2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

2D Laser SLAM with Closed Shape Features: Fourier Series Parameterization and Submap Joining

Jiaheng Zhao[1], Tiancheng Li[1], Tong Yang[2], Liang Zhao[1], and Shoudong Huang[1]

***Abstract*— One of the valuable directions in feature based SLAM is to parameterize and estimate features accurately. In the real world, closed shape features are especially common. It is necessary to study the feature based SLAM problem on closed shape features. The main contribution of this paper is a 2D laser SLAM approach with Fourier series based feature parameterization and submap joining. In this paper, the Fourier series are introduced to parameterize irregular closed shape features and the SLAM problem with Fourier series feature parameterization is formulated. A submap joining process is also derived in order to reduce the high dependence on precise initial guess and the computing time. The proposed method has been evaluated on both synthetic and actual data and is able to obtain accurate trajectory and feature boundaries. The practical experiment also shows that our method surpasses Cartographer under certain scenarios. We also show that our method has the ability to be applied to the general environment.**

***Keywords* — Closed shape feature, Fourier series, 2D laser SLAM, Submap joining.**

I. Introduction

Feature based Simultaneous localization and mapping (SLAM) problem focuses on parameterizing objects in the environment and estimating robot locations and orientations as well as feature parameters. In the applications of 2D laser rangefinder, various features are parameterized and used in SLAM problems such as point features, circular features, and polynomial features.

For an object in the environment, researchers generally intend to either estimate a representative center of a feature or approximate the boundaries. One common sense of the former way is that the estimation of features should not be affected by their own geometrical structures, such as landmarks or similar stacks. In [1], the centers of tree trunks are regarded as point features to be estimated. The latter way takes the shape of features into account and tries to represent features via selected functions. For instance, line function [2], ellipse function [3] and curve function [4] have been studied as feature parameterization these years. However, it is not easy to seek out an appropriate way to parameterize features since simple structural objects (circle, ellipse, rectangle, etc.) are rare and feature boundaries are not always in regular shapes. In order to address such problems, some researchers seek help from polynomial functions. Gee et al. [5] utilized a cubic polynomial function to fit sparse laser points to acquire dense observation, but they do not estimate features with the polynomial function. Instead, the polynomial function is used to interpolate consecutive sparse laser observations and generate a dense model.

This paper tries to find a proper feature parameterization which can both represent closed shape features and be less sensitive to initial guess. The process of fitting shape with Fourier series is a typical application on shape retrieval in image processing [6], [7]. Shapes can be sampled first and fitted through Fast Fourier Transform. Rakshit et al. [8] used Fourier series to extract non-circular human pupil iris boundaries. Su and Xiang [9] proposed a Fourier series based approach to characterize 2D general-shape particles. Jiang et al. [10] implemented Fourier Transform to assist scan matching. But to the best of our knowledge, there are no applications and specific formulations in feature based SLAM using Fourier series to parameterize features.

Inspired by above researches, we propose a Fourier series based feature parameterization method and apply it to SLAM problem. The advantages of Fourier series include: (1) it can represent a feature's boundary accurately; (2) the number of coefficients is changeable, which makes the feature representation flexible. Truncated Fourier series removes the part of data with high frequencies, which is usually the case of random noise; (3) feature's center has less influence on the estimated boundary, which makes the result reliable since an accurate initial guess from the noisy raw data is not available.

In our previous work [11], we took pre-defined polynomial functions as features and proposed an implicit function parameterization approach to solve feature based SLAM problem. The approach can estimate features in general shape with prior knowledge and calculate the uncertainty theoretically. However, it is difficult to provide proper polynomial coefficients as an initial guess if the accurate prior knowledge of complex closed shape features is not available, resulting in being sensitive to the initial guess.

Normally, closed shape feature representation method such as implicit function based method is sensitive to initial guess. Besides parameterizing features with Fourier series, another efficient way to reduce the impact of inaccurate initial guess is submap joining. Since the size of local submaps is usually small, the noisy odometry and laser information can be locally regarded as trustful, reducing the requirement of good initial guess. Moreover, submap joining method is less time-consuming than solving the problem in full least squares way. Huang et al. [12] proposed an Extended Information Filter

[1]Jiaheng Zhao, Tiancheng Li, Liang Zhao and Shoudong Huang are with the Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Sydney, Australia. {jiaheng.zhao, tiancheng.li-1}@student.uts.edu.au; {liang.zhao, shoudong.huang}@uts.edu.au;

[2]Tong Yang is with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, Hangzhou, P.R. China. tong.yang@zju.edu.cn

based sparse local map submap joining approach, then they extended the application using multiple iterations to improve the consistency [13]. Chen et al. [14] applied submap joining to improve the real-time ability of their algorithm. Ni et al. [15] proposed a divide-and-conquer scheme to solve the optimizing problem. Zhan et al. [16] presented a method to construct a global map via corresponding undirected connected graph. Nevertheless, all of them focused on point features. Wang et al. [17] combined points, patches and planes in local map building and proposed a submap joining method using both point and plane features, but their work focused on vision SLAM and cannot represent closed shape features.

The main contributions of this paper are as follows:

- We propose a feature based SLAM approach implementing Fourier series to parameterize closed shape features (Section III). Results (Section V) show that using Fourier series as feature parameterization increases the accuracy of estimated trajectory as well as providing precise feature boundaries.
- We formulate submap joining problem with closed shape features represented by Fourier series parameterization and develop a submap joining framework (Section IV). Results (Section V) demonstrate the improvement of importing submap joining method.

II. Closed Shape Feature Parameterization with Fourier Series

This section elaborates the notation used in this paper and the feature parameterization method via Fourier series.

A. Notation and annotation

In this paper, the semicolon is to represent vertical vector concatenation. An observed point in the jth frame is defined as ${}^{\{j\}}\mathbf{p} \in \mathbb{R}^2$. A 2D point set observed at the jth frame of feature k is denoted by ${}^{\{j\}}\mathbf{P}_k = [{}^{\{j\}}\mathbf{p}_1^k, \cdots, {}^{\{j\}}\mathbf{p}_M^k]$. We also assume observed points have zero-mean Gaussian noise $\mathbf{n}_z \in \mathbb{R}^2 \sim N(\mathbf{0}, \Sigma_z)$.

For an angle $\phi \in [-\pi, \pi)$, let $R(\phi) \in SO(2)$ be the corresponding rotation matrix which is abbreviated as R. The translation is denoted by $\mathbf{t} = [t_x, t_y]^\intercal$. $R_{ij}, \mathbf{t}_{ij}$ means the rotation and translation from frame i to frame j. If i is the global frame $\{G\}$, it is usually omitted to simplify the formula.

Suppose a robot pose $\mathbf{\Xi}_j = [\mathbf{t}_j; \phi_j]$, we use $T(\mathbf{\Xi}_j, {}^{\{j\}}\chi)$ to represent the process of transforming a point/point cloud/feature from frame $\{j\}$ to global frame $\{G\}$, and $T^{-1}(\mathbf{\Xi}_j, {}^{\{G\}}\chi)$ indicates the inverse process. Suppose a feature $\mathbf{\Phi}_k$ is in closed shape, whose boundary point set is denoted by $\mathbf{P}_k$.

B. Fourier series

Fourier series can be used to represent a periodic function via a sum of sine and cosine functions. For example, suppose $f(x)$ is a continuous function with period T and x is the indeterminate. $f(x)$ can be decomposed into triangular functions with different periods as :

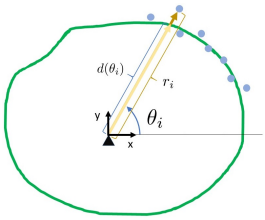


Fig. 1. Illustration of $r_i, \theta_i, d(\theta_i)$. The black triangle is the feature center.

$$f(x) := \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\frac{2\pi n x}{T} + \sum_{n=1}^{\infty} b_n \sin\frac{2\pi n x}{T} \quad (1)$$

C. Coefficients estimation

In this section, we discuss the estimation of coefficients from points within one time sequence and omit the superscript for simplification. Given the boundary set $\mathbf{P}_k$, the estimation of Fourier series coefficients a_n, b_n, are well defined and can be calculated. As shown in Fig. 1, denote r_i as the actual distance from point $\mathbf{p}_i^k$ to the feature's center, θ_i^k as the corresponding angle of $\mathbf{p}_i^k$ with respect to the x-axis in global frame. We use $d(\theta_i^k)$ to denote the fitted distance from the known or assumed center $\mathbf{C}_k$ to the ith point on the boundary, where $\theta_i^k \in [-\pi, \pi)$. The local feature coordinate's x-axis points from the center to right. The observed points on the feature boundary is sorted in ascending angle during fitting. Based on the above conception, we assume:

Assumption 1: All features in the environment are in closed shapes (as shown in Fig. 9a). For any bearing beam pointing from the feature center to infinity, it has only one intersection with the feature's boundary.

The periodic distance function $d(\theta_i)$ of a feature can be represented by:

$$d(\theta_i) = \sum_{n=0}^{N} [a_n \cos(n\theta_i) + b_n \sin(n\theta_i)] \quad (2)$$

where the period is 2π according to Eq. (1). N can be any positive finite number. The larger the N, the closer the fitted boundary is to the observed points.

The coefficients of the Fourier series is a group of numbers such that the estimated boundary and the observed data has minimal difference, i.e., the solution of the following cost function [8]:

$$\underset{a_n, b_n}{\operatorname{argmin}} \; F = \sum_{i=1}^{M} (d(\theta_i) - r_i)^2 \quad (3)$$

where M is the size of $\mathbf{P}_k$. This is a linear least squares problem and has a closed-form solution.

D. Partial observation complement and center estimation

According to Section II-C, the estimation of Fourier series coefficients is based on the assumption that all points are referenced at the feature's center. Therefore, feature points

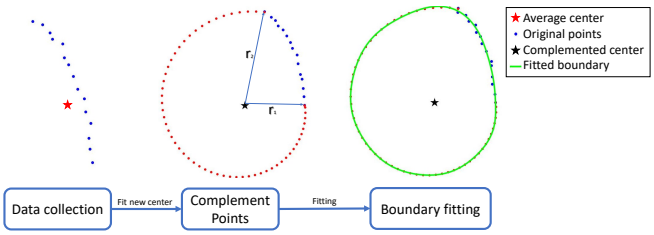


Fig. 2. Coefficients fitting process.

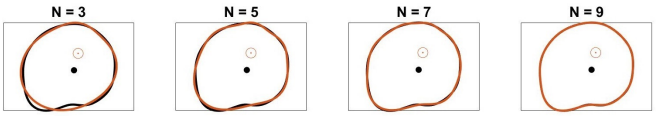


Fig. 3. Different feature centers result in the same boundary. The larger the N is, the closer the fitted boundary is to the ground truth. Black line is the ground truth, red line is the fitted boundary, black dot is the actual center, and the red circle-dot is the selected center.

need to be converted from global frame to feature's local coordinate in advance.

For any feature point set $\mathbf{P}_k$ with the center $\mathbf{C}_k$, the corresponding points in feature's local coordinate are denoted as $\hat{\mathbf{P}}_k$. One possible way to get $\mathbf{C}_k$ is finding the average center of $\mathbf{P}_k$. Thus for each point within the point set, $\hat{\mathbf{p}}_i^k = \mathbf{p}_i^k - \mathbf{C}_k$, where $\mathbf{C}_k = [C_{k,x}, C_{k,y}]^\intercal$ is the center of the point set belonging to the feature $\mathbf{\Phi}_k$.

However, one problem of this way is that the center will be shifted if only part of the boundary is observed, leading to wrong fitting. Starting from this concern, we propose a complement process. As shown in Fig. 2, partial points will lead to a bad average center for fitting. Thus, we initially use a circle function to approximately find a center as the complemented center. Then partial points are transformed to the frame defined in the complemented center, after which all points are sorted in the order of ascending angles θ. Next, the starting point and the end point among these partial points are selected (with the help of ascending angles), denoting their distances to the feature's center as r_1 and r_2, respectively. The complemented points can be predicted by interpolating r_1 and r_2 linearly. Finally, the boundary is fitted using original partial points and the complemented points together.

The complemented center is still inaccurate as the unknown points are unpredictable. The complemented center could be different from the actual center. Fortunately, thanks to the advantage of Fourier series fitting, it is reasonable to process center complement since the fitted boundary is less-influenced by the selected feature center. As shown in Fig. 3, the boundary is well-fitted among different selected fitting centers even if the center is not located at the ground truth. The only condition to the selected center is that it must locate within the boundary and satisfy Assumption 1.

The complemented points are helpful during initializing Fourier series. Note that once the coefficients are initialized, the complemented points will be dropped and do not join in the SLAM optimization. With the initialized coefficients the estimation process is able to deal with features which can only be observed partially.

E. Choice of N

N is a chosen number so that using truncated Fourier series to estimate the boundary of the feature possesses enough precision, and N can be increased to acquire a more accurate result. As shown in Fig. 3, the fitted boundary is fitted well enough for $N \geq 7$, while $N = 3$ shows an approximated boundary. As N increases, the resultant boundary gets closer to the actual points. Note that N should be limited to less than the number of fitting points, otherwise the result is over-fitting causing the boundary oscillating. Empirically, N can be set to 5 or 7 for most of the smooth closed shape features, and it can also be set to 15 or 17 for most of the rectangular features.

III. SLAM Problem Formulation with Fourier Series Feature Parameterization

In this section, the SLAM problem of feature parameterization with Fourier series is formulated in details.

The original observation in our approach is raw laser points. One inevitable issue is that it is difficult to acquire correct data association in general. However, considering the application environment of our approach, we are able to associate laser points on the same feature aided by odometry information. Hence, an assumption on data association is made to make formulations easy to understand:

Assumption 2: The raw laser points are segmented to point sets corresponding to different features. The data association is done by clustering points.

The raw observation points represent a real position in Euclidean space and they cannot be used directly to estimate coefficients of Fourier series. Consequently, laser points in Euclidean coordinates should be transformed to each feature's local polar coordinate first, and the center of these points is kept for the following optimization.

Assume the initial pose $\mathbf{\Xi}_0 = [0; 0; 0]$ coincides with the global frame, and a point set ${}^{\{j\}}\mathbf{P}_k$ with a size of M_k^j observed in the jth frame belongs to feature $\mathbf{\Phi}_k$. For the pose set $\mathbf{\Xi}_{j=1:L}$, we aim to optimize vector $\mathbf{X} = [\mathbf{\Xi}_{j=1:L}; \mathbf{\Phi}_k]$ where $\mathbf{\Phi}_k \overset{\text{def.}}{=} [\mathbf{C}_k; \mathbf{V}_k]$ and $\mathbf{V}_k = [a_0^k, a_1^k, b_1^k, \cdots, a_N^k, b_N^k]^\intercal$. Then the optimization problem is to minimize the following energy function:

$$\underset{\mathbf{X}}{\operatorname{argmin}} F(\mathbf{X}) = \sum_{\mathbf{\Xi}_{j=1}}^{L} \left(E_{\mathrm{o},j} + \sum_{k}^{K} \left[E_{\mathrm{f},j,k} + E_{\mathrm{c},j,k} \right] \right) \quad (4)$$

$E_{\mathrm{o},j}$ is the energy cost of the odometry, $E_{\mathrm{f},j,k}$ is the distance cost between estimated boundaries and observed points, $E_{\mathrm{c},j,k}$ is the energy cost of observed feature centers fitted

at each step. Each term in Eq. (4) is defined by:

$$
\begin{aligned}
E_{\mathrm{o},j} &= \|f(\mathbf{Z}_{\mathrm{o},j}, \boldsymbol{\Xi}_{j-1}, \boldsymbol{\Xi}_j)\|^2_{\Sigma^{-1}_{\mathrm{o},j}} \\
E_{\mathrm{f},j,k} &= \left\| \sum_{n=0}^{N} \left[a_n^k \cos(n\boldsymbol{\theta}^{jk}) + b_n^k \sin(n\boldsymbol{\theta}^{jk}) \right] - \mathbf{r}^{jk} \right\|^2_{\Sigma^{-1}_{\mathrm{f},j,k}} \\
E_{\mathrm{c},j,k} &= \left\| T^{-1}(\boldsymbol{\Xi}_j, \mathbf{C}_k) - {}^{\{j\}}\mathbf{C}_k \right\|^2_{\Sigma^{-1}_{\mathrm{c},j,k}}
\end{aligned}
\tag{5}
$$

where $\mathbf{Z}_{\mathrm{o},j}$ is the observation vector of jth odometry, $\Sigma^{-1}_{\mathrm{o},j}$ is the odometry covariance at the jth step. $\Sigma^{-1}_{\mathrm{f},j,k}$ is the covariance of feature k's boundary error at jth step. $\Sigma^{-1}_{\mathrm{c},j,k}$ is the covariance of fitted center at jth step. ${}^{\{j\}}\mathbf{C}_k$ is obtained through a circle-fitting process at each time sequence. The coefficients a_n^k and b_n^k are initialized by the complemented points. $\boldsymbol{\theta}^{jk} = [\theta_1^{jk}, \cdots, \theta_{M_k^j}^{jk}]^\intercal$ and $\mathbf{r}^{jk} = [r_1^{jk}, \cdots, r_{M_k^j}^{jk}]^\intercal$, where θ_i^{jk} and r_i^{jk} are calculated by:

$$
\begin{aligned}
\theta_i^{jk} &= \arctan\left(\frac{\hat{y}_i^{jk}}{\hat{x}_i^{jk}}\right) \quad , \quad r_i^{jk} = \sqrt{(\hat{x}_i^{jk})^2 + (\hat{y}_i^{jk})^2} \\
\begin{bmatrix} \hat{x}_i^{jk} \\ \hat{y}_i^{jk} \end{bmatrix} &= \begin{bmatrix} c_j {}^{\{j\}}x_i^k - s_j {}^{\{j\}}y_i^k + t_{j,x} - C_{k,x} \\ s_j {}^{\{j\}}x_i^k + c_j {}^{\{j\}}y_i^k + t_{j,y} - C_{k,y} \end{bmatrix}
\end{aligned}
\tag{6}
$$

where $\hat{\mathbf{p}}_i^{jk} = [\hat{x}_i^{jk}, \hat{y}_i^{jk}]^\intercal$ is the ith point in feature k transformed from the jth frame, and θ_i^{jk} is the angle of $\hat{\mathbf{p}}_i^{jk}$, r_i^{jk} is the distance from $\hat{\mathbf{p}}_i^{jk}$ to the feature's center $\mathbf{C}_k$, and both of them are calculated by global points transformed from the jth frame. c_j, s_j represent $\cos\phi_j$ and $\sin\phi_j$.

Eq. (4) is able to be solved by iterated methods such as Newton's method or Levenberg-Marquardt algorithm. In this paper, Levenberg-Marquardt algorithm is taken as the problem solver. The covariance matrices $\Sigma_{\mathrm{o},j}$, $\Sigma_{\mathrm{f},j,k}$ and $\Sigma_{\mathrm{c},j,k}$ of the problem will change in each iteration, which are used as the weights for the three terms in Eq. (4). The specific derivation of the covariance matrices as well as the Jacobian matrix can be found in the Appendix.

IV. Submap Joining

Generally, a SLAM problem considering all the poses and features in the state vector requires a relatively precise initial guess for iteratively solving. Submap joining is an efficient way to avoid such high dependence on precise initial guess. This section explains local map building and how to join local maps with Fourier series feature parameterization.

A. Local map building

A local map $\mathcal{L}_l$ is denoted by:

$$
\mathcal{L}_l := \left\{ {}^{\{\mathcal{L}_l\}}\boldsymbol{\Xi}_e, {}^{\{\mathcal{L}_l\}}\bar{\boldsymbol{\Phi}}_1, \cdots, {}^{\{\mathcal{L}_l\}}\bar{\boldsymbol{\Phi}}_k \right\} \tag{7}
$$

where $\mathcal{L}_l$ denotes the lth local map, $\{\mathcal{L}_l\}$ is the frame of $\mathcal{L}_l$ and is defined at the first robot pose within the local map, ${}^{\{\mathcal{L}_l\}}\boldsymbol{\Xi}_e$ is the robot end pose in $\mathcal{L}_l$, ${}^{\{\mathcal{L}_l\}}\bar{\boldsymbol{\Phi}}_k$ are local features and ${}^{\{\mathcal{L}_l\}}\bar{\boldsymbol{\Phi}}_k \overset{\text{def.}}{=} [{}^{\{\mathcal{L}_l\}}\mathbf{C}_k, {}^{\{\mathcal{L}_l\}}\mathbf{P}_k]$. ${}^{\{\mathcal{L}_l\}}\mathbf{P}_k$ denotes all the points transformed to the robot end pose, and these points are observed at each pose. Similar to [12], the robot end pose of local map $\mathcal{L}_{l-1}$ is the same as the robot start pose of local map $\mathcal{L}_l$, and the first local map starts at the same origin as the global map (as is shown in Fig. 4).

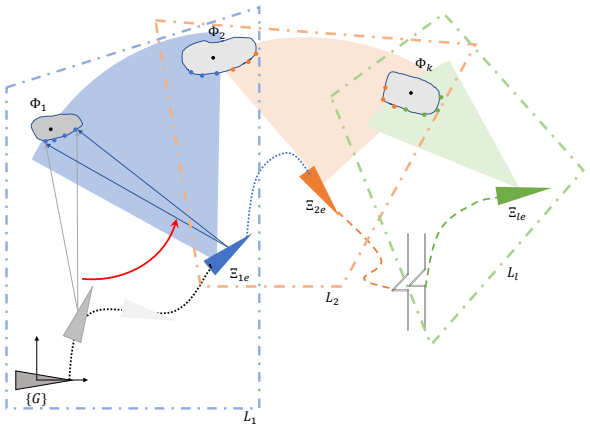


Fig. 4. Illustration of submap joining. The origin of the first local map $\mathcal{L}_1$ coincides with the global map frame. The robot end pose of each local map (e.g. $\mathcal{L}_1$) is the robot start pose of the next local map (e.g. $\mathcal{L}_2$). A local map is build by a series robot poses and feature observations.

A new local map is built following two rules: (1) after the robot has several new observations, or (2) after the robot moves for a certain distance. The building of each local map can be regarded as a "full" SLAM problem starting at the first local robot pose. The problem can be solved via Eq. (4). Once all the poses and features are optimized, the estimated feature's centers and laser points are transformed to the robot end pose of this local map. After the local map is built, all the poses except the robot end pose are removed, as well as the calculated Fourier series coefficients. The results of one local map consist of transformed points, feature centers, and the robot end pose.

B. Submap joining process

After all the local maps are built, submap joining process will optimize the robot end poses and features by considering all the local maps as integrated observations so as to obtain estimated global Fourier series coefficients. Consider a simplified case: l local maps are built containing K features. The state vector $\mathbf{X}^{\mathcal{G}}$ in submap joining process is defined as

$$
\mathbf{X}^{\mathcal{G}} = [\boldsymbol{\Xi}_{1e}; \cdots; \boldsymbol{\Xi}_{le}; \boldsymbol{\Phi}_1; \cdots; \boldsymbol{\Phi}_K] \tag{8}
$$

where $\boldsymbol{\Xi}_{1e}, \cdots, \boldsymbol{\Xi}_{le}$ are the global robot end poses of each local maps; $\boldsymbol{\Phi}_1, \cdots, \boldsymbol{\Phi}_K$ are global features.

The aim of submap joining is to obtain the global map by merging and optimizing local maps. The optimization problem is:

$$
\underset{\mathbf{X}^{\mathcal{G}}}{\operatorname{argmin}} M(\mathbf{X}^{\mathcal{G}}) = \sum_{\boldsymbol{\Xi}_{je}, j=1}^{l} \left(\tilde{E}_{\mathrm{o},je} + \sum_{k=1}^{K} \left(\tilde{E}_{\mathrm{f},j,k} + \tilde{E}_{\mathrm{c},j,k} \right) \right) \tag{9}
$$

where

$$\begin{aligned}\tilde{E}_{\mathrm{o},je} &= \left\| f(\mathbf{Z}_{\mathrm{o},je}, \boldsymbol{\Xi}_{(j-1)e}, \boldsymbol{\Xi}_{je}) \right\|^2_{\tilde{\Sigma}^{-1}_{\mathrm{o},je}} \\ \tilde{E}_{\mathrm{f},j,k} &= \left\| d(\boldsymbol{\theta}^{jk}) - \mathbf{r}^{jk} \right\|^2_{\tilde{\Sigma}^{-1}_{\mathrm{f},j,k}} \\ \tilde{E}_{\mathrm{c},j,k} &= \left\| T^{-1}(\boldsymbol{\Xi}_{je}, \mathbf{C}_k) - {}^{\{j\}}\mathbf{C}_k \right\|^2_{\tilde{\Sigma}^{-1}_{\mathrm{c},j,k}}\end{aligned} \tag{10}$$

Noting that $\mathbf{Z}_{\mathrm{o},je}$ is the estimated robot end pose ${}^{\{\mathcal{L}_j\}}\boldsymbol{\Xi}_e$ of local map $\mathcal{L}_j$, and $f(\cdot)$ is under the same definition as Eq. (5). $\boldsymbol{\theta}^{jk}$ and $\boldsymbol{r}^{jk}$ are defined similarly to Eq. (6) with respect to the jth robot end pose. Since the calculation of covariance matrices $\tilde{\Sigma}_{\mathrm{o},je}$, $\tilde{\Sigma}_{\mathrm{f},j,k}$ and $\tilde{\Sigma}_{\mathrm{c},j,k}$ is quite similar to that in Eq. (5) (given in the Appendix), we omit the specific derivation in this paper.

A larger N in building local maps and a smaller N in submap joining process is helpful during submap joining. The accumulated error caused by sensor noises does not influence results significantly because of the limited size of poses when building local maps. Hence, a larger N contributes to more precise feature boundaries and more accurate local maps. On the other hand, joining maps has to handle all the points corresponding to every feature from each local map, where the accumulated error is really large. In this situation, a smaller N can lower the impact of noisy points both in fitting boundaries and solving the optimization problem. Moreover, a smaller N can also reduce the calculation time because of the decreased size of state vectors.

V. Experiments

In this section several simulations and experiments were performed to analyze the performance of proposed methods: firstly, we evaluated the validity of the Fourier series represented SLAM algorithm and compared with the method using polynomial parameterization [11] (for simplification, we take "Implicit" as an abbreviation of polynomial parameterization based SLAM method, and denote "FS" as our approach); secondly, the map was compared between cases whether implementing our submap joining method. In the end of this section, two experiments were conducted to validate the feasibility of our method on a more general environment. The algorithm was tested on the laptop with Intel i7, 16GB RAM. All the code are finished in MATLAB.

A. Simulation environment

The simulation environment shown in Fig. 5 is a 15m×10m space containing several irregular closed shape features. The robot starts at $[0, 0, 0]^\intercal$ and odometry information is provided via a virtual wheel encoder with a random Gaussian noise $N(0, \mathrm{diag}[0.05^2, 0.05^2, 4e^{-6}])$ for (x, y, ϕ). The noise of raw observed points is a random Gaussian noise $N(0, \mathrm{diag}[0.05^2, 0.05^2])$. A 2D lidar is simulated with the valid range of 10m and the angle resolution of $0.33°$. Range-Azimuth model is adopted for simulation, but the range-bearing data is transformred to Cartesian coordinate to form the observation. Only points within valid range and hit on features can be observed. In order to evaluate the accuracy of the feature coefficients estimate, we need the ground truth of the Fourier series coefficients. The feature's ground truth are generated by three steps: (a) set feature's ground truth center and manually select 30 boundary points for each feature; (b) fit points with Fourier series with a preset N; (c) use the fitted Fourier series as the ground truth to generate the feature shape using Eq. (2).

B. Accuracy evaluation

The results of Implicit and FS are compared and shown in Fig. 5. Both algorithms share the same initial guess of the robot poses[1]. A second order implicit function was used in the Implicit method. $N = 7$ was used in our FS algorithm.

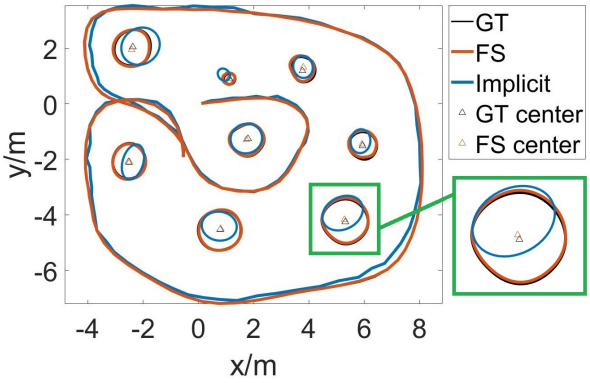


Fig. 5. Trajectory comparison between Implicit and FS.

The estimated trajectories of both the two methods are close to the ground truth, while the one obtained from Implicit drifts at the bottom. Features estimated in FS method coincide approximately with the ground truth features' boundary, which illustrates FS method has the ability to represent real features as closely as possible. It should be noted that the estimated centers in our approach do not have a significant influence on the estimation process. Even though the estimated center does not always locate at the feature's real center, the estimated boundary still shows a good performance.

Fig. 6. Pose error of every step. The dash line in each sub figure is the average difference between the estimated result and the ground truth for all the steps.

[1]Since Implicit method is sensitive to initial guess, we use actual poses with perturbation as initial guess for both methods.

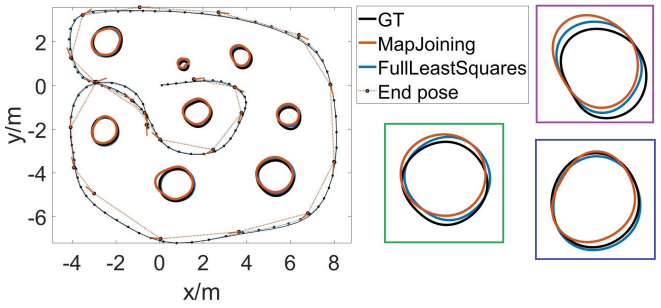


Fig. 7. Submap joining. Each point-line marker denotes the end pose of the local map with respect to the global frame.

Fig. 6 illustrates pose error of each step. It can be seen that the average error of FS in x, y and θ is smaller than that of Implicit. The final Root Mean Square Error (RMSE) in translation and rotation of FS are 0.0526m and 0.01rad, while that of Implicit are 0.1513m and 0.0125rad.

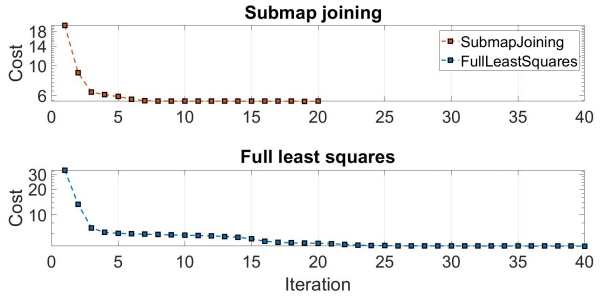


Fig. 8. Cost value changes with iteration. The iteration number of Full least squares is 148 and truncated at 40. Submap joining stops after 20 iterations. The Y axis is scaled by logarithm operation. It is clear that Full least squares continues iterating from 5 to 16 with slight changes.

C. Submap joining result

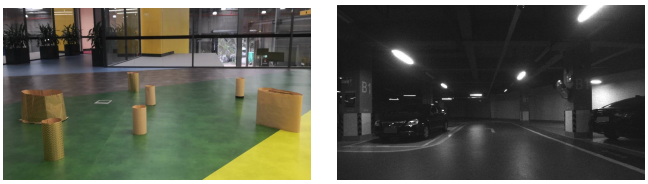
In this part, a local map was built every 5 valid sequences (a sequence is regarded as valid only if the odometry increment is beyond a threshold. In this paper, the thresholds for translation and rotation are set to 0.1m and 2°). Within each local map, the accumulated error caused by odometry noise is small, so the odometry information can be approximately regarded as accurate. As a result, the algorithm can directly import odometry as the initial guess without importing unexpected local minima.

We evaluated submap joining method with method utilizing full least squares in two aspects: map performance and time consumption. Both methods were provided with the same odometry and weights, and run 50 times with different random noise to compare average runtime.

The results of one run are shown in Fig. 7. In contrast to the result of full least squares, the estimated features of submap joining undergo subtle changes against the ground truth features, while the full least squares result is closer to the ground truth.

However, submap joining process wins considering time consumption. The average runtime among 50 times experiments of full least squares is 17.133 seconds, while the submap joining method is 7.3657 seconds.

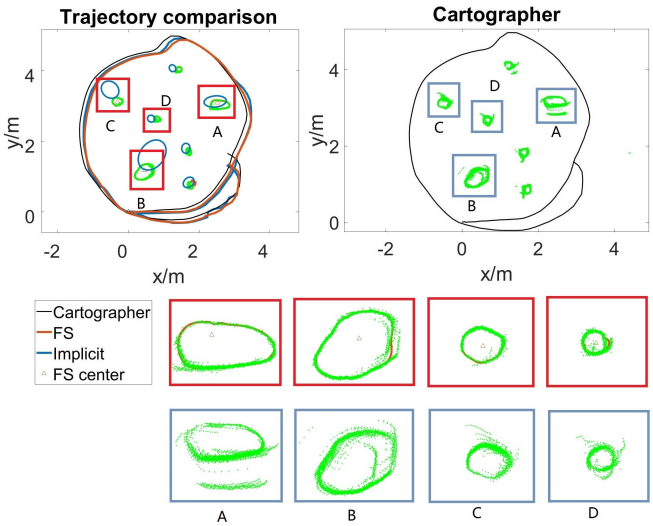
We also compared the cost function value changes w.r.t. iteration number in Fig. 8. It can be found that full least squares does not descend as quickly as submap joining, and full least squares need more iterations to get converged, which proves that submap joining effectively improves the calculation performance.



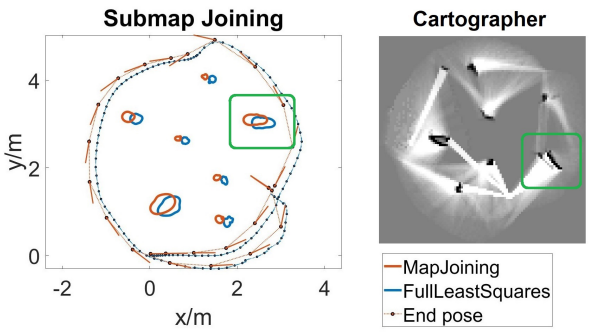
(a) Scenario A: a lounge consisting of irregular closed shape features.

(b) Scenario B: underground car park.

Fig. 9. Environment of practical experiments.



(a) Trajectory comparsion. All laser points are back-projected to the global frame. The first row is the result of FS, and the second row is the result of Cartographer.



(b) Submap joining result and the map from Cartographer.

Fig. 10. Practical experiment.

D. Experiment with artificial features

Our approach was tested on a real scenario, as is shown in Fig. 9a. The experiment was conducted in a lounge (about

20m $\times$ 30m) consisting of several irregular features (all of which are manually made) and glass walls. The data was collected via a Fetch robot [18]. The valid laser range is set to 4m to filter outlier laser points hit on or pass through the glass walls. The data association was executed by clustering the discrete points of features. We projected the points back to the initial frame via odometry information since the number of features was known and features were sparsely placed. After that, points of each feature were segmented and associated among all time sequences.

The experiment compared our method with Implicit method and the state of the art Cartographer [19]. The results[2] are depicted in Fig. 10a and Fig. 10b. In Fig. 10a, the top-left image shows the difference in trajectory among three methods. Since the ground truth is not accessible, one intuitive way to compare trajectories is to evaluate the performance of back-projected laser points via estimated poses of all the methods. The top-right image shows the back-projected laser points of Cartographer. Four rectangle areas are highlighted. By comparing from region A to region D, it is clear to see that our method provides a close-to-real boundary, while Cartographer shows a larger dispersion at the end of the trajectory. Although the trajectory of Implicit method is close to our method, it shows inconsistency in the bottom of the trajectory.

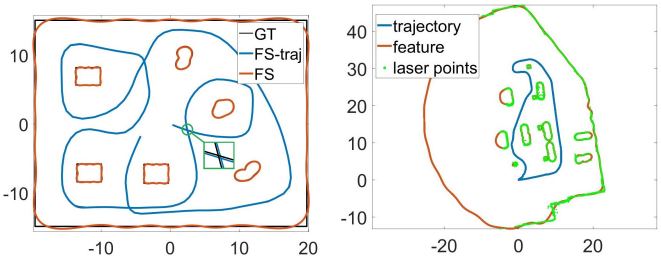
Fig. 10b shows the map of submap joining, full least squares and Cartographer, respectively. Compared with full least squares, submap joining method provides a better feature boundary even if its trajectory drifts slightly, and the runtime is less than that of full least squares. In the circled area of Cartographer's map, it can be found that the map of Cartographer is ambiguous.

E. Experiments with more general features

In this part, we intend to illustrate the validity of our method in general environment. Two experiments were conducted in this section. One is a simulated environment consisting of rectangles, irregular closed shape features and a rectangular room boundary. The simulation properties were set the same as the first simulation. The number of Fourier series coefficients for each type of features were set differently. In this example, N of irregular closed shape features, rectangles and room boundary is 5, 15 and 33, respectively. The result is shown in Fig. 11a. Clearly, the irregular features are close to the ground truth, while the fitted rectangular features oscillate at the boundary. Nevertheless, it can still represent the feature reasonably well.

The second environment is an underground car park with width of about 50m and length of over 80m (the environment is shown in Fig. 9b, and the result is shown in Fig. 11b). The robot was driven along the aisle and passed through the parking place, providing laser scan messages with a valid range of 20m and odometry messages. Since the main purpose of this part is to prove our method can be used under actual application, we undertook strict data pre-processing such as manually selecting interesting features, filtering environment noise and assigning correct data association using prior information. It can be found that there are no observed points in the left part of the figure but the fitted wall boundary still exists, this is caused by our feature initialization strategy. The estimated boundary in the left part has no meaning since no observations occur. But the estimated boundary is close to the actual shape where the observations exist.

We use the two experiments to validate that our method is able to be adapted to more general cases.



(a) Result: simulated environment. (b) Result: underground car park.

Fig. 11. General environment experiments. The estimated boundary can represent actual features if observation is sufficient. It can also handle not-closed observed features.

VI. Conclusion

In this paper, we proposed a novel 2D feature based SLAM approach utilizing Fourier series as feature parameterization. We also formulated a submap joining method with the feature parameterization. Compared to implicit function based method, simulated experiments concluded that our method does not rely significantly on initial guess and can provide close-to-real feature boundaries. Moreover, submap joining method is able to speed up the calculation without leading to an unacceptable result. Practical experiment shows that our method surpass Cartographer under certain scenarios. We also shows that our method has the ability to be applied to the general environment.

In the future, we will extend our method to be adapted to more general features and loose the application constraints. We also consider to evaluate our method on public datasets.

Appendix

This appendix derived the Jacobian matrix and covariance matrices of Eq. (5). Since a common odometry model is utilized, we only consider the feature part for simplification purpose. Denote dR_j as the derivative of R_j with respect to ϕ_j and let B and D represent:

$$\begin{aligned} B &= \sum_{n=0}^{N} \left[a_n^k \cos(n\boldsymbol{\theta}^{jk}) + b_n^k \sin(n\boldsymbol{\theta}^{jk})\right] - \mathbf{r}^{jk} \\ D &= T^{-1}(\boldsymbol{\Xi}_{je}, \mathbf{C}_k) - {}^{\{j\}}\mathbf{C}_k \end{aligned} \tag{11}$$

The Jacobian of D is $J_D = \begin{bmatrix} \frac{\partial D}{\partial \mathbf{t}_j} & \frac{\partial D}{\partial \phi_j} & \frac{\partial D}{\partial \mathbf{C}_k} \end{bmatrix}$:

$$\frac{\partial D}{\partial \mathbf{t}_j} = -R_j^{\intercal}\ ,\ \frac{\partial D}{\partial \phi_j} = dR^{\intercal}\left(\mathbf{C}_k - \mathbf{t}_j\right)\ ,\ \frac{\partial D}{\partial \mathbf{C}_k} = R_j^{\intercal} \tag{12}$$

[2]Note that Cartographer failed achieving reasonable result under current valid range and default configuration. Hence, we tuned weights and other parameters of Cartographer to keep the valid range remaining the same.

For each row in B, the Jacobian of B_i with respect to the state vector is:

$$J_{B_i} = \begin{bmatrix} \frac{\partial B_i}{\partial \mathbf{t}_j} & \frac{\partial B_i}{\partial \phi_j} & \frac{\partial B_i}{\partial \mathbf{C}_k} & \frac{\partial B_i}{\partial a_{s_1=0:N}} & \frac{\partial B_i}{\partial b_{s_2=1:N}} \end{bmatrix} \tag{13}$$

where

$$\begin{aligned}
\frac{\partial B_i}{\partial \mathbf{t}_j} &= \sum_{n=0}^{N} [a_n \frac{\partial \cos(n\theta_i^{jk})}{\partial \mathbf{t}_j} + b_n \frac{\partial \sin(n\theta_i^{jk})}{\partial \mathbf{t}_j}] - \frac{\partial r_i^{jk}}{\partial \mathbf{t}_j} \\
\frac{\partial \cos(n\theta_i^{jk})}{\partial \mathbf{t}_j} &= L \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \ , \ \frac{\partial \sin(n\theta_i^{jk})}{\partial \mathbf{t}_j} = L \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \\
\frac{\partial r_i^{jk}}{\partial \mathbf{t}_j} &= \frac{(\hat{\mathbf{p}}_i^{jk})^\intercal}{|\hat{\mathbf{p}}_i^{jk}|} \ , \ L = \frac{n \sin(n\theta_i^{jk})}{(\hat{\mathbf{p}}_i^{jk})^\intercal \hat{\mathbf{p}}_i^{jk}} (\hat{\mathbf{p}}_i^{jk})^\intercal
\end{aligned} \tag{14}$$

$$\begin{aligned}
\frac{\partial B_i}{\partial \phi_j} &= \sum_{n=0}^{N} [a_n \frac{\partial \cos(n\theta_i^{jk})}{\partial \phi_j} + b_n \frac{\partial \sin(n\theta_i^{jk})}{\partial \phi_j}] - \frac{\partial r_i^{jk}}{\partial \phi_j} \\
\frac{\partial \cos(n\theta_i^{jk})}{\partial \phi_j} &= -n \sin(n\theta_i^{jk}) \frac{({}^{\{j\}}\hat{\mathbf{p}}_i^k)^\intercal R_j^\intercal \hat{\mathbf{p}}_i^{jk}}{(\hat{\mathbf{p}}_i^{jk})^\intercal \hat{\mathbf{p}}_i^{jk}} \\
\frac{\partial \sin(n\theta_i^{jk})}{\partial \phi_j} &= n \cos(n\theta_i^{jk}) \frac{({}^{\{j\}}\hat{\mathbf{p}}_i^k)^\intercal R_j^\intercal \hat{\mathbf{p}}_i^{jk}}{(\hat{\mathbf{p}}_i^{jk})^\intercal \hat{\mathbf{p}}_i^{jk}} \\
\frac{\partial r_i^{jk}}{\partial \phi_j} &= \frac{1}{|\hat{\mathbf{p}}_i^{jk}|} ({}^{\{j\}}\hat{\mathbf{p}}_i^k)^\intercal R_j^\intercal \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \hat{\mathbf{p}}_i^{jk} \\
\frac{\partial B_i}{\partial \mathbf{C}_k} &= -\frac{\partial B_i}{\partial \mathbf{t}_j} \\
\frac{\partial B_i}{\partial a_{s_1}} &= \cos(s_1 \theta_i^{jk}) \ \ , \ \ \frac{\partial B_i}{\partial b_{s_2}} = \sin(s_2 \theta_i^{jk})
\end{aligned} \tag{15}$$

The covariance in this problem is not explicit to acquire. We adopt implicit covariance Lemma 1 [11, Lemma 1] to approximately calculate the covariance matrices (there is a typo in Lemma 1: $\Sigma_f = J_z \Sigma_z J_z^\intercal$).

The observed center ${}^{\{j\}}\mathbf{C}_k$ is fitted through circle function, which is applicable for Lemma 1. Then:

$$\begin{aligned}
f_1 &= \sum_{i=1}^{M_k^j} \left\| |{}^{\{j\}}\hat{\mathbf{p}}_i^k - {}^{\{j\}}\mathbf{C}_k|^2 - r_i^2 \right\|_{\Sigma_{f_1}^{-1}}^2 \\
J_{f_{1z}} &= 2({}^{\{j\}}\hat{\mathbf{p}}_i^k - {}^{\{j\}}\mathbf{C}_k)^\intercal \ , \ J_{f_{1x}} = -2({}^{\{j\}}\hat{\mathbf{p}}_i^k - {}^{\{j\}}\mathbf{C}_k)^\intercal \\
\Sigma_{f_1} &= J_{f_{1z}} \Sigma_z J_{f_{1z}}^\intercal \qquad , \ \Sigma_{{}^{\{j\}}\mathbf{C}_k}^{-1} = J_{f_{1x}}^\intercal \Sigma_{f_1}^{-1} J_{f_{1x}}
\end{aligned} \tag{16}$$

hence, $\Sigma_{\mathrm{c},j,k}^{-1}$ is calculated by $\Sigma_{\mathrm{c},j,k}^{-1} = J_D^\intercal \Sigma_{{}^{\{j\}}\mathbf{C}_k}^{-1} J_D$.

The calculation of $\Sigma_{\mathrm{f},j,k}$ also needs Lemma 1. For each row in B, the Jacobian of B_i w.r.t. the observed points is:

$$\begin{aligned}
J_{B_i,z} &= \sum_{n=0}^{N} [a_n \frac{\partial \cos(n\theta_i^{jk})}{\partial {}^{\{j\}}\mathbf{p}_i^k} + b_n \frac{\partial \sin(n\theta_i^{jk})}{\partial {}^{\{j\}}\mathbf{p}_i^k}] - \frac{\partial r_i^{jk}}{\partial {}^{\{j\}}\mathbf{p}_i^k} \\
\frac{\partial \cos(n\theta_i^{jk})}{\partial {}^{\{j\}}\mathbf{p}_i^k} &= \frac{\partial \cos(n\theta_i^{jk})}{\partial \mathbf{t}_j} \cdot R_j \\
\frac{\partial \sin(n\theta_i^{jk})}{\partial {}^{\{j\}}\mathbf{p}_i^k} &= \frac{\partial \sin(n\theta_i^{jk})}{\partial \mathbf{t}_j} \cdot R_j \\
\frac{\partial r_i^{jk}}{\partial {}^{\{j\}}\mathbf{p}_i^k} &= \frac{\partial r_i^{jk}}{\partial \mathbf{t}_j} \cdot R_j
\end{aligned} \tag{17}$$

where $J_{B,z} = [J_{B_1,z}; \cdots; J_{B_M,z}]$. Following Lemma 1, $\Sigma_{\mathrm{f},j,k}$ is calculated by:

$$\Sigma_{\mathrm{f},j,k} = J_{B,z} \Sigma_z J_{B,z}^\intercal \tag{18}$$

REFERENCES

[1] J. E. Guivant, F. R. Masson, and E. M. Nebot, "Simultaneous localization and map building using natural features and absolute information," *Robotics and Autonomous Systems*, vol. 40, no. 2-3, pp. 79–90, 2002.

[2] S. Kim and S.-Y. Oh, "Slam in indoor environments using omni-directional vertical and horizontal line features," *Journal of Intelligent and Robotic Systems*, vol. 51, no. 1, pp. 31–43, 2008.

[3] J. Zhao, S. Huang, L. Zhao, Y. Chen, and X. Luo, "Conic feature based simultaneous localization and mapping in open environment via 2d lidar," *IEEE Access*, vol. 7, pp. 173 703–173 718, 2019.

[4] M. Liu, S. Huang, and G. Dissanayake, "Feature based slam using laser sensor data with maximized information usage," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1811–1816.

[5] T. Gee, J. James, W. Van Der Mark, P. Delmas, and G. Gimel'farb, "Lidar guided stereo simultaneous localization and mapping (slam) for uav outdoor 3-d scene reconstruction," in *2016 International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE, 2016, pp. 1–6.

[6] D. Zhang and G. Lu, "Shape retrieval using fourier descriptors," in *In Proceedings of 2nd IEEE Pacific Rim Conference on Multimedia*. Citeseer, 2001.

[7] N. B. Puhan, N. Sudha, and A. S. Kaushalram, "Efficient segmentation technique for noisy frontal view iris images using fourier spectral density," *Signal, Image and Video Processing*, vol. 5, no. 1, pp. 105–119, 2011.

[8] S. Rakshit and D. M. Monro, "Pupil shape description using fourier series," in *2007 IEEE Workshop on Signal Processing Applications for Public Security and Forensics*. IEEE, 2007, pp. 1–4.

[9] D. Su and W. Xiang, "Characterization and regeneration of 2d general-shape particles by a fourier series-based approach," *Construction and Building Materials*, vol. 250, p. 118806, 2020.

[10] G. Jiang, L. Yin, G. Liu, W. Xi, and Y. Ou, "Fft-based scan-matching for slam applications with low-cost laser range finders," *Applied Sciences*, vol. 9, no. 1, p. 41, 2019.

[11] J. Zhao, L. Zhao, S. Huang, and Y. Wang, "2d laser slam with general features represented by implicit functions," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4329–4336, 2020.

[12] S. Huang, Z. Wang, and G. Dissanayake, "Sparse local submap joining filter for building large-scale maps," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1121–1130, 2008.

[13] S. Huang, Z. Wang, G. Dissanayake, and U. Frese, "Iterated slsjf: A sparse local submap joining algorithm with improved consistency," in *Proceedings of the 2008 Australasian Conference on Robotics and Automation, ACRA 2008*, 2008.

[14] Y. Chen, S. Huang, R. Fitch, and J. Yu, "Efficient active slam based on submap joining, graph topology and convex optimization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[15] K. Ni, D. Steedly, and F. Dellaert, "Tectonic sam: Exact, out-of-core, submap-based slam," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 1678–1685.

[16] Z. Zhan, W. Jian, Y. Li, X. Wang, and Y. Yue, "A slam map restoration algorithm based on submaps and an undirected connected graph," *arXiv preprint arXiv:2007.14592*, 2020.

[17] J. Wang, J. Song, L. Zhao, S. Huang, and R. Xiong, "A submap joining algorithm for 3d reconstruction using an rgb-d camera based on point and plane features," *Robotics and Autonomous Systems*, vol. 118, pp. 93–111, 2019.

[18] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich, "Fetch and freight: Standard platforms for service robot applications," in *Workshop on autonomous mobile service robots*, 2016.

[19] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1271–1278.