf{Q}^{-1} = \mathbf{Q} \Lambda \mathbf{Q}^T, \quad (\mathbf{Q}^{-1} = \mathbf{Q}^T \text{ for } \Sigma = \Sigma^T) \quad (20)$$

where

$$\begin{cases} \Lambda = \text{diag}(\lambda_1, \lambda_2), \quad \lambda_1 > \lambda_2 > 0 \\ \mathbf{Q} = [\mathbf{v}_1 \ \mathbf{v}_2], \quad \|\mathbf{v}_1\| = \|\mathbf{v}_2\| = 1, \quad \mathbf{v}_1^T \mathbf{v}_2 = 0 \end{cases} \quad (21)$$

The principal component direction of the points \mathbf{v}_1 and \mathbf{v}_2 are the solution for \mathbf{v}_1^* and \mathbf{v}_2^* :

$$\mathbf{v}_1^* = \arg \max_{\|\mathbf{v}\|=1} \sum_i^N [\mathbf{v}^T ({}^B\mathbf{p}_i - \bar{\mathbf{p}})]^2 = \mathbf{v}_1 \quad (22)$$

$$\mathbf{v}_2^* = \arg \min_{\|\mathbf{v}\|=1} \sum_i^N [\mathbf{v}^T ({}^B\mathbf{p}_i - \bar{\mathbf{p}})]^2 = \mathbf{v}_2 \quad (23)$$

The indicator parameter of corridor $\eta_{Corridor}$ can be obtained by substituting (22) and (23) into (17):

$$\eta_{Corridor} \equiv \frac{\max_{\|\mathbf{v}\|=1} \sum_i^N [\mathbf{v}^T ({}^B\mathbf{p}_i - \mathbf{p}_c)]^2}{\min_{\|\mathbf{v}\|=1} \sum_i^N [\mathbf{v}^T ({}^B\mathbf{p}_i - \mathbf{p}_c)]^2} = \frac{\lambda_1}{\lambda_2} \quad (24)$$

Eq. IV-C shows that the value can be used to evaluate if a robot is going into a corridor like environment and SLAM parameters can then be modified accordingly.

C. Adaptive Threshold Design

Threshold of scan angle $\Delta\theta_{Thd}$ should be automatically tuned according to the indicator $\eta_{Corridor}$ of distribution shape of scan points. When LiDAR is located in corridor, $\Delta\theta_{Thd}$ should be adjusted lower to avoid excessive removal of scan points in the corridor. As for the programming of $\eta_{Corridor}$ and $\Delta\theta_{Thd}$, the $\eta_{Corridor}$ is mapped to the angular indicator $\theta_{Corridor}$ so that its unit is consistent with $\Delta\theta_{Thd}$:

$$\theta_{Corridor} \equiv \tan^{-1} \frac{1}{\eta_{Corridor}} \quad (25)$$

Therefore, the range $[1, \infty)$ of the corridor indicator $\eta_{Corridor}$ is mapped to $45 \sim 0$ degree of $\theta_{Corridor}$. On the basis of attempted experience, the mapping relationship between $\theta_{Corridor}$ and $\Delta\theta_{Thd}$ is designed to be as follows:

$$\Delta\theta_{Thd} \equiv \begin{cases} 3^\circ & \text{if } \theta_{Corridor} \leq 25^\circ \\ \frac{7}{15} \theta_{Corridor} - \frac{26}{3}^\circ, & \text{if } \theta_{Corridor} \in (25^\circ, 40^\circ] \\ 10^\circ, & \text{if } \theta_{Corridor} > 40^\circ \end{cases} \quad (26)$$

From (26), we can find that $\theta_{Corridor}$ is close to 0 degrees when the scan points are distributed in an elongated shape

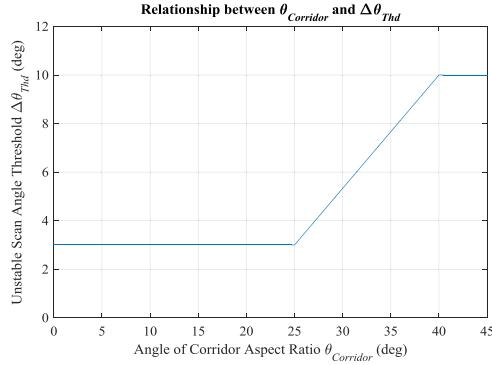


Fig. 19. The adaptive threshold of scan angle is designed on the basis of the angular indicator of corridor.

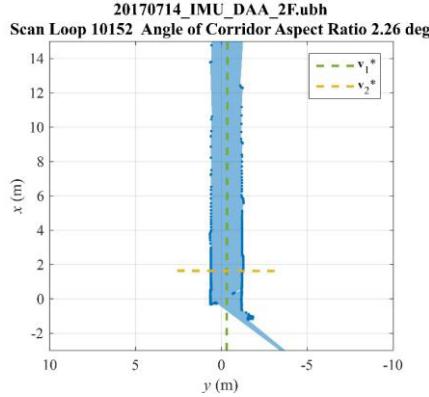


Fig. 20. Corridor detection on the scan point distribution using principal component analysis.

and the setting of $\Delta\theta_{Thd}$ is reduced to 3 degrees; $\theta_{Corridor}$ will go up to 45 degrees when the scan points are distributed in a square shape, and the setting of $\Delta\theta_{Thd}$ is increased to 10 degrees until it reaches saturation point. The mapping relationship is as shown in Fig. 19.

With (25), the distribution shape of the scan points can determine whether they are distributed in an elongated shape. Based on this result, threshold of scan angle $\Delta\theta_{Thd}$ can be adjusted to proper value. For example, $\theta_{Corridor}$ is 2.26 degrees in corridor, as shown in Fig. 20, and 3 degrees of $\Delta\theta_{Thd}$ can be obtained from (26). Thus, more scan points in corridor can be kept, as shown in Fig. 21.

In addition, further on we demonstrate how the corridor detector can be used to enhance the environment adaptability during the SLAM. Since there are only few features in the corridor environments, as illustrated in Fig. 16, it is easy to cause SLAM failures.

Consider the case in which the robot is moving from a room and then steps into a corridor. The dimensions of the corridor are 1.8 meters in width and 35 meters in depth. The features are monotonous and there aren't any beams or columns in such environment. Therefore, the corridor detector is applied to evaluate whether the robot is located in a corridor or not. Once the robot recognizes its location is indeed a corridor and the environment has not yet been explored, the ICP correspondence matching threshold d_{THD}

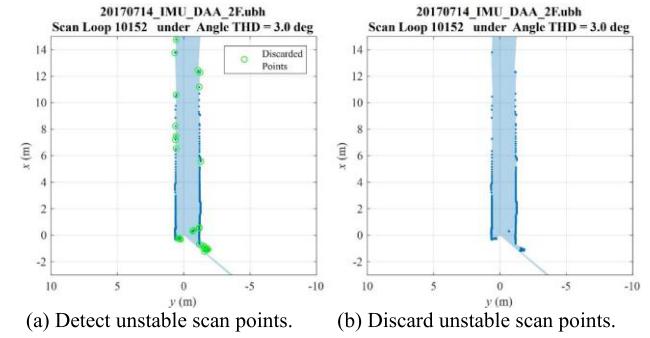


Fig. 21. Unstable point removal based on the adaptive threshold of scan angle with corridor detection.

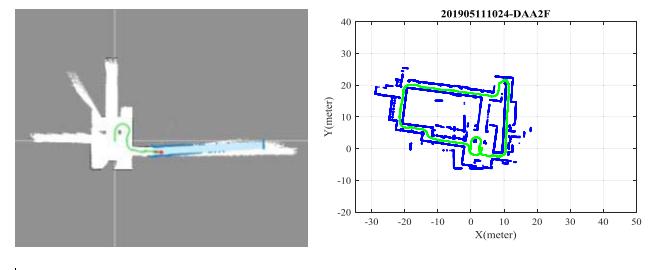


Fig. 22. SLAM result without the use of corridor detector based sensor model adaption law.

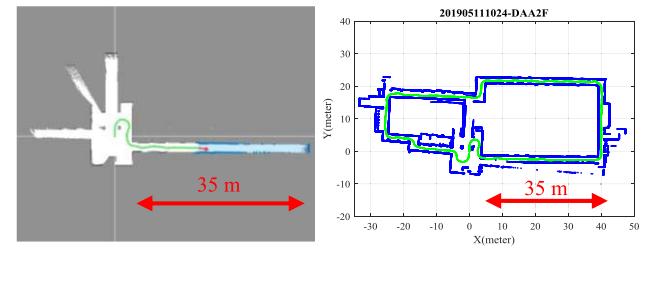


Fig. 23. SLAM result with the aid of corridor detector based sensor model adaption law.

and the mapping factor C used in the sensor model (14) are going to be set by 150 mm and 0.6, respectively. Fig. 22 is the experimental result by using fixed SLAM parameters. Since the corridor cannot be constructed in a short period of time, it gives rise to mapping divergence. On the contrary, as demonstrated in Fig. 23, applying the corridor detector makes it able to adapt to the variation of the environment and to accelerate the recognition speed of the two sides of the corridor as well as the bottom wall, then a successful SLAM can be achieved. Therefore, experiments show that the corridor detector provides readily usable information about the type of environment (i.e., room versus corridor) and then changes the mapping probability accordingly. Fig. 23(b) shows that the result of the map construction is not completely closed, which can be further improved by using loop closure.

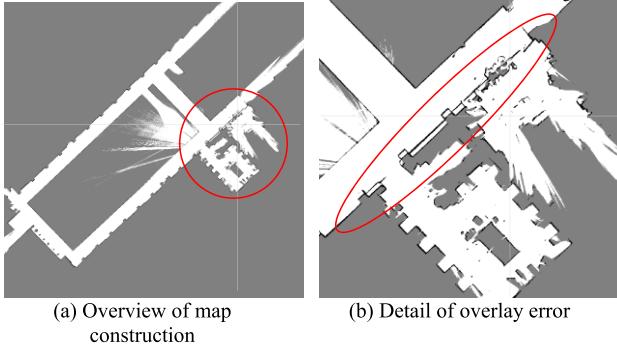


Fig. 24. The cumulative errors in the process of localization and mapping will result in overlay error. (the scan area is located on the 3rd floor of DAA).

V. LOOP CLOSURE

Both pose estimation error and mapping error occurred in the process of localization and mapping algorithm. These errors will affect the subsequent localization and mapping, and the cumulative errors will result in overlay error, which makes the same obstacle inconsistently mapped on the map. For example, the scan on the 3rd floor of DAA as shown in Fig. 24 (a): we started from the laboratory and went around the 3rd floor in a clockwise direction, and then back to the laboratory. In the process of localization and mapping algorithm, the mapped wall when leaving the laboratory and returning to the laboratory is inconsistently shown on the map, as shown in Fig. 24 (b).

This kind of deformed and non-closed map can be divided into several submaps for shifting and adjustment, so that the map can be more freely to correct the non-closed area in the map and estimated path of sensor in each submap. The correction mechanism for this is called “loop closure”, by which the cumulative errors in localization process can be eliminated.

A. Map Segmentation

In the process of loop closure, it is necessary to ensure the correctness of shapes of each submap and the connection between the maps. First, the estimated scan path obtained from SLAM algorithm is cut into several pieces with fixed length, and several grid maps can be reconstructed along those paths, as shown in Fig. 25 and Fig. 26. Each submap is generated adaptively according to the traveling distance as well as $\eta_{Corridor}$. To put it clearly, when the traveling distance is greater than 7 meters and $\eta_{Corridor} < 10$, one submap is going to be issued. Note that a large value of $\eta_{Corridor}$ indicates that the system is now in a corridor like environment. Such environment usually lacks of solid feature points and is not adequate for loop closure. Hence, the corridor detector is able to evaluate the adequacy of the submap generation. Finally, the occupied grids of each grid map are chosen as the feature points of submap for the loop closure.

B. Pose Correction of Submaps

In the loop closure, all the submaps must be rotated and shifted except the first submap. That means the pose of all

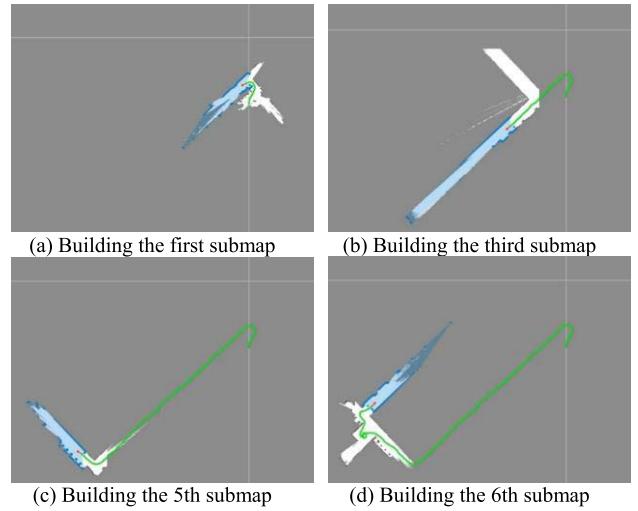


Fig. 25. Map segmentation for submaps. (localization and mapping result from the 3rd floor of DAA).

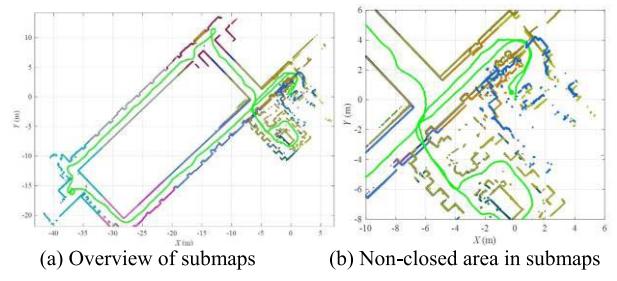


Fig. 26. Submaps without loop closure. (localization and mapping result from the scan data on the 3rd floor of DAA).

submaps will be readjusted. In plane motion, the rotation has one degree of freedom (yaw angle ψ) and the translation has two degrees of freedom (translation vector \mathbf{t}); hence, the solution for the pose of all submaps (with a total of n submaps) is as follows:

$$\mathbf{x} = [\psi^{(2)} \mathbf{t}^{(2)T} \psi^{(3)} \mathbf{t}^{(3)T} \dots \psi^{(n)} \mathbf{t}^{(n)T}]^T$$

where

$$\mathbf{t}^{(i)} = [t_x^{(i)} t_y^{(i)}]^T \quad (27)$$

where the superscript represents the index of submap. The goal of loop closure is to adjust all the submaps so that the corresponding feature points between maps can coincide. Therefore, the optimization issue is as follows:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \sum_{i=2}^n \sum_{j=1}^{N^{(i)}} \left\| \mathbf{T}(\mathbf{p}_j^{(i)}) - \mathbf{q}_j^{(i)} \right\|^2 \quad (28)$$

$$\mathbf{T}(\mathbf{p}_j^{(i)}) = \mathbf{R}^{(i)} \mathbf{p}_j^{(i)} + \mathbf{t}^{(i)},$$

$$\mathbf{R}^{(i)} = \begin{bmatrix} \cos \psi^{(i)} & -\sin \psi^{(i)} \\ \sin \psi^{(i)} & \cos \psi^{(i)} \end{bmatrix}, \quad \mathbf{p}_j^{(i)} = \begin{bmatrix} p_{x,j}^{(i)} \\ p_{y,j}^{(i)} \end{bmatrix} \quad (29)$$

where $\mathbf{p}_j^{(i)}$ represents the j^{th} of $N^{(i)}$ feature points in the i^{th} submap and \mathbf{T} represents the transformation of translation and rotation. $\mathbf{q}_j^{(i)}$ represents the matching feature points in other submaps for $\mathbf{p}_j^{(i)}$. The approach is to determine the closest

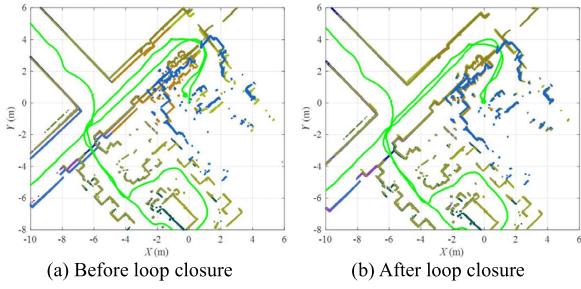


Fig. 27. Loop closure effect. (localization and mapping result from the scan data on the 3rd floor of DAA).

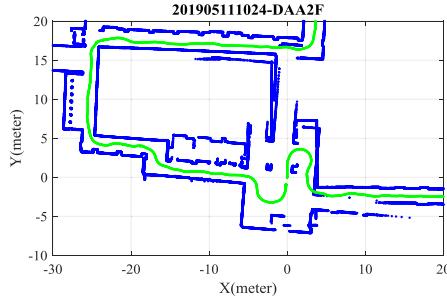


Fig. 28. Map restructure with the aid of corridor detector based loop closure.

point to $\mathbf{p}_j^{(i)}$ in other submaps after translation and rotation (the point pairs that are too far apart will not be included in the calculation), which is:

$$\mathbf{q}_j^{(i)} = \begin{cases} \mathbf{T}(\mathbf{p}_\ell^{(k)}), & k \neq 1 \\ \mathbf{p}_\ell^{(1)}, & \text{else} \end{cases}, \quad \text{which is the closest} \quad (30)$$

point to $\mathbf{T}(\mathbf{p}_j^{(i)})$ in other submaps ((30))

As (28) is a nonlinear optimization issue which can be resolved by Levenberg-Marquardt (LM) algorithm [40], an iterative method, and the closest point in (30) should be updated in each process of iteration.

LM algorithm can be used to solve \mathbf{x}^* , the best pose of submaps, and carries out the loop closure, as shown in Fig. 27. After the process of loop closure, the dislocated wall will be coincident on the map and the LiDAR scan locus will also be corrected (green locus in the diagram).

The loop closure algorithm is also applied to Fig. 23(b), where the detail mismatched map stitching result is shown in Fig. 28. Similarly, applying the proposed corridor detector based loop closure can achieve good mapping recovery as illustrated in Fig. 29

VI. EXPERIMENT VALIDATION AND COMPARISON

In order to verify the performance of the proposed algorithm, a couple of experiments were considered. Note that no odometry information was used for the following SLAM experiment.

First, an experiment was conducted around the DAA-3F, where the developed SLAM module was equipped on a RC car as shown in Fig. 30 and three rounds LiDAR scans were

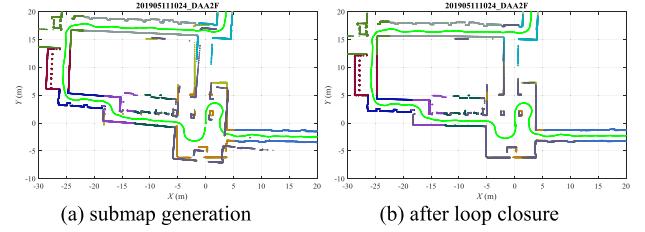


Fig. 29. Corridor detector based submap generation for DAA-2F and its loop closure result.



Fig. 30. DAA-3F experimental scene.

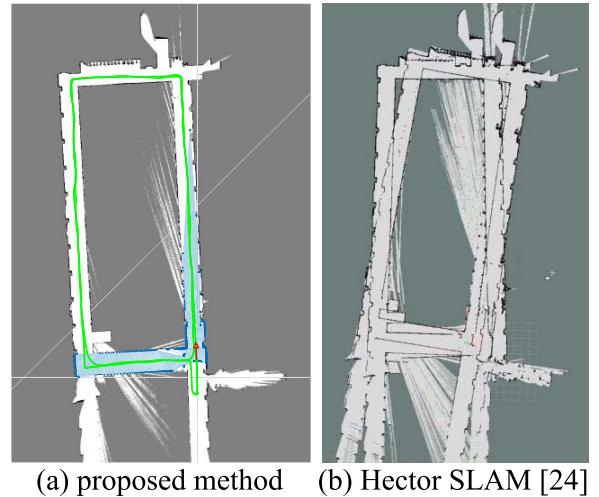


Fig. 31. SLAM performance comparison (DAA-3F).

collected. The collected dataset was also fed into the well known Hector SLAM [24] provided by the robot operation system (ROS). Compared with the Hector SLAM, Fig. 31 evidently shows that the proposed algorithm can provide more robust performance in localization and mapping.

As for the localization result of SLAM, the precision of the proposed SLAM needs to be further verified. Therefore, the localization results must be compared with the ground truth. This study also utilizes the scan data and ground truth [41], [42] provided by the Computer Science and Artificial Intelligence Laboratory (CSAIL) at the Massachusetts Institute of Technology (MIT), to verify the SLAM algorithm.

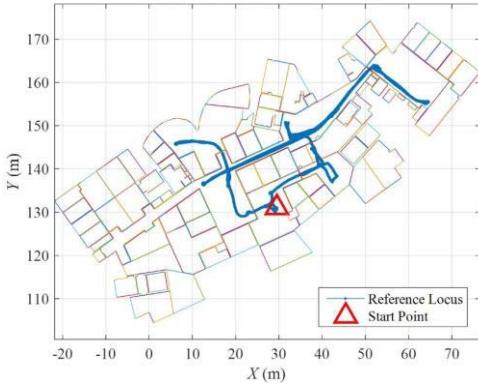


Fig. 32. Floor plan and the ground truth for scan data 2012_01_25_12_14_25.

TABLE I
SCAN DATA INFORMATION AT MIT STATA CENTER

Name of scan data	2012_01_25_12_14_25
Scan location	Stata Center 2F
Scan frequency (Hz)	20
Scan time (min:sec)	18:38
Number of scans	22294
Scan distance (m)	348

The selected scan data “2012_01_25_12_14_25”, as shown in Table 1. The scan process was carried out for about 20 minutes on the second floor of the Stata Center on January 25, 2012. The scan path can be referred to Fig. 32, and the entire scan path is about 350 m. The LiDAR used for this scan is Hokuyo UTM-30LX with a scan frequency of 20 Hz.

The following experiment uses the scan data from CSAIL to test SLAM algorithm with various parameter settings. The respective results and the ground truth provided by CSAIL will be used to calculate tracking errors (distance between the position of ground truth and estimated position) and yaw angle errors, and the advantages and disadvantages of various parameter selection will also be compared.

A. Distance Threshold of Matching Points in ICP

A localization precision comparison for the ICP distance threshold of 100, 150, 200 and 300 mm is carried out on the basis of the parameter settings in Table 2. From Fig. 33 and Table 3, it shows that the strict distance threshold of the matching points is beneficial for the removal of matched outlier which will reduce the localization error.

B. Weight of Rigid Alignment in ICP

A localization precision comparison for the weight of rigid alignment in ICP such as uniform weight ($w_i = 1, \forall i$), distance weight ${}^R w_i$ in (7), and inverse weight ${}^I w_i$ in III is carried out on the basis of the parameter settings in Table 2. The result in Fig. 34 indicates that the localization error of uniform weight starts to diverge at about 640 seconds of scan time. On the basis of Table 4, the tracking error of distance

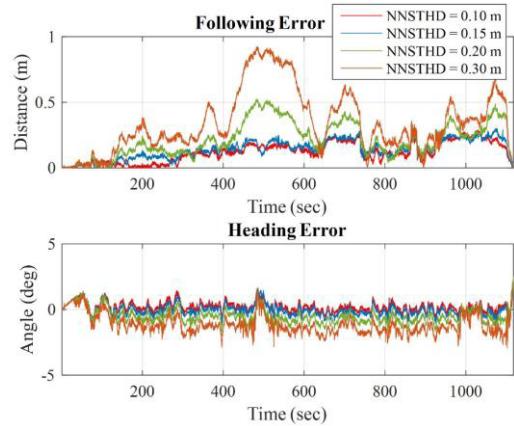


Fig. 33. Localization error with different settings of distance threshold of ICP matching point.

TABLE II
PARAMETER SETTING FOR THE BEST EXPERIMENTS

Parameter name	Parameter setting
ICP weight design	weight design (8)
Distance threshold of ICP matching point (mm)	100
Inverse sensor model and its scaling factor	Quadratic distribution model $C = 0.3$
Threshold of small scan angle (deg)	adaptive threshold (26)

TABLE III
LOCALIZATION ERROR WITH DIFFERENT SETTINGS OF DISTANCE THRESHOLD OF MATCHING POINTS

d_{THD} (mm)	100	150	200	300
Maximum of tracking error (m)	0.290	0.302	0.523	0.922
Root mean square of tracking error (m)	0.142	0.160	0.252	0.417
Maximum of yaw angle error (deg)	2.300	2.287	2.529	1.917
Root mean square of yaw angle error (deg)	0.422	0.425	0.745	1.360

weight ${}^R w_i$ and inverse weight ${}^I w_i$ are below 30 cm. As for the distance weight at 780 seconds with 40 degree/sec of LiDAR rotation, there are four scans that are not aligned with the map and results in 6 degree of yaw angle error. In contrast, the inverse weight ${}^I w_i$ is more stable on the estimation of the yaw angle. It is discovered that the weighting design in III focuses more on the scan points far from LiDAR. Thus, the influence of near scan points on pose estimation is reduced. Therefore, the scan points far from LiDAR are still able to align with the matching points in the map with the faster movement of LiDAR, and increase the bandwidth of pose estimation of the sensor.

C. Inverse Sensor Model and Scale Factor

The localization precision of different inverse sensor model with the scaling factor C of inverse sensor model is tested on the basis of the parameter setting in Table 2. If Gaussian

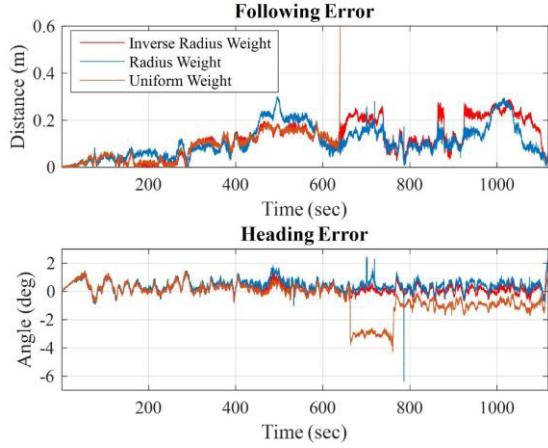


Fig. 34. Localization error with different settings of rigid alignment weight during the scanning process.

TABLE IV
LOCALIZATION ERROR WITH DIFFERENT SETTINGS
OF RIGID ALIGNMENT WEIGHT

Weight design	Uniform weight $w_i = 1$	Weight (7) r_{w_i}	Weight (8) t_{w_i}
Maximum of tracking error (m)	12.235	0.301	0.290
Root mean square of tracking error (m)	3.818	0.129	0.142
Maximum of yaw angle error (deg)	4.258	6.375	2.300
Root mean square of yaw angle error (deg)	1.118	0.579	0.422

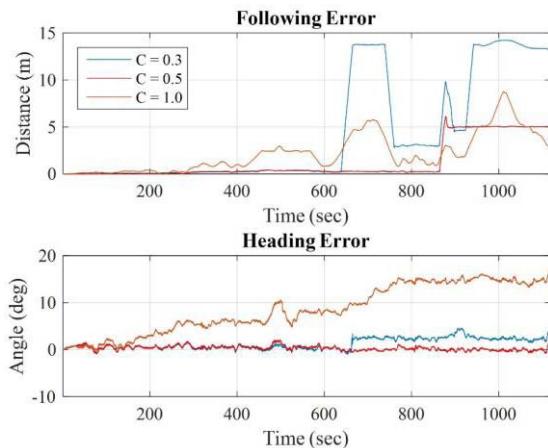


Fig. 35. Localization error with different scaling factor of Gaussian distribution model during the scanning process.

distribution model is used as the inverse sensor model for map construction with the setting C of 0.3, 0.5, and 1.0, the localization error will occur as shown in Fig. 35 and Table 5. The localization results indicate that localization divergence occurred at 640 seconds and 860 seconds, with different setting of C for Gaussian distribution model.

If the quadratic distribution model is used with the scaling factor settings of 0.3, 0.5, and 1.0, the localization error will

TABLE V
LOCALIZATION ERROR WITH DIFFERENT SCALING FACTOR OF GAUSSIAN DISTRIBUTION MODEL

Value of C in Gaussian distribution model	$C = 0.3$	$C = 0.5$	$C = 1.0$
Maximum of tracking error (m)	14.223	6.114	8.738
Root mean square of tracking error (m)	7.040	2.373	2.984
Maximum of yaw angle error (deg)	4.630	2.112	17.347
Root mean square of yaw angle error (deg)	1.688	0.574	9.989

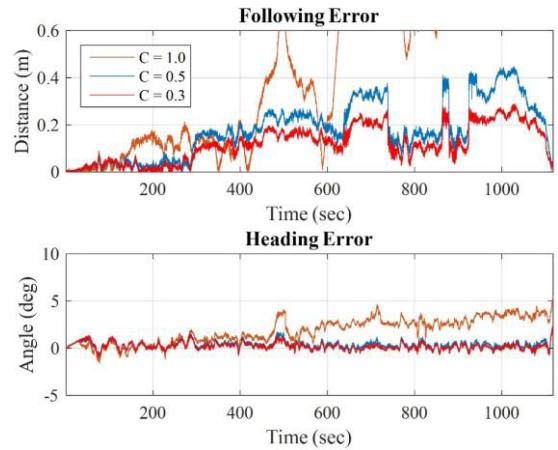


Fig. 36. Localization error with different scaling factor of quadratic distribution model.

TABLE VI
LOCALIZATION ERROR WITH DIFFERENT SCALING FACTOR OF QUADRATIC DISTRIBUTION MODEL

Value of C	$C = 0.3$	$C = 0.5$	$C = 1.0$
Maximum of tracking error (m)	0.290	0.447	2.919
Root mean square of tracking error (m)	0.142	0.215	1.060
Maximum of yaw angle error (deg)	2.300	2.536	5.573
Root mean square of yaw angle error (deg)	0.422	0.517	2.279

occur as shown in Fig. 36. With the setting of $C = 1.0$ in the quadratic distribution model, the overly fast environment recognition will reduce the robustness of the map, which will affect the localization precision. The localization error is significantly accumulated in the process, which will result in localization failure. According to Fig. 36 and Table 6 , the better setting for quadratic distribution model is $C = 0.3$ with the tracking error less than 30 cm.

On the basis of above experiment, the scaling factor C of the Gaussian distribution model in (13) and quadratic distribution model in (14) can be respectively set to 0.5 and 0.3 under the condition of 20 Hz scan updating rate, 20 mm grid resolution and width of distribution model $\sigma = 20$ mm.

As for determining whether the grids are occupied in the map, the recognition speed in quadratic distribution model is

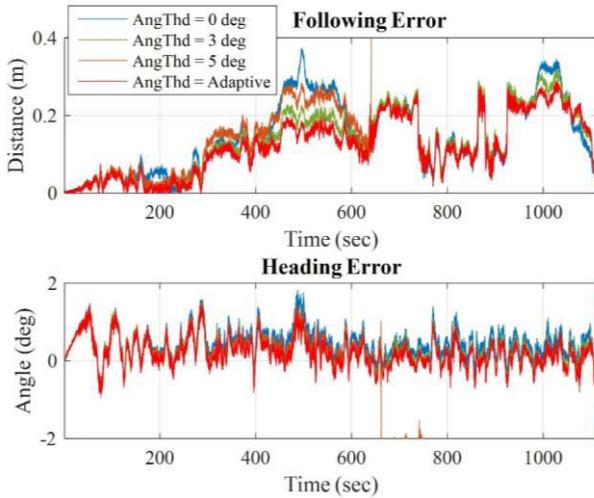


Fig. 37. Localization error with different settings for the threshold of scan angle during the scanning process.

TABLE VII
LOCALIZATION ERROR WITH DIFFERENT SETTINGS
FOR THE THRESHOLD OF SMALL SCAN ANGLE

$\Delta\theta_{Thd}$ (deg)	0	3	5	Design in (26)
Maximum of tracking error (m)	0.372	0.320	10.675	0.290
Root mean square of tracking error (m)	0.171	0.155	4.101	0.142
Maximum of yaw angle error (deg)	2.376	2.365	1.504	2.300
Root mean square of yaw angle error (deg)	0.602	0.450	1.831	0.422

slower than in the Gaussian distribution model; in contrast, once the grids are verified to be occupied, the former's reliability and robustness are higher than the latter's. Therefore, the quadratic distribution model is chosen.

D. Threshold of Unstable Scan Angle

Based on the parameter setting in Table 2, a localization precision test with the scan angle threshold settings $\Delta\theta_{Thd}$ of 0, 3, 5 degrees and the adaptive threshold in (26) is carried out. From the localization error comparison for $\Delta\theta_{Thd}$ in Fig. 37 and Table 7, it can be seen that localization with the threshold of 5 degrees starts to fail after 640 seconds. As LiDAR is going through the corridor with dimension of 13.2 m long and 1.8 m wide at the time, the removal mechanism of unstable scan points discards the scan point that is 10.3 m away from the corridor, which causes the failure of translation estimation. On the basis of all localization results, it can be seen that the self-adjusting threshold (26) effectively discards unstable scan points and reduces the pose estimation error, as well as avoids excessively discard of scan points in the corridor.

E. Loop Closure

The localization and mapping algorithm has a maximal tracking error of about 30 cm, with the parameter setting in Table 2. Therefore, the deviation exists in between the

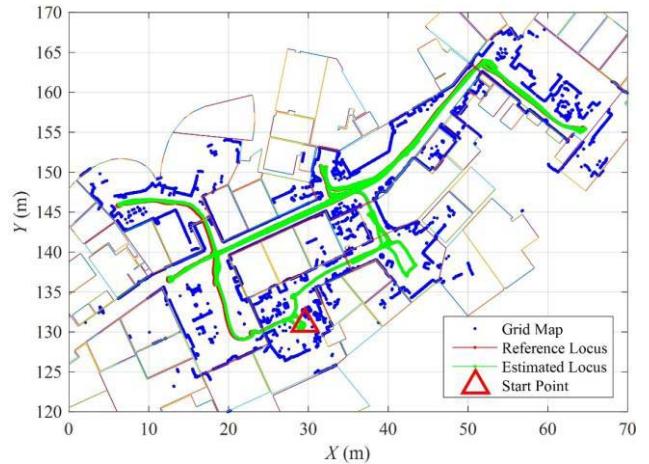


Fig. 38. A overview of SLAM result compared with ground truth.

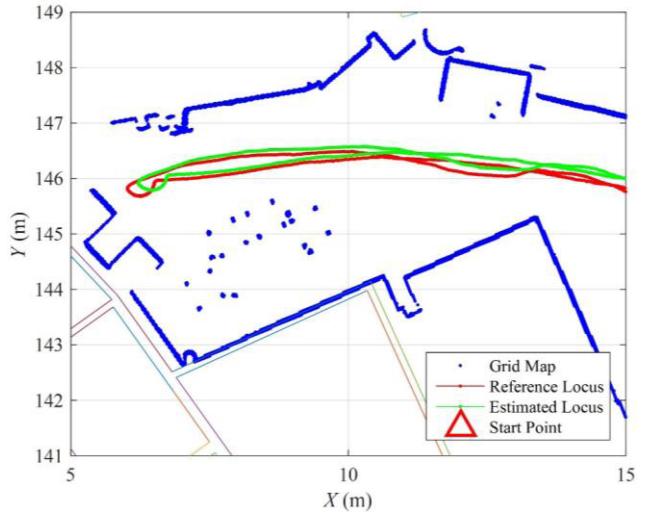


Fig. 39. Detail of SLAM result compared with ground truth.

TABLE VIII
LOCALIZATION ERROR BEFORE AND AFTER LOOP CLOSURE

Loop closure status	Before loop closure	After loop closure
Maximum of tracking error (m)	0.290	0.231
Root mean square of tracking error (m)	0.142	0.104
Maximum of yaw angle error (deg)	2.300	2.281
Root mean square of yaw angle error (deg)	0.422	0.426

estimated scan path and the ground truth path, as well as in between the constructed map and the floor plan, as shown in Fig. 38, Fig. 39. In order to correct this error, submaps are created in the interval of every 5 m along the estimated scan path based on the localization and mapping result in Fig. 40 (a), as shown in Fig. 40 (b). The loop closure is also carried out to allow the map to adjust its geometric shape and correct the cumulative errors in localization and mapping.

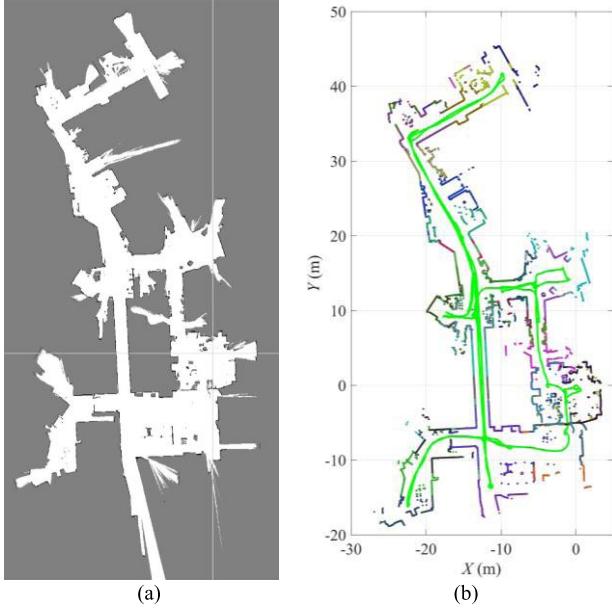


Fig. 40. (a) SLAM result and (b) segmentation of submap.

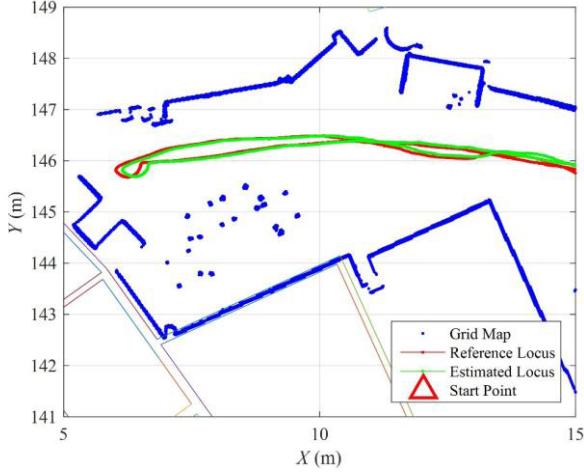


Fig. 41. Detailed localization result after loop closure.

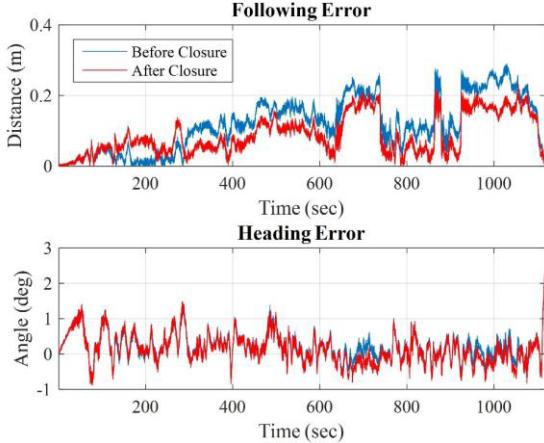


Fig. 42. Localization error before and after loop closure during the scanning process.

After loop closure, the localization error and mapping error are corrected to some extent, as shown in Fig. 41. From the localization error comparison shown in Fig. 42 and Table 8,

it can be seen that the maximum of tracking error is reduced from about 30 cm to 23 cm after the loop closure.

VII. CONCLUSION AND FUTURE WORK

This paper presents a robust 2D SLAM technology. The proposed method is capable of adapting to uncertain environment variations and adjusting SLAM parameters automatically. To enhance the real-time computation efficiency, an analytic solution of ICP is also carried out. With the aid of the corridor detector, grid map occupancy probability update, unstable points removal as well as the submap generation, a robust 2D SLAM can be achieved. In addition to our own experimental tests, we also considered the data set from MIT Stata Center dataset to test the localization precision of algorithm. As for the precision test for localization and mapping, the test results of the proposed SLAM are summarized as follows: the maximum tracking error is about 30 cm, from the scan data in the route up to about 350 m; and after the loop closure, the maximum error can be attenuated to 20 cm. Finally, since the proposed SLAM algorithm considers a single LiDAR only, it reduces the development cost and system integration effort significantly. The relative application is not only limited to wheel robotics but for portable mapping devices as well. Experiments are taken into consideration in order to verify the feasibility of the proposed method.

APPENDIX A ANALYTIC SOLUTION OF RIGID BODY ALIGNMENT

Assume that there are two sets of two-dimensional points P and Q , which are composed by $\mathbf{p}_i = [px, i \quad py, i]^T$ and $\mathbf{q}_i = [qx, i \quad qy, i]^T$ respectively ($i = 1 \sim N$). After the rotating and shifting points set P , \mathbf{p}_i and \mathbf{q}_i should coincide. The relationship between the two sets of points can be expressed as follows:

$$\mathbf{q}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t} \quad (31)$$

where \mathbf{R} is a rotation matrix and is also a 2-by-2 orthogonal matrix; \mathbf{t} is a 2-by-1 translation vector. We want to solve \mathbf{R} and \mathbf{t} on the basis of the given matching point \mathbf{p}_i and \mathbf{q}_i from the points set P and Q . The optimization problem is described as follows:

$$\begin{aligned} \min_{\mathbf{R}, \mathbf{t}} f(\mathbf{R}, \mathbf{t}) \text{ subject to } \mathbf{R}^T \mathbf{R} = \mathbf{I}, \\ f(\mathbf{R}, \mathbf{t}) = \sum_i^N w_i \| (\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i \|^2 \end{aligned} \quad (32)$$

where $f(\mathbf{R}, \mathbf{t})$ represents the cost function and w_i is the weight coefficient of the matching point \mathbf{p}_i and \mathbf{q}_i . Eq. (32) describes that the best solution for \mathbf{R} and \mathbf{t} results in the smallest sum of squares of the distances between the matching points after rigid motion. In order to solve this problem, we first expand the cost function f :

$$\begin{aligned} f(\mathbf{R}, \mathbf{t}) &= \sum_i^N w_i \| (\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i \|^2 \\ &= \left(\sum_i^N w_i \right) \mathbf{t}^T \mathbf{t} + 2 \left[\sum_i^N w_i (\mathbf{p}_i^T \mathbf{R}^T - \mathbf{q}_i^T) \right] \mathbf{t} \\ &\quad + \sum_i^N w_i (\mathbf{p}_i^T \mathbf{R}^T \mathbf{R} \mathbf{p}_i + \mathbf{q}_i^T \mathbf{q}_i) \end{aligned} \quad (33)$$

It can be seen from (33) that the cost function f is a positive definite quadratic function of \mathbf{t} , therefore the extremum of f for \mathbf{t} must be the global minimum.

Differentiating f with respect to \mathbf{t} gives:

$$\frac{\partial f}{\partial \mathbf{t}} \Big|_{\mathbf{t}=\mathbf{t}^*} = 2 \left(\sum_i^N w_i \right) \mathbf{t}^* + 2 \sum_i^N w_i (\mathbf{R}\mathbf{p}_i - \mathbf{q}_i) = \mathbf{0} \quad (34)$$

Rearranging (34), optimal translation vector \mathbf{t}^* can be obtained as follows:

$$\mathbf{t}^* = \bar{\mathbf{q}} - \mathbf{R}\bar{\mathbf{p}} \quad (35)$$

where $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$ are the centroids (or center of mass) of the points sets P and Q respectively:

$$\bar{\mathbf{q}} \equiv \frac{\sum_i^N w_i \mathbf{q}_i}{\sum_i^N w_i}, \quad \bar{\mathbf{p}} \equiv \frac{\sum_i^N w_i \mathbf{p}_i}{\sum_i^N w_i} \quad (36)$$

Therefore, as long as the obtained \mathbf{t}^* has a relation with \mathbf{R} as seen in (35), the cost function f will reach its minimum. Substitute the relation formula (35) into the cost function (33), replace the variable \mathbf{t} , and get the following:

$$f(\mathbf{R}, \mathbf{t}^*) = f(\mathbf{R}, \bar{\mathbf{q}} - \mathbf{R}\bar{\mathbf{p}}) = \sum_i^N w_i \| \mathbf{R}\mathbf{x}_i - \mathbf{y}_i \|^2 \quad (37)$$

where \mathbf{x}_i and \mathbf{y}_i are defined as the coordinates for the corresponding centroid $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$:

$$\begin{cases} \mathbf{x}_i \equiv \mathbf{p}_i - \bar{\mathbf{p}} \\ \mathbf{y}_i \equiv \mathbf{q}_i - \bar{\mathbf{q}} \end{cases} \quad (38)$$

Try to expand (37) again to get:

$$\begin{aligned} f(\mathbf{R}, \mathbf{t}^*) &= \sum_i^N w_i \| \mathbf{R}\mathbf{x}_i - \mathbf{y}_i \|^2 \\ &= \sum_i^N w_i \mathbf{x}_i^T \mathbf{x}_i - 2 \sum_i^N w_i \mathbf{y}_i^T \mathbf{R}\mathbf{x}_i + \sum_i^N w_i \mathbf{y}_i^T \mathbf{y}_i \end{aligned} \quad (39)$$

In (39), only $\sum_i^N w_i \mathbf{y}_i^T \mathbf{R}\mathbf{x}_i$ varies with the variable \mathbf{R} , so the original optimization problem in (32) can be transformed into:

$$\max_{\mathbf{R}^T \mathbf{R} = \mathbf{I}} \sum_i^N w_i \mathbf{y}_i^T \mathbf{R}\mathbf{x}_i, \quad \begin{cases} \mathbf{x}_i \equiv \mathbf{p}_i - \bar{\mathbf{p}} \\ \mathbf{y}_i \equiv \mathbf{q}_i - \bar{\mathbf{q}} \end{cases} \quad (40)$$

The rotation matrix \mathbf{R} can be solved according to (40). After the optimal rotation matrix \mathbf{R}^* is solved, then the optimal \mathbf{t}^* can be obtained based on (35).

In order to solve (40), it is necessary to expand \mathbf{R} , \mathbf{x}_i and \mathbf{y}_i in the formula. As for the planar rigid motion, only the yaw angle ψ has freedom of rotation, which is:

$$\mathbf{R} = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \quad (41)$$

In addition, the 2-by-1 vector of points coordinate \mathbf{x}_i and \mathbf{y}_i are:

$$\mathbf{x}_i = \begin{bmatrix} x_{x,i} \\ x_{y,i} \end{bmatrix}, \quad \mathbf{y}_i = \begin{bmatrix} y_{x,i} \\ y_{y,i} \end{bmatrix} \quad (42)$$

After substituting (41) and (42) into (40), we can get:

$$\sum_i^N w_i \mathbf{y}_i^T \mathbf{R}\mathbf{x}_i = \cos \psi (s_{11} + s_{22}) + \sin \psi (s_{12} - s_{21}) \quad (43)$$

where s_{11}, s_{12}, s_{21} and s_{22} are from the cross-covariance matrix of \mathbf{p}_i and \mathbf{q}_i with the weight w_i :

$$\begin{aligned} \mathbf{S} &= \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \equiv \sum_i^N w_i (\mathbf{q}_i - \bar{\mathbf{q}})(\mathbf{p}_i - \bar{\mathbf{p}})^T \\ &= \begin{bmatrix} \sum_i^N w_i x_{x,i} y_{x,i} & \sum_i^N w_i x_{x,i} y_{y,i} \\ \sum_i^N w_i x_{y,i} y_{x,i} & \sum_i^N w_i x_{y,i} y_{y,i} \end{bmatrix} \end{aligned} \quad (44)$$

The following algebra substitution is utilized to handle the trigonometric function in (43):

$$\begin{cases} \sin \gamma = (s_{11} + s_{22}) / \Gamma \\ \cos \gamma = (s_{12} - s_{21}) / \Gamma \end{cases} \quad (45)$$

where Γ is the normalization factor:

$$\Gamma = \sqrt{(s_{11} + s_{22})^2 + (s_{12} - s_{21})^2} \quad (46)$$

Substitute (45) into (43), simplify with the angle-sum identities in sine function, and get the following:

$$\sum_i^N w_i \mathbf{y}_i^T \mathbf{R}\mathbf{x}_i = \Gamma \sin(\psi + \gamma) \quad (47)$$

According to (47), $\sum_i^N w_i \mathbf{y}_i^T \mathbf{R}\mathbf{x}_i$ will reach its maximum when $\sin(\psi + \gamma)$ reaches its maximum value of 1. That is, the optimal yaw angle ψ^* must be complementary to γ , which is:

$$\sin(\psi^* + \gamma) = 1, \quad \text{as } \psi^* = \left(\frac{1}{2}\pi - \gamma \right) + 2n\pi \quad (48)$$

where $\forall n \in \mathbb{Z}$. According to (45) and (48), we can calculate the value of sine and cosine function for the best yaw angle ψ^* :

$$\begin{cases} \cos \psi^* = \sin \gamma = (s_{11} + s_{22}) / \Gamma \\ \sin \psi^* = \cos \gamma = (s_{12} - s_{21}) / \Gamma \end{cases} \quad (49)$$

Substitute (49) into (41) yields the best rotation matrix estimate \mathbf{R}^* :

$$\mathbf{R}^* = \frac{1}{\Gamma} \begin{bmatrix} s_{11} + s_{22} & s_{21} - s_{12} \\ s_{12} - s_{21} & s_{11} + s_{22} \end{bmatrix} \quad (50)$$

Therefore, the calculation process for the entire rigid alignment is: calculate the cross-covariance matrix \mathbf{S} in (44) from the given matching point \mathbf{p}_i and \mathbf{q}_i , substitute it into (50) to obtain the best rotation matrix \mathbf{R}^* , and then substitute it into (35) to obtain the translation vector \mathbf{t}^* . In this way, the cost function in (32) reaches its minimum, and the rigid alignment between the two points sets P and Q is completed.

REFERENCES

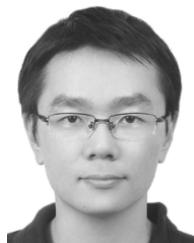
- [1] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
- [2] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE J. Robot. Autom.*, vol. 3, no. 3, pp. 249–265, Jun. 1987.
- [3] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.
- [4] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Robot. Automat.*, vol. 7, no. 3, pp. 376–382, Jun. 1991.
- [5] J. A. Castellanos and J. D. Tardos, *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Boston, MA, USA: Kluwer, 2000.

- [6] M. Csorba, *Simultaneous Localisation and Map Building*. Oxford, U.K.: Univ. Oxford, 1998.
- [7] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, Jun. 2001.
- [8] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 242–257, Jun. 2001.
- [9] J. Leonard, H. Durrant-Whyte, and I. J. Cox, "Dynamic map building for autonomous mobile robot," in *Proc. IEEE Int. Workshop Intell. Robots Syst., Towards New Frontier Appl.*, Aug. 1990, vol. 1, pp. 89–96.
- [10] P. Newman, "On the structure and solution of the simultaneous localisation and map building problem," Ph.D. dissertation, Dept. Mech. Mechatron. Eng., Austral. Centre Field Robot., Univ. Sydney, Sydney, NSW, Australia, Mar. 1999, vol. 41.
- [11] S. Williams, G. Dissanayake, and H. Durrant-Whyte, "Towards terrain-aided navigation for underwater robotics," *Adv. Robot.*, vol. 15, no. 5, pp. 533–549, Jan. 2001.
- [12] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, vol. 9, no. 2, p. 61, 1988.
- [13] S. Thrun, "Learning occupancy grids with forward models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Expanding Societal Role Robot. Next Millennium*, Oct./Nov. 2001, vol. 3, pp. 1676–1681.
- [14] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.
- [15] J. Buhmann *et al.*, "The mobile robot Rhino," *Ai Mag.*, vol. 16, no. 2, p. 31, Jun. 1995.
- [16] W. Burgard *et al.*, "Experiences with an interactive museum tour-guide robot," *Artif. Intell.*, vol. 114, no. 1, pp. 3–55, Oct. 1999.
- [17] D. Guzzoni, A. Cheyer, L. Julia, and K. Konolige, "Many robots make short work: Report of the SRI International mobile robot team," *AI Mag.*, vol. 18, no. 1, p. 55, May 1997.
- [18] S. Thrun *et al.*, "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *Int. J. Robot. Res.*, vol. 19, no. 11, pp. 972–999, Nov. 2000.
- [19] B. Yamauchi and P. Langley, "Place recognition in dynamic environments," *J. Robotic Syst.*, vol. 14, no. 2, pp. 107–120, Feb. 1997.
- [20] B. Yamauchi, P. Langley, A. C. Schultz, J. Grefenstette, and W. Adams, *Magellan: An Integrated Adaptive Architecture for Mobile Robotics*, Institute for the Study of Learning and Expertise, Palo Alto, CA, USA, 1998.
- [21] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 889–916, Sep. 2007.
- [22] M. Bosse and R. Zlot, "Map matching and data association for large-scale two-dimensional laser scan-based SLAM," *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 667–691, Jun. 2008.
- [23] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.
- [24] S. Kohlbrecher, O. V. Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *Proc. IEEE Int. Symp. Saf., Rescue Robot.*, Nov. 2011, pp. 155–160.
- [25] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1271–1278.
- [26] K. Konolige, G. Grisetti, R. Kümmeler, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 22–29.
- [27] L. Li and L. Schulze, "Comparison and evaluation of SLAM algorithms for AGV navigation," in *Proc. 8th Int. Conf. Eng. Manage.*, 2018, pp. 213–221.
- [28] R. Yagfarov, M. Ivanou, and I. Afanasyev, "Map comparison of lidar-based 2D SLAM algorithms using precise ground truth," in *Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 1979–1983.
- [29] V. Kirnos, V. Antipov, A. Priorov, and V. Kokovkina, "The LIDAR Odometry in the SLAM," in *Proc. 23rd Conf. Open Innov. Assoc. (FRUCT)*, Nov. 2018, pp. 180–185.
- [30] G. Jiang, L. Yin, G. Liu, W. Xi, and Y. Ou, "FFT-based scan-matching for SLAM applications with low-cost laser range finders," *Appl. Sci.*, vol. 9, no. 1, p. 41, Dec. 2018.
- [31] G. Xu *et al.*, "Precise point set registration using point-to-plane distance and correntropy for LiDAR based localization," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 734–739.
- [32] S. Z. Mehdi and J.-H. Ryu, "2D pose nodes sampling heuristic for fast loop closing," in *Proc. Annu. Spring Conf. Korean Soc. Automot. Eng. (KSAE)*, May 2017, pp. 111–115.
- [33] R. Opronolla, G. Fasano, M. Grassi, A. Savvaris, and A. Moccia, "PCA-based line detection from range data for mapping and localization-aiding of UAVs," *Int. J. Aerosp. Eng.*, vol. 2017, Apr. 2017, Art. no. 4241651.
- [34] D. Wu, Y. Meng, K. Zhan, and F. Ma, "A LIDAR SLAM based on point-line features for underground mining vehicle," in *Proc. Chin. Autom. Congr. (CAC)*, Nov./Dec. 2018, pp. 2879–2883.
- [35] M. Peter, S. R. U. N. Jafri, and G. Vosselman, "Line segmentation of 2D laser scanner point clouds for indoor slam based on a range of residuals," *Remote Sens. Spatial Inf. Sci.*, vol. 4, no. 2, pp. 363–369, Sep. 2017.
- [36] S. Agarwal, K. S. Parunandi, and S. Chakravorty, "Robust pose-graph SLAM using absolute orientation sensing," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 981–988, Apr. 2019.
- [37] Y.-T. Wang, C.-C. Peng, A. A. Ravankar, and A. Ravankar, "A single LiDAR-based feature fusion indoor localization algorithm," *Sensors*, vol. 18, no. 4, p. 1294, Apr. 2018.
- [38] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, p. 91, Apr. 1992.
- [39] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
- [40] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963.
- [41] MIT Computer Science Artificial Intelligence Laboratory. *MIT Stata Center Data Set*. Accessed: Jun. 7, 2018. [Online]. Available: <http://projects.csail.mit.edu/stata/>
- [42] M. Fallon, H. Johannsson, M. Kaess, and J. J. Leonard, "The MIT stata center dataset," *Int. J. Robot. Res.*, vol. 32, no. 14, pp. 1695–1699, 2013.



Li-Hsin Chen received the B.S. and M.S. degrees from the Department of Aeronautics and Astronautics (DAA), National Cheng Kung University (NCKU), Tainan, Taiwan, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree.

His research interests include system dynamics analysis, optimization, and SLAM technology.



Chao-Chung Peng was born in Kaohsiung, Taiwan, in 1980. He received the B.S. degree from the Department of Aeronautics and Astronautics (DAA), National Cheng Kung University (NCKU), Tainan, Taiwan, in 2003, and the Ph.D. degree in 2009.

From 2008 to 2009, he was a Research Assistant with the Department of Engineering, Leicester University, U.K. In 2009, he was awarded a membership from the Phi Tau Phi Scholastic Honor Society. From 2010 to 2012, he was a Post-Doctoral Fellow with the Department of Mechanical Engineering,

NCKU. In 2012, he moved to ADLINK Technology and served as a Senior Engineer in the Embedded System Development Section, Measurement and Automation Department. From 2014 to 2016, he was with the Automation and Instrumentation System Development Section, Iron and Steel R&D Department, China Steel Corporation (CSC). Since 2016, he has been an Assistant Professor with the Department of Aeronautics and Astronautics, NCKU. His research interests include high performance motion control and applications, unmanned vehicle design and advanced flight control system development, autonomous robotics and intelligence SLAM technology, and system modeling and diagnosis.