



GP-SLAM: laser-based SLAM approach based on regionalized Gaussian process map reconstruction

Bo Li¹ · Yingqiang Wang¹ · Yu Zhang² · Wenjie Zhao¹ · Jianyuan Ruan¹ · Ping Li²

Received: 10 February 2019 / Accepted: 24 January 2020 / Published online: 13 February 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Existing laser-based 2D simultaneous localization and mapping (SLAM) methods exhibit limitations with regard to either efficiency or map representation. An ideal method should estimate the map of the environment and the state of the robot quickly and accurately while providing a compact and dense map representation. In this study, we develop a new laser-based SLAM algorithm by redesigning the two core elements common to all SLAM systems, namely the state estimation and map construction. Utilizing Gaussian process (GP) regression, we propose a new type of map representation based on the regionalized GP map reconstruction algorithm. With this new map representation, both the state estimation method and the map update method can be completed with the use of concise mathematics. For small- or medium-scale scenarios, our method, consisting of only state estimation and map construction, demonstrates outstanding performance relative to traditional occupancy-grid-map-based approaches in both accuracy and especially efficiency. For large-scale scenarios, we extend our approach to a graph-based version.

Keywords Simultaneous localization and mapping · Laser-based · Gaussian process

1 Introduction

One of the key issues for mobile robots to achieve true autonomy is simultaneous localization and mapping (SLAM), which, for decades, has been a research topic of great interest in the field of autonomous systems. Although many types of SLAM methods exist, all SLAM systems contain two essential components: state estimation and map construction. State estimation, which is the “localization” in the term “SLAM,” seeks to find the rotation and translation that best align two or several frames of sensor data. This is also known as scan matching in 2D SLAM (Olson and Arbor 2009), point set registration for pattern recognition or for 3D point cloud-based SLAM (Maiseli et al. 2017), and visual odometry in visual SLAM (Scaramuzza and Fraundorfer 2011). Map construction includes representing the map in an appropriate form and fusing new measurement data into the existing map, which corresponds to the “mapping” part of the term “SLAM”. In SLAM systems, these two components integrate tightly and mutually affect each other. Once a certain form of map representation is chosen, a method of state estimation is chosen accordingly. Likewise, an incorrect state estimate in a single step may cause divergence in the map’s data fusion process, often causing catastrophic failure of the entire SLAM algo-

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s10514-020-09906-z>) contains supplementary material, which is available to authorized users.

✉ Yu Zhang
zhangyu80@zju.edu.cn

Bo Li
jameslb20@hotmail.com

Yingqiang Wang
wangyinqiang@zju.edu.cn

Wenjie Zhao
zhaowenjie8@zju.edu.cn

Jianyuan Ruan
ruanjy@zju.edu.cn

Ping Li
pli@iipc.zju.edu.cn

¹ School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, China

² State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, China

rithm. The exact combination of map construction and state estimation methods is determined based on the situation, sensors, and platform with respect to availability, cost, quality, and applicability, resulting in a wide variety of SLAM algorithms.

Existing SLAM implementations typically rely on laser range finders, RGB-D sensors, or cameras. Our research focuses on the use of laser range finders owing to their high accuracy, even in poor lighting conditions, and large field of view. Laser range finders provide a 2D or 3D range scan as a set of 10^2 to 10^6 points described by spatial coordinates that include measurement noise. Directly using these raw data to represent the map would generate a high-dimensional map state vector, which is computationally difficult to embed into a SLAM framework. Thus, in laser-based SLAM, the raw laser data are often transformed into a lower-dimensional representation from which the state estimation method is determined. Early research on the topic, during what is known as the “classical age” (Cadena et al. 2016), focused on applying SLAM to ground vehicles in indoor environments using 2D laser scanners. A standard approach from this period relied on an occupancy grid map, which represents the map as a field of random variables arranged in an evenly spaced grid. Although regarded as a mature field of research, the occupancy-grid-map-based methods (Grisetti et al. 2007) may reach their limits in both accuracy and especially efficiency when applied to mobile robots characterized by fast and unstable dynamics. Limited by their computationally expensive map-generating algorithm and scan match method, there is little room for occupancy-grid-map-based methods to improve. Other approaches represent the environment as sparse features, including landmarks (Montemerlo et al. 2002, 2003; Kaess et al. 2008) and geometric features (Choi et al. 2008; Leung et al. 2008). With well-developed feature extraction techniques, these types of approaches can be fast. However, the robotics community has come to the realization that the traditional sparse feature-based SLAM maps are insufficient for navigation autonomy and for the interaction of a robot with its environment (Teixeira and Chli 2007), e.g., semantic mapping (Kostavelis and Gasteratos 2015). Therefore, a comprehensive and compact map representation that can fully reveal the spatial structure of the environment is of interest. An ideal method should be fast, like feature-based SLAM, but should also utilize a compact and dense map representation. It should be able to estimate the pose of the robot and the map of its environment as accurately as occupancy-grid-map-based methods, but with faster computational speed.

In this research, we aim to create a new SLAM method by redesigning the two core elements of the SLAM systems, namely state estimation and map construction. To achieve such a goal, we must solve three sequential problems. (1) A new type of map representation is needed. This task should

be completed during the map construction procedure. Taking a frame of a laser measurement as the input, the map construction reforms the raw data into a low-dimensional form of map representation as the output, which should be compact, be convenient for state estimation, and can be updated recursively and efficiently. (2) An accurate state estimation method must be developed based on the new map representation. Taking the world frame of the map and a current frame of the map as the input, the state estimation procedure outputs the transformation of the current frame with respect to the world frame. (3) A corresponding applicable and efficient map updating method must be designed. When the current frame is aligned to the world frame using the transformation obtained from the state estimation procedure, the map update takes the world frame of the map and the aligned current frame of the map as input, and merges these two frames into an updated world map as output.

We propose a new type of map representation utilizing Gaussian process (GP) regression. A mathematical tool for Bayesian regression and classification borrowed from the machine learning community, GP has been proven to be powerful for processing and reforming complex data sets. Based on a regionalized GP map reconstruction algorithm, we build a compact point-set-like map while lowering the dimension of the map state vector. With this newly formed map representation, both the state estimation and the map update method can be defined using mathematical expressions. Our SLAM approach is established by iteratively carrying out the state estimation procedure and the map update procedure, and can provide sufficiently accurate estimation for medium-scale scenarios. For large-scale scenarios, we extended our approach to a graph-based version of SLAM by using the aforementioned basic version as the frontend and added a two-stage graph optimization algorithm as the backend. The main contributions of this paper are summarized as follows:

- A new type of compact and dense map representation is proposed, which fully reflects the structure of the environment.
- A new scan-to-map point set registration algorithm is presented based on the new type of map representation.
- A new SLAM scheme is developed, achieving better performance in accuracy and especially efficiency, relative to the widely used occupancy-grid-based method (Gmapping). The new SLAM scheme is flexible and can be extended to fit the framework of existing types of SLAM methods.

The remainder of this paper is organized as follows. After a discussion of related work in Sect. 2, the proposed algorithm is discussed in detail in Sect. 3, including the regionalized GP map reconstruction algorithm, the iterative GP point set registration method, the recursive-least-squares-based map

update method, and the entire SLAM procedure. In Sect. 4, we present the SLAM results of a series of experimental evaluations and compare these with the results of the Gmapping algorithm (Grisetti et al. 2007). Finally, we conclude and discuss future work in Sect. 5.

2 Related work

2.1 Gaussian process for map construction

In the recent past, Gaussian processes have been applied to environment modeling with laser range finders. Vasudevan et al. (2009) and Plagemann et al. (2008) applied GP to model large terrain with laser range measurements. Gerardo-Castro et al. (2015) proposed a 3D surface reconstruction method for laser and radar data in a realistic-field robotics scenario using Gaussian process implicit surfaces. Guizilini and Ramos (2019) proposed a regression methodology for terrain modeling based on sparse projections to a reproducing kernel Hilbert space, and variational Bayesian inference was used for the probabilistic treatment of output variables. Unlike these scenarios, in which only one surface was modeled, indoor environments contain multiple surfaces and multiple layers (e.g., two walls perpendicular to each other), which cannot be modeled by a single GP. To address this problem, we propose a regionalized GP mapping algorithm that decomposes the entire map into several sub-maps and separately models the laser measurements in each sub-map with a GP.

An additional benefit of this regionalized approach is the significant reduction in computation time. This idea is similar to local regression (Park et al. 2011), which is widely used for fast GP computation. This approach starts from the belief that a pair of distant observations is minimally correlated. As such, predictions at a test location can be performed using only a small number of neighboring points. There are two ways to implement the local regression. One is local kriging, which decomposes the entire domain into subdomains and calculates the predictions at the test location using the training points contained only within the subdomain. The local kriging approach will generate discontinuities on the boundaries of the subdomains, but can quickly calculate the predictions. A second approach is to build multiple local estimators and combine them using a weighting scheme, such as the Bayesian committee machine (Tresp 2000), local probabilistic regression (Urtasun and Darrell 2008), or a mixture of Gaussian process experts (Rasmussen and Ghahramani 2002). Although these methods avoid the discontinuity problem of local kriging, the computation time is significantly higher than local kriging. The regionalized GP mapping method proposed in this research can be classified as a local kriging approach.

O’Callaghan and Ramos (2012) presented a continuous Bayesian occupancy map representation for dynamic environments built on GP, which can be considered as a continuous version of the traditional grid occupancy map. Jadidi et al. (2014, 2018) followed a similar approach. While O’Callaghan and Ramos modeled the map with one GP, they trained two GPs for both a free region and obstacles, and then fused them to produce a unique continuous occupancy map, termed the “C-frontier map.” Kim and Kim (2013) applied local Gaussian processes with overlapping training data to build a 3D occupancy map. Use of overlapping training data was proposed to overcome the discontinuity problem on the boundaries of the local Gaussian processes. Doherty et al. (2017) proposed an inference-based 3D occupancy mapping method, and combined this method with developments in sparse kernels and data structures to map the environment. In all studies, the coordinates of the points were used as test locations and the occupancy values of the grid occupancy map were used as observations. Unlike these studies, we set the x (or y) coordinates of the points as the test locations and the y (or x) coordinates as the observations. By using the test locations combined with the output predictions from the GP as hypothetical points, we build a compact point cloud map and quantify the uncertainty of the hypothetical points with variances calculated by the GP.

In a different study, Wachinger et al. (2014) proposed an image registration method using GP for clinical practice, while our method is designed for laser measurements.

2.2 State estimation method

With the new proposed form of map representation, our state estimation approach falls into the category of laser-based point set registration, which is also known as a scan match problem. There are two types of scan match approaches: scan-to-scan matching and scan-to-map matching (Hess et al. 2016).

Scan-to-scan matching computes the relative pose between two time-adjacent measurements using two frames of range sensor data, which accumulates errors quickly. The most widely used method for this type of approach is the iterative closest point (ICP) algorithm proposed by Besl and McKay (1992), which creates closest point correspondences between two point sets and then iteratively calculates the rigid transformation by minimizing an error metric between two corresponding pairs of points.

Scan-to-map matching determines the relative pose between the current frame and the world frame, and requires a consistent map that can be updated iteratively. Compared to scan-to-scan matching, scan-to-map matching limits the accumulation of errors, but involves extra processes for map construction.

In this study, the proposed registration method is a type of scan-to-map matching approach. However, because we represent the map as a set of predictive points, our method has similarities with the ICP method. Both methods follow an iterative approach and calculate the rigid transformation between corresponding point pairs analytically. A taxonomy of ICP variants was established by Rusinkiewicz and Levoy (2001), in which six distinct stages of the algorithm were identified: sampling, matching, weighting, rejecting, error metrics, and minimization of the error metric. In our approach, the downsampled, weighted, and corresponding hypothetical point pairs can be directly derived from the GP by setting the test point locations in related positions and selecting hypothetical output points with variances below a certain threshold. In other words, the first four steps are already completed after the regionalized GP map reconstruction procedure using our approach. Thus, our registration method does not require an expensive search for point correspondence, as does ICP, which is an advantage that our approach has over ICP-based approaches. The rigid transformation can be calculated using singular value decomposition (SVD), similar to the “error minimization” step in ICP.

3 Approach

3.1 Regionalized GP map reconstruction algorithm

In our work, GP regression is applied to model and reform point data from a laser range finder for map reconstruction. There are two goals to achieve during our map reconstruction procedure. The first is to downsample the raw point data and convert them into a form that is convenient for map updating. The second is to adjust the newly formed points into the desired locations, which is a fundamental step for state estimation. Considering these two goals, our approach can be regarded as a type of interpolation. We formulate the interpolation as a GP regression by taking the measurement point data as the training points in the GP and using the test locations and predictions as the map construction result. The steps described below are directly based on the work by Rasmussen and Williams (2006). The nonlinear regression problem is usually stated as follows: given a training data set $\mathcal{D} = \{(s_i, l_i), i = 1, \dots, N\}$ of N pairs of locations $l_i \in \mathcal{R}^d$ and noisy observations $s_i \in \mathcal{R}$ obtain the predictive distribution for the realization at the test locations l_* , denoted by $s_* = f(l_*)$. In standard GP regression, we assume that the noisy observations s_i are given by $s_i = f(l_i) + \varepsilon_i, i = 1, \dots, N$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ are the independent, normally distributed noise terms. The central idea is to model every finite set of s_i from the process as jointly Gaussian distributed with a covariance function $k(\cdot, \cdot)$ over a domain $\Omega \subset \mathcal{R}^d$. Defining $s = [s_1, s_2, \dots, s_N]$,

$\mathbf{l} = [l_1, l_2, \dots, l_N]$, and $\mathbf{l}_* = [l_{1*}, l_{2*}, \dots, l_{M*}]$ where bold font indicate vector variables, the joint distribution is

$$P(f_*, s) = \mathcal{N}\left(m(s), \begin{bmatrix} k_{**} & k_{l*}^T \\ k_{l*} & \sigma^2 I + K_{ll} \end{bmatrix}\right), \quad (1)$$

where $k_{**} = k(\mathbf{l}_*, \mathbf{l}_*)$, $k_{l*} = k(\mathbf{l}, \mathbf{l}_*)^T$, K_{ll} is an $n \times n$ matrix with $K_{ll}(i, j) = k(l_i, l_j)$, which is the so-called kernel function, $m(s)$ is the mean value of s , and I is the identity matrix. The subscripts of k_{**} , k_{l*} , and K_{ll} represent two sets of locations between which the covariance is computed. In this work, we choose the exponential covariance function as the kernel function to cope with general environments. The exponential covariance function is expressed as

$$k(l_i, l_j) = \exp(-\kappa |l_i - l_j|) \quad (2)$$

where κ is a length-scale parameter. By the conditional distribution of Gaussian variables, the predictive distribution of f_* given s is

$$P(f_*|s) = \mathcal{N}\left(k_{l*}^T (\sigma^2 I + K_{ll})^{-1} s, k_{**} - k_{l*}^T (\sigma^2 I + K_{ll})^{-1} k_{l*}\right) \quad (3)$$

The predictive mean $k_{l*}^T (\sigma^2 I + K_{ll})^{-1} s$ provides the point prediction of $f(x)$ at test locations \mathbf{l}_* , whose uncertainty is measured by the covariance $k_{**} - k_{l*}^T (\sigma^2 I + K_{ll})^{-1} k_{l*}$.

In the context of the problem at hand, s and l are the Cartesian coordinates $(x_i, y_i) \in \Omega$ of each point obtained from the laser measurements, which means that we should assume $y_i = f(x_i) + \varepsilon_i$ or $x_i = f(y_i) + \varepsilon_i$. However, the functional relationship between x and y is not as explicit as that in other studies, where one surface is to be modeled. For example, a set of points arranged in a line segment parallel to the y axis cannot be represented by $y = f(x)$. More generally, a set of points tending to distribute along the x direction should be defined in the form $y = f(x)$ for interpolation to obtain better predictions and more valid predictive points. Otherwise, these points should be in the form $x = f(y)$. To deal with this issue, we propose a regionalized GP map reconstruction algorithm. In Fig. 1, we use a frame of typical laser range scan data as an example to illustrate our method. The domain Ω is decomposed into n square sub-regions $\{\Omega_i\}, i = 1, \dots, n$, where the decomposition rule is determined by the size of all the measurements and the side length of the square sub-regions. The side length of the square sub-regions, namely the grid size, is an important parameter in our method. We restrict the region of domain Ω to a rectangle that exactly covers all the points, whose boundary is set to be the integral multiple of the grid size. Then the rectangle is divided into

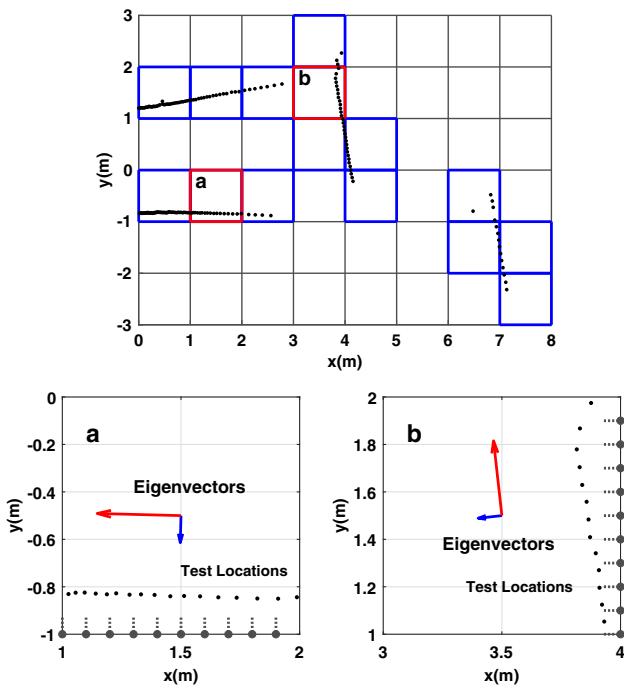


Fig. 1 Illustration of the decomposition rule and the procedure for determining the functional relationship between the x and y coordinates. Black dots indicate the laser scan data. Each sub-region boundary is indicated by a thin gray line (grid lines). The selected sub-regions are indicated by blue and red lines. The blue arrows indicate the normal directions in regions *a* and *b*. The combinations of gray dotted lines and gray dots indicate the test locations (Color figure online)

evenly spaced, disjoint and square sub-regions, which are indicated by the gray lines in Fig. 1. After the division, the sub-regions that contain no points or fewer than two points are removed. The remaining sub-regions are used for map construction, as indicated by the blue lines in Fig. 1.

Let $P_i = \{p_{ij} = [x_{ij}, y_{ij}]^T, j = 1, \dots, m_i\}$ be the subset of the point set data that belong to Ω_i . We then utilize the principal component analysis (PCA) to find a normal direction for each subset of points, where the center of each subset of points is defined as

$$g_i = \frac{1}{m_i} \sum_{j=1}^{m_i} p_{ij}. \quad (4)$$

The covariance matrix C_i of the point positions can be computed by

$$C_i = \sum_{j=1}^{m_i} (p_{ij} - g_i)(p_{ij} - g_i)^T \quad (5)$$

and the eigenvalue decomposition of matrix C_i is

$$C_i = [v_{i1} \ v_{i2}] \begin{bmatrix} \lambda_{i1} & 0 \\ 0 & \lambda_{i2} \end{bmatrix} [v_{i1} \ v_{i2}]^T \quad (6)$$

where $\lambda_{i=1,2}$ are the eigenvalues of C_i , and $v_{i=1,2}$ are the corresponding normalized eigenvectors. We choose the eigenvector $v_{i,\min}$ of C_i corresponding to the minimal eigenvalue $\lambda_{\min} = \min(\lambda_{i1}, \lambda_{i2})$ as the normal direction of the subset of points P_i . The angle of the normal direction relative to the x and y axes are calculated from

$$[angle_{ix} \ angle_{iy}]^T = \cos(I \cdot v_{i,\min}). \quad (7)$$

In our approach, the functional relationship between the x and y coordinates in each sub-region is identified based on these angles. For example, if the angle of the normal direction relative to the y -axis is smaller than that of the x -axis, as is the case in region *a*, the functional relationship is determined from $y_{ij} = f(x_{ij}) + \varepsilon_i$. The x coordinates are therefore set to be the locations of the training points l_i . In the opposite case for region *b*, the functional relationship is set as $x_{ij} = f(y_{ij}) + \varepsilon_i$ and the y coordinates are set as the training locations l_i .

To obtain an explicit functional relationship for each region, the following situations involving the training points contained in the sub-region should be avoided: a concave curve, multiple curves, or two lines perpendicular to each other. For these situations, the functional relationships cannot be identified explicitly and the GP may produce ambiguous or incorrect predictions, which are considered illegal. Avoiding the abovementioned situations can be achieved by decreasing the grid size of the sub-regions. Theoretically, each region will have an explicit functional relationship if the size of the sub-regions is sufficiently small. However, this size cannot be set to be too small, as the computational speed will decrease. We speculate that too many “for” loops spawned in the process could be the reason for this phenomenon. Because the grid size is fixed during the SLAM process, the regionalized GP map construction will inevitably generate some illegal predictive points. However, the illegal points only take up a very small portion of all predictive points (approximately 1–3%).

Once the functional relationship of each sub-region is identified, the test locations l_* are selected, as illustrated in Fig. 1. If the functional relationship is of the form $y_{ij} = f(x_{ij}) + \varepsilon_i$, the test locations are set to be a series of evenly distributed x coordinates within the sub-region. Otherwise, a series of evenly distributed y coordinates within the sub-region are selected as the test locations.

By evenly distributing the test locations, the point predictions, along with their variances, can be calculated using GP regression. To approximately maintain continuity, point predictions in each Ω_i are modeled around the mean value of s_i . We combine the predictions and their test locations as the predictive points in our approach. Each resultant prediction is actually a linear combination of the observations in the sub-region. The coefficient of each element in the lin-

Algorithm 1 Regionalized GP map reconstruction

(1)**Regionalization:** Decompose Ω into $\{\Omega_i\}, i = 1, \dots, n$. $P_i \in \Omega_i$, $P_i = \{p_{ij} = [x_{ij}, y_{ij}]^T\} (j = 1, \dots, m_i)$

(2)**Choosing:** For each $P_i, i = 1, \dots, n$, find the normal direction using (4), (5) and (6).

Compute the relative angle using (7).

if $angle_{ix} < angle_{iy}$ **then**

$$l_i = [y_{i1} \ y_{i2} \ \dots \ y_{im_i}]$$

$$s_i = [x_{i1} \ x_{i2} \ \dots \ x_{im_i}]$$

else

$$l_i = [x_{i1} \ x_{i2} \ \dots \ x_{im_i}]$$

$$s_i = [x_{i1} \ x_{i2} \ \dots \ x_{im_i}]$$

end if

(3)**GP regression:** For each $\{\Omega_i\}$, select test locations l_{i*} , then calculate s_{i*} and σ_i using (3)

if $angle_{ix} < angle_{iy}$ **then**

$$P_{i*} = [s_{i*} \ l_{i*} \ \sigma_i]^T$$

else

$$P_{i*} = [l_{i*} \ s_{i*} \ \sigma_i]^T$$

end if

Reconstruct the point map as $P_* = \bigcup_{i=1}^n P_{i*}$

(4)**Selection:** Select the predictive points in P_* with $\sigma < \sigma_{threshold}$

ear combination depends on the distance between the test location of the predictive point and the training locations. Ambiguous predictions with large variances will always be obtained at the test locations that are distant from the training locations; these predictions must be removed to obtain an accurate map. We use a similar example of typical laser range scan data to illustrate this final step of our map reconstruction method in Fig. 2. The predictive points and their variances are depicted as scatter plots in Fig. 2a. The latent and accurate point-set-like map is hidden within the entirety of the predictions. By selecting the predictive points with a variance σ below a certain threshold (Fig. 2b, c) a point-set-like map is finally obtained (Fig. 2d). The overall process is summarized in Algorithm 1.

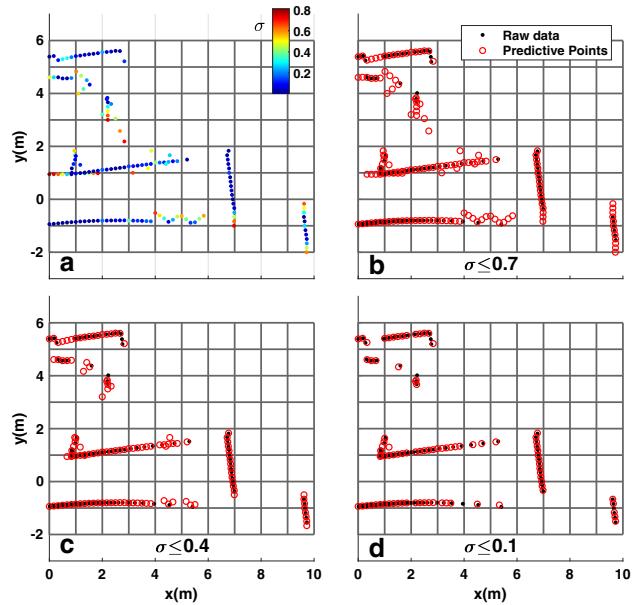


Fig. 2 Illustration of prediction selection procedure. Each sub-region boundary is indicated by the gray lines. **a** Scatter plots of all predictive points, along with their variances, obtained from regionalized GP map construction. **b-d** Predictive points for σ below a certain threshold ($\sigma \leq 0.7$, $\sigma \leq 0.4$, and $\sigma \leq 0.1$). Red circles and black dots indicate the predictive points and the laser scan data, respectively (Color figure online)

Returning to the two goals mentioned at the beginning of this sub-section, these goals can be achieved by fixing the number and values of the test locations. Fixing the number of test locations prevents the dimension of the map state in each sub-region from exceeding the designated number as the SLAM process proceeds. Fixing the values of the test locations ensures that the corresponding relationships between predictive point pairs used for state estimation are fixed, allowing the map state vectors updated during the map update step to be determined. The details on how we utilize this new map representation to implement both state estimation and map updating will be further discussed in Sects. 3.2 and 3.3.

3.2 Iterative GP point set registration

In this study, the state estimation is treated as a point set registration problem. Following an approach similar to that of ICP, we proceed with an iterative registration approach and calculate the rigid transformation between corresponding point pairs from two frames of laser measurements. Thus, our registration method can be categorized into the class of “fine registration” (Solvi et al. 2007), which requires an initial value for the transformation of the first step of the iterative process; this initial transformation can be provided by other sensors, such as inertial measurement sensors, or by applying “coarse registration” (Solvi et al. 2007). Assume

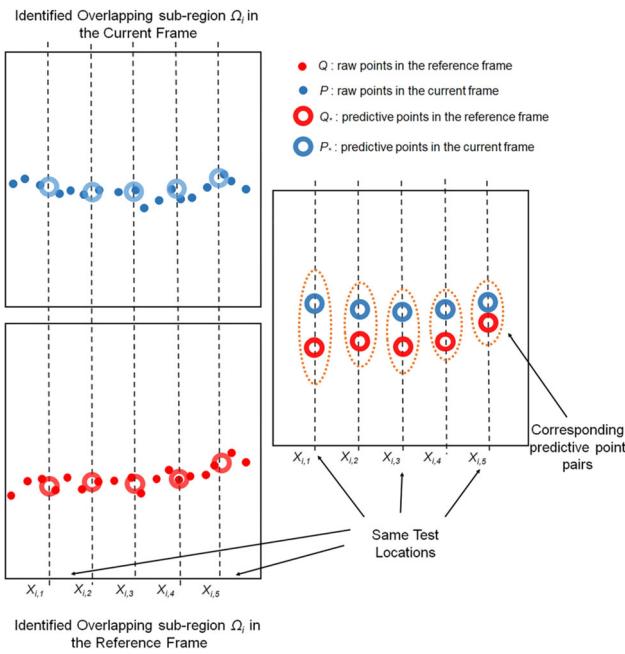


Fig. 3 Illustration of the point correspondence construction

that there are two unaligned point sets Q and P . Then, $Q = [q_1, q_2, \dots, q_m]$ is the reference frame of the point set data and $P = [p_1, p_2, \dots, p_n]$ is the current frame, where $q_i = [x_i, y_i]^T$ and $p = [x_j, y_j]^T$ (Fig. 4a). The points P in the current frame can be roughly transformed to the coordinate system of the reference frame Q when an initial transformation is known. The corresponding point pairs $M(P, Q) = \{(p_i, q_i), p_i \in P, q_i \in Q, i = 1, \dots, k\}$ are then determined based on both frames of point data. In contrast to ICP, which finds the nearest neighbor point having the minimum distance, we set up the point correspondence within the regionalized GP map construction step by using the predictive point pairs P_* and Q_* with the same test locations and variances below the same threshold.

The details of the construction of the point correspondence are illustrated in Fig. 3: The points Q in the reference frame and P in the current frame are assumed to be placed in the same frame after the initial transformation. We construct the point correspondence in each sub-region by identifying overlapped sub-regions first. Two sub-regions in their respective frame are considered to overlap only if two conditions are satisfied: they share the same center in the square region and have the same functional relationship. Once the overlapped sub-regions are identified, the test locations for both sub-regions will be considered the same according to the selection rule for test locations introduced in Sect. 3.1. The predictive points with the same test location are considered as a corresponding point pair, as shown in Fig. 3.

Redefining P_* and Q_* in x - y coordinates as $P_* = \{(p_{i*} = [x_{i,p}, y_{i,p}]^T, \sigma_{i,p})\}$ and $Q_* = \{(q_{i*} = [x_{i,q}, y_{i,q}]^T, \sigma_{i,q})\}$,

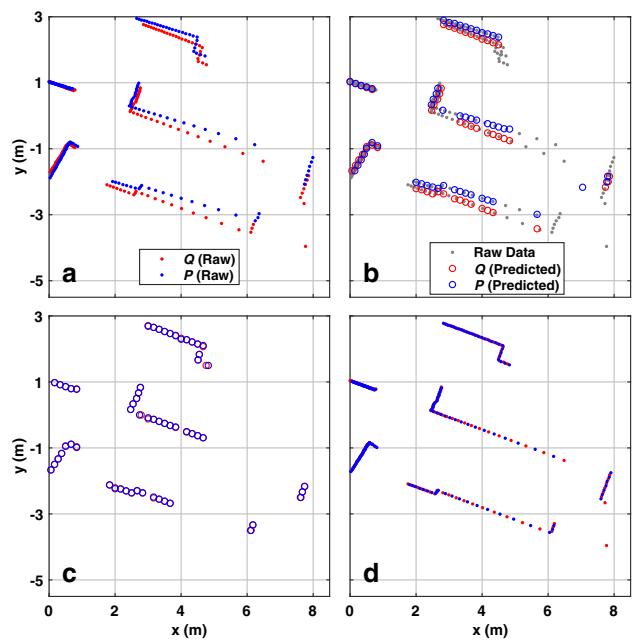


Fig. 4 Illustration of the iterative GP points set registration method using two typical frames of laser scan measurement. **a** Two unaligned points sets P (blue dots) and Q (red dots). **b** Corresponding predictive points pairs P_* (blue circles) and Q_* (red circles) and raw data P and Q (grey dots). **c** Aligned predictive points pairs P_* (blue circles) and Q_* (red circles) after 20 iterations, which almost overlap. **d** Aligned raw laser measurements from **a** (Color figure online)

the corresponding point pairs are expressed as $M(P_*, Q_*) = \{(p_{i*}, q_{i*}), p_{i*} \in P_*, q_{i*} \in Q_*, i = 1, \dots, k\}$ (Fig. 4b). The point pairs have either the same x coordinate or the same y coordinate. The rigid transformation is obtained by solving the Euclidean distance error minimization problem:

$$\arg \min_{R,t} \left(\sum_{i=1}^k \|Rp_{i*} + t - q_{i*}\|^2 \right). \quad (8)$$

This can be solved analytically using the SVD method (Arun et al. 1987). More precisely, letting g_p and g_q be the center of $\{p_{i*}\}$ and $\{q_{i*}\}$, respectively, which are calculated using (4), the covariance is

$$H = \sum_{i=1}^k (p_{i*} - g_p)(q_{i*} - g_q)^T. \quad (9)$$

Letting $U \Lambda V^T$ be the singular value decomposition of H , the optimal transformation, R and t , can be computed from

$$\begin{cases} R = V \text{diag}(1, 1, \det(VU^T)) U^T \\ t = g_q - Rg_p, \end{cases} \quad (10)$$

where R is the rigid rotation matrix and t is the translation. When a transformation is calculated based on the correspond-

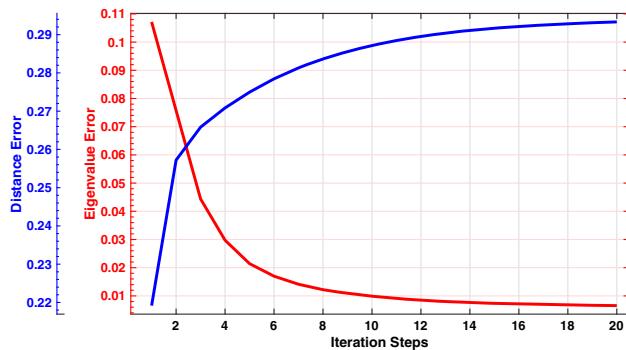


Fig. 5 Convergence rates using point-to-point error metric (blue line) and E error metric (red line) for the data in Fig. 4 (Color figure online)

ing predictive point pairs in a given iteration, it becomes the initial transformation for the next iteration, and the map construction process for the current frame of point data P is repeated to obtain a better point correspondence, from which a new transformation is computed. The iterative procedure continues until an error metric falls below a certain threshold.

Unlike the ICP-based approaches, the error metric cannot be assigned as the sum of the squared distances between corresponding points (the so-called “point-to-point” error metric) or the sum of the squared distance from each point to the plane perpendicular to the destination normal (the so-called “point-to-plane” error metric) (Chen and Medioni 1991). This is because these error metrics would not converge in decreasing fashion as the iterative registration process proceeds, which is demonstrated in Fig. 5. The reason for this phenomenon is that the hypothetical points being aligned in each iteration step change every time, because we reconstruct the point data P from the current frame using the GP at the beginning of each loop iteration. Therefore, we propose a new error metric, called “E” error metric, in which the difference between the eigenvalues of the covariance matrices of the corresponding predictive point pairs P_* and Q_* is measured. The covariance matrices C_p and C_q can still be calculated using (4) and (5), and the eigenvalue decomposition of matrices C_p and C_q is

$$C_p = V_p \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} V_p^T, C_q = V_q \begin{bmatrix} \tau_1 & 0 \\ 0 & \tau_2 \end{bmatrix} V_q^T. \quad (11)$$

We then use the absolute difference between the eigenvalues as the error metric for our approach, as given by (12), which converges as the iterative process proceeds, as demonstrated in Fig. 5.

$$\text{error}(P_*, Q_*) = \sum_{i=1}^2 |\lambda_i - \tau_i|. \quad (12)$$

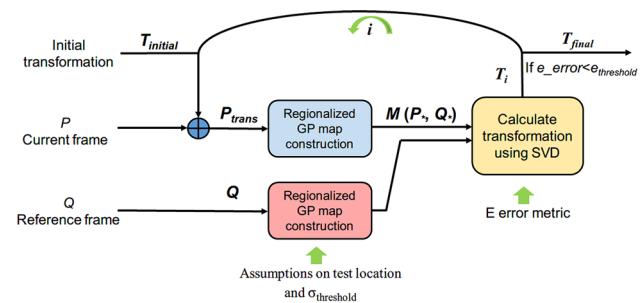


Fig. 6 Flow diagram of the iterative GP point set registration algorithm

The iterative process will be terminated if the “E” error metric is less than 0.006 or the procedure exceeds 20 iterations. Then, we obtain the registration result of the aligned predictive point pairs as shown in Fig. 4c. The transformations R and t also align the raw laser measurements with high accuracy, as shown in Fig. 4d. These results indicate that our method can also be used as an independent scan-to-scan point set registration method. Alternatively, the scan-to-scan version of our registration method can be developed by modeling the reference frame of the scan using regionalized GP and computing the relative pose between two consecutive measurements. The overall registration method is summarized in Fig. 6. Note that in our SLAM procedure, we use a “scan-to-map” matching strategy. The reference frame Q_* will be used as the world map and is not needed for the map construction step, because it is obtained from the map update step. The content in the pink box in Fig. 6 is included to maintain the integrity of the registration algorithm and for readers who intend to use it as a scan-to-scan point set registration method.

3.3 Map update

In our SLAM approach, we use the “scan-to-map” matching strategy; when the first frame of data is read, it is reconstructed into a GP predictive points map using the map construction method proposed in Sect. 3.1, which is initialized as the world map. Frame by frame, the data are aligned to the world map using the transformation calculated from the registration method proposed in Sect. 3.2, after which they are reconstructed into a GP predictive point map as the current map. In each loop, the current map should be fused with the world map to produce an updated world map. We now introduce the map update method for our SLAM approach. The test locations of the GP predictive points contained within both the world map and the current map can be divided into three classes: locations contained only within the world map, locations contained only within the current map, and overlapping locations appearing in both maps. The predictive points of the locations in the first class will remain

the same, and the predictive points from the current map with the test locations in the second class will be directly added to the world map as a newly explored area of the environment. For the overlapping regions, we use a recursive estimation method to update the GP map.

A common approach to solve this GP regression problem for sequential data processing is the recursive Gaussian process proposed by Huber (2014), which updates the entire predictive distribution recursively. However, this method involves calculating the inverse matrix of covariance matrix K_{ll} , which will decrease the computational speed. In this research, considering the efficiency of the proposed method, we choose to update each predictive mean at selected test locations instead of updating the posterior GP distribution. Similar to the occupancy grid map, which assumes independence between grid cells, we assume each predictive point in our GP point map to be independent from the others.

With this assumption, we propose a map update algorithm using a recursive least-squares estimation method (Haykin 2001) for the overlapping regions. The recursive least-squares estimation method is typically stated as follows: assuming \mathbf{A} is a constant vector and \mathbf{Z} denotes the first k observations of \mathbf{A} , the measurement equation is then

$$\mathbf{Z} = \mathbf{H}\mathbf{A} + \mathbf{V}, \quad (13)$$

where $\mathbf{Z} = [Z_1, \dots, Z_k]^T$, $\mathbf{H} = [H_1, \dots, H_k]^T$, and $\mathbf{V} = [V_1, \dots, V_k]^T$. \mathbf{H} denotes the known measurement matrices and \mathbf{V} denotes the noise vectors. The $(k+1)$ th measurement is

$$Z_{k+1} = H_{k+1}\mathbf{A} + V_{k+1}, \quad (14)$$

The weighted least-squares estimation of \mathbf{A} obtained from the first k observations is

$$\hat{\mathbf{A}}_k = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbf{Z}, \quad (15)$$

where \mathbf{W} is a weight matrix given by

$$\mathbf{W} = \begin{bmatrix} W_1 & & & \\ & W_2 & & 0 \\ 0 & \ddots & \ddots & \\ & & & W_k \end{bmatrix}. \quad (16)$$

Let P_k be the covariance matrix of the estimation calculated from $P_k = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1}$. Then, the $(k+1)$ th weighted least-squares estimation and its covariance matrix are computed from:

$$P_{k+1} = P_k - P_k H_{k+1}^T \left(W_{k+1}^{-1} + H_{k+1} P_k H_{k+1}^T \right)^{-1} H_{k+1} P_k, \quad (17)$$

$$\hat{\mathbf{A}}_{k+1} = \hat{\mathbf{A}}_k + P_{k+1} H_{k+1}^T \left(Z_{k+1} - H_{k+1} \hat{\mathbf{A}}_k \right). \quad (18)$$

We then perform a recursive least-squares estimation using (17) and (18) for each prediction in the world map.

Assume that there is an arbitrary pair of overlapping predictive points are $p_{map} = (s_{map}, l, \sigma_{map})^T$ and $p_{current} = (s_{current}, l, \sigma_{current})^T$, belonging to the world map and the current map, respectively. The variables in (13), (14), and (16) take on the values $Z_k \equiv s_{map}$, $H_k = H_{k+1} \equiv 1$, $W_k = \sigma_{map}^{-1}$, and $W_{k+1} = \sigma_{current}^{-1}$, respectively. Because we execute the map update procedure during each loop iteration in the SLAM process, the first k observations are reduced to 1, and $\hat{\mathbf{A}}_k = Z_k \equiv s_{map}$ in (15). The least-squares estimation $\hat{\mathbf{A}}_{k+1} \equiv s_{update}$ and its covariance $P_{k+1} \equiv \sigma_{update}$ are the values we use for the map update. Under this condition, the equations above take a one-dimensional form and (17) and (18) are transformed to (19) and (20), which are similar to the standard binary state Bayes filter approach (Thrun et al. 2005):

$$\sigma_{update} = \frac{\sigma_{map}\sigma_{current}}{\sigma_{map} + \sigma_{current}}, \quad (19)$$

$$s_{update} = \frac{\sigma_{map}s_{current} + \sigma_{current}s_{map}}{\sigma_{map} + \sigma_{current}}. \quad (20)$$

Finally, we update the single predictive GP point in the overlapped region as $p_{map} = (s_{update}, l, \sigma_{update})$.

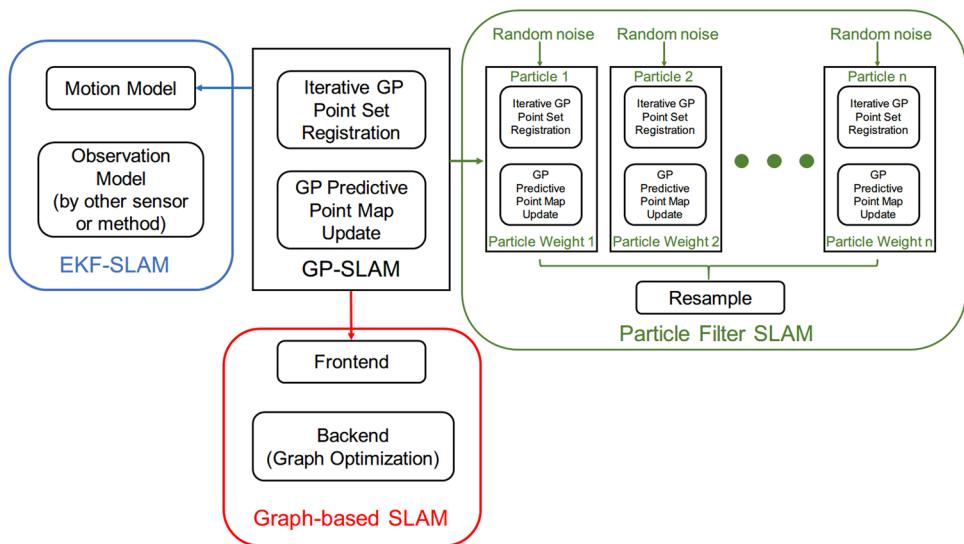
Our map update method guarantees local convergence for the predictions. The proof is as follows: assume the covariance of a single predictive point in the very first world map is σ_0 and its covariance in the i th current frame is σ_i . Since we select the predictive points with σ below the same threshold, $\sigma_i \approx \sigma_0$. Then (19) can be rewritten as $\sigma_{update} = (\frac{1}{k+1})\sigma_0$ at the k th step in the SLAM process. Therefore, as the SLAM process proceeds,

$$\lim_{k \rightarrow \infty} \left(\frac{1}{k+1} \right) = 0. \quad (21)$$

This indicates that the covariance of the predictive points in the map will converge to a very small value after a few steps. From (20), the updated value $s_{current}$ will tend toward s_{map} because the coefficient $\sigma_{current}$ is much larger. This local convergence property is an important self-rectifying mechanism in our SLAM approach.

3.4 Simultaneous localization and mapping

Our SLAM approach is established by iteratively carrying out the state estimation procedure and the map update procedure introduced in Sects. 3.2 and 3.3, respectively. Because our approach provides both localization and mapping estimations, it can serve as a complete SLAM solution. We call this

Fig. 7 Extensions of GP-SLAM

method GP-SLAM because its main idea is motivated by GP regression. The architecture of our SLAM approach can be reduced to a concise form composed of only two core parts: map construction and state estimation. This architecture is similar to that of the solution presented by Kohlbrecher et al. (2011), which operates on the occupancy grid map principle, but is simpler than those of the major existing SLAM approaches. There are three main types of SLAM methods: EKF-SLAM (Smith et al. 1990), particle filter SLAM (Montemerlo et al. 2002, 2003), and graph-based SLAM (Grisetti et al. 2010). EKF-SLAM and particle filter SLAM can be classified as recursive Bayesian estimation types, which use the extended Kalman filter and the particle filter algorithm, respectively, to solve the problem. Graph-based SLAM uses graph relationships to model the constraints on the estimated states and uses nonlinear optimization to solve the problem. A graph-based SLAM method includes two main components: the frontend and the backend. The frontend abstracts the sensor data into a model and estimates the state, and is fast and runs online. The slower backend, which runs offline, performs the graph optimization. Although our method contains only two of the core elements of SLAM, it can be flexible. Taken as the motion model and combined with an observation model obtained from other sensors or methods, our method can be extended to a version akin to EKF-SLAM. By drawing samples (adding random noise) around the estimated pose, our method can be extended to a type of particle filter SLAM. It can also be utilized as an odometry method and can serve as the frontend for graph-based SLAM. These extensions to the proposed method are summarized in Fig. 7. However, the experimental results show that in medium-scale scenarios, such extensions are unnecessary, as the basic version of our approach is sufficiently accurate.

For large-scale scenarios, our approach may fail to provide an accurate result. For these situations, we use the GP-SLAM as a frontend and extend it to an integral graph-based SLAM by adding a two-stage graph optimization procedure. The first stage of the graph optimization is similar to the work presented by Konolige et al. (2010). In this stage, we select key frame scans once after the robot travels a certain distance or rotates through a certain angle, with the poses of the robot corresponding to the key frames being set as the nodes. After selecting the key frame scans, the laser data are divided into two layers. The bottom layer is the scan stream layer composed of all raw laser data and the upper layer is the key frame layer. Once a certain number of nodes (e.g., 150) is selected, we set up the pose-pose constraints using an ICP-based method, and use the optimizer proposed by Grisetti et al. (2010) to solve the graph optimization problem. The second stage of our graph optimization involves loop closure by adding the constraints between nodes that close the loop. Because the optimized poses of the robot after the first stage are sufficiently accurate, loop closure detection can be achieved by acquiring information on the position of the robot. To accomplish this, we first calculate the distance between the nodes using the result calculated from the first stage of optimization. The node pairs with distances below a certain threshold are chosen as potential loop closure node candidates. To avoid creating excessively small loops, we do not consider the most recent nodes for a given node as candidates when calculating the distance between nodes. Once a candidate node is selected, we return to the scan stream layer and search the frames around the associated key frame of the candidate node using a scan matcher to find the scan that actually closes the loop. Once a closing loop is detected, the searched scan is added to the key frame layer and its correlated pose is added to the nodes. After loop closure

detection, the constraints between the loop closing nodes are constructed. For the other constraints, the same approach as that of the first stage of graph optimization is followed. Having obtained these newly forged pose-pose constraints in the second stage, we solve the optimization function again and obtain the optimized poses of the robot.

To accomplish the extension mentioned above, a Gaussian approximation of the GP point registration uncertainty is desirable, because in SLAM, the covariance of the estimated transformation is needed to update the estimated covariance for EKF-SLAM, to obtain a proposal distribution for particle filter SLAM, and to weigh the constraints in a pose graph for graph-based SLAM. As our registration algorithm is a point set registration, the covariance can be calculated using the methods proposed by Bengtsson and Baerveldt (2001) or Censi (2007), which are used to estimate the covariance for ICP.

4 Experiments

The approach described above has been implemented and tested using data sets gathered from real scenarios. Our approach was tested on a standard PC with a 3.4-GHz processor running in a MATLAB-2017b environment. We also implemented the Gmapping method for benchmarking in Experiments 1, 2 and 3-A. The Gmapping method has been implemented in many industrial and commercial applications and is one of the most popular standard 2D laser-based SLAM algorithms. Because Gmapping is based on the Rao-Blackwellized particle filter, to ensure a fair comparison with regard to efficiency, we use only one particle and set the noise added to the particle to be zero. The results of both methods in Experiments 1 and 2, and Experiment 3-A, are obtained without any graph optimization or loop closure. In Experiment 3-B, we show only the results of the graph-based version of our approach for large-scale scenarios. Gmapping was not performed for comparison in this part, as there is no graph optimization or loop closure in Gmapping. We consider Gmapping as orthogonal to our backend graph optimization, because these two techniques can be combined. The experiments measure the computation time, the accuracy, and the storage consumption for the two methods. The accuracies of the trajectory and the map are quantitatively and qualitatively measured in several ways. For Experiments 1 and 2, in which the ground truth of the trajectory is not available, we use the mean squared error (MSE) to evaluate the obtained maps. Following the same procedure as that utilized by Holý (2018), we consider the distance between the closest points in aligned scans as the true value of the MSE. The MSE between two frames of the scan is calculated as follows: the raw scans are both transferred into the world frame using the transformations obtained from the state estimation



Fig. 8 **a** Workshop environment. The viewpoint is near the starting point in the experiment. **b** Sensors used for collecting measurement data

procedure, and then the closest point pairs are determined by finding the points that have the minimum distance. The mean of all the distances between the closest point pairs is taken as the MSE. In our research, for one aligned scan in the current SLAM step, we did not just calculate the MSE from this scan to the most recent scan, but calculated MSEs with respect to all previous aligned scans if there was a sufficient number of overlapping parts. Two scans are considered to have sufficient overlapping parts if the number of the closest point pairs, whose minimum distance are less than 0.05 m, is larger than 30. The mean of these MSEs is used to quantitatively evaluate the accuracy of each method for one step in the SLAM process. Using the same notions as that in Sect. 3.2, we express the the closest point pairs in one pair of scans as $M(P, Q) = \{(p_i, q_i), p_i \in P, q_i \in Q, i = 1, \dots, k\}$, where $q_i = [x_i, y_i]^T$ and $p = [x_j, y_j]^T$. Assuming the number of the scans, which are identified to have sufficient overlapping parts with respect to the current scan, is k_o , the mean of the MESs M are calculated from

$$M = \left(\sum_{j=1}^{k_o} \left(\left(\sum_{i=1}^k \|p_i - q_i\|^2 \right) / k \right)_j \right) / k_o . \quad (22)$$

For Experiment 1, we also use the 2D plan of the environment as the ground truth of the map for qualitative analysis. For Experiment 3, where ground truth data are available, we compare the predicted trajectory to the correct one.

4.1 Experiment 1: workshop

The data sets used in Experiment 1 were collected by the authors with a Hokuyo UTM-30LX-EW laser range finder at an unfurnished and unused workshop in Hangzhou, China (Fig. 8). The data sets are available at https://github.com/zju-libo/Hokuyo_dataset. The environment of the workshop, with dimensions of 60 m × 30 m, is shown in Fig. 8a. The UTM-30LX-EW was equipped on a Raspberry Pi, along with a MEMS inertial measurement unit (IMU) (Fig. 8b) with a scanning frequency of 40 Hz. The Raspberry Pi was used for collecting and storing the measurement data. The system was mounted on a small cart driven manually. A total of 2669 scans were performed during the experiment. We planned to

use the IMU data as the initial transformation for our registration process. However, when performing the registration at such a high scanning rate, we found that estimating the initial transformation as a constant velocity model expressed as an identity matrix I was sufficient. The threshold $\sigma_{threshold}$ in our method was set to 0.1 and the size of each square sub-region in the map was $1.5 \text{ m} \times 1.5 \text{ m}$, encompassing 20 evenly arranged test points.

The maps generated by our approach and Gmapping are shown in Fig. 9a and b, respectively, along with the estimated trajectories of approximately 80 m in length. In Fig. 10, we show the maps and trajectories overlaid with the 2D plan of the environment. From these figures, both methods can produce a high-quality map without any significant errors.

The raw aligned laser data are shown in Fig. 9a for reference only; they do not need to be stored for actual applications of the proposed approach.

The MSE sequences of both methods are shown for each SLAM step in Fig. 11. The average MSEs of our method and Gmapping are 0.043 m and 0.073 m, respectively, indicating that ours is apparently more accurate.

Fig. 9 **a** Map and trajectory results obtained using our approach in Workshop experiment. Red dots and gray dots indicate the GP predictive points and the aligned laser measurements, respectively. The black line indicates the trajectory. **b** Map and trajectory results obtained by Gmapping. Gray, white, and black squares indicate the occupancy value in each grid cell. The red line indicates the trajectory (Color figure online)

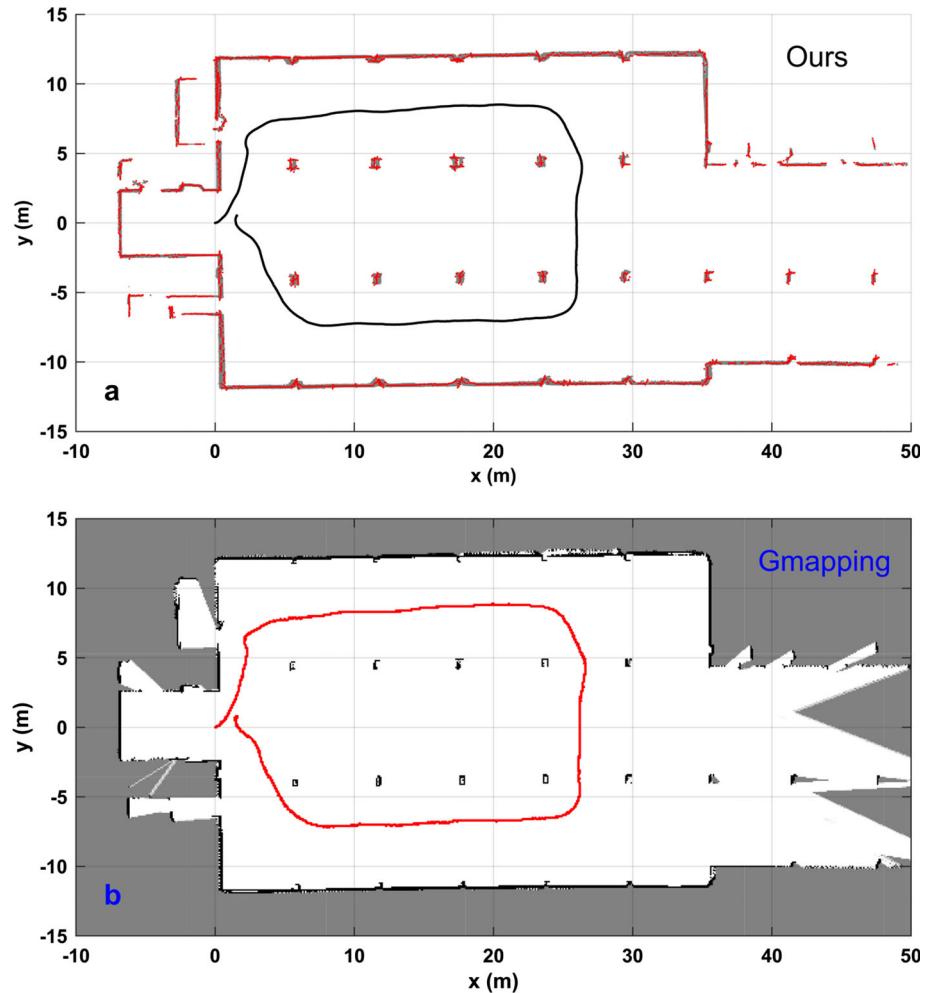
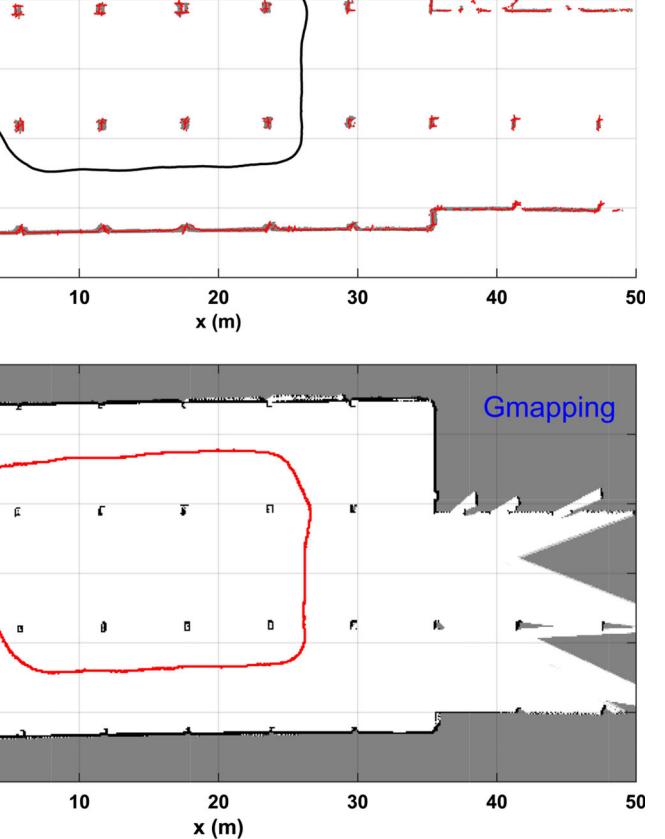


Fig. 10 Map and trajectory results obtained by both methods overlaid with map ground truth data

The computation time sequences for both methods are shown for each SLAM step in Fig. 12. As shown in the figure, our method processes each scan more quickly than Gmapping. The computation time of our approach averages 0.32 s,



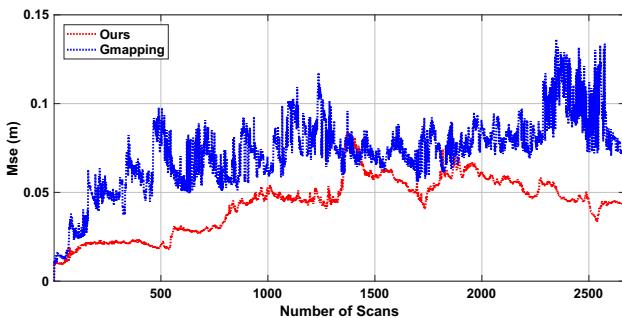


Fig. 11 MSE sequences of both methods for each SLAM step in the workshop experiment

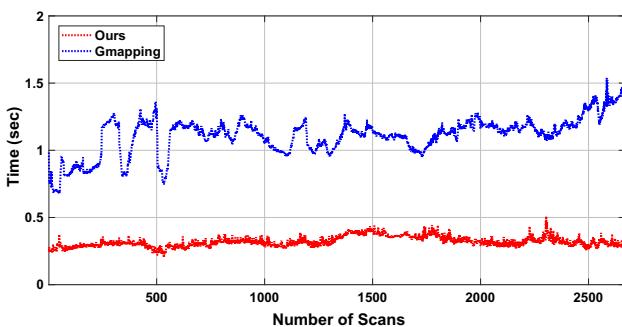


Fig. 12 Computation time sequences of both methods for each SLAM step of the workshop experiment

which is nearly three times faster than that of Gmapping (1.11 s).

The memory required for storing the map produced by our method is much lower than that required for storing the grid occupancy map. There are three types of information provided by a grid occupancy map: the occupied region, the unoccupied region, and the un-explored region. A map generated by our method provides only one, which is the occupied region containing the predictive points. To quantitatively compare the memory required for storing maps, two criteria have to be clarified. One is the resolution, and the other is the boundary of the grid occupancy map. Because the predictive points are used to present the map in our approach, the resolution of our map depends on the density of the test locations. For example, if the side length of the sub-regions is 1 m and the number of test locations contained in one sub-region is 10, the resolution of our map will be 0.1 m. In contrast, the resolution of the grid occupancy map is equal to the size of its grid cell. The memory consumption of the grid occupancy map grows linearly with the size of map, as it also stores the information of the unoccupied region. When the grid occupancy map is used to model the environment, the usual way is to pre-set the size of the map to be relatively large. In this experiment, the size of the map is set to be the same as that in Fig. 9b. The storage consumption of both methods over time in Experiment 1 is shown in Fig. 13. The resolutions

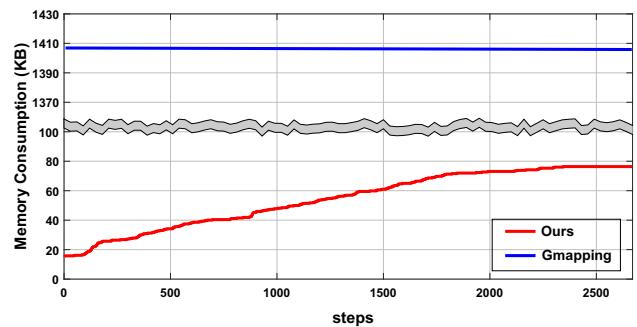


Fig. 13 Storage consumption of both methods for each SLAM step of the workshop experiment

of the maps are both set to be 0.1 m. As shown in the figure, our method consumes nearly 20 times less memory than the grid occupancy map. The maximum memory consumption cost to store the map shown in Fig. 9a is 76.3 KB using our method, while it is 1406.3 KB for the grid occupancy map. The storage consumption of Gmapping remained constant during the SLAM process because the explored region did not exceed the pre-set size of the map.

Because the state estimation procedure of our method can also be transformed into a scan-to-scan version and can be used as an independent point set registration method, as mentioned in Sect. 3.2, we provide an additional comparison between the scan-to-scan version of our registration method and ICP in this subsection. Here, 100 consecutive scans are used to demonstrate the results, as the difference between the two methods is rather small with only two scans used. All laser scans are aligned to the world frame using the transformations obtained from the two methods, and are respectively shown in Fig. 14. The pose of each measurement relative to the world frame is obtained by accumulating the former calculated transformations consecutively. The result of the scan-to-scan version of our method, with some ambiguous parts in the map, is not as good as that of the scan-to-map version, which indicates that there exist larger alignment errors. However, the scan-to-scan version of our method outperforms the ICP, whose map are almost divergent. Although our scan-to-scan registration method achieves higher accuracy, the ICP method is much faster because there is no map construction procedure involved in this method. The average computation time of ICP was 0.09 s, while the average computation time of the scan-to-scan version of our method was 0.29 s.

4.2 Experiment 2: Intel Research Lab

To test the robustness of our method, in Experiment 2, we chose a more challenging data set: the Intel Research Lab data set from the Robot Data Repository (<http://ais.informatik.uni-freiburg.de/slamevaluation/datasets.php>), which was re-

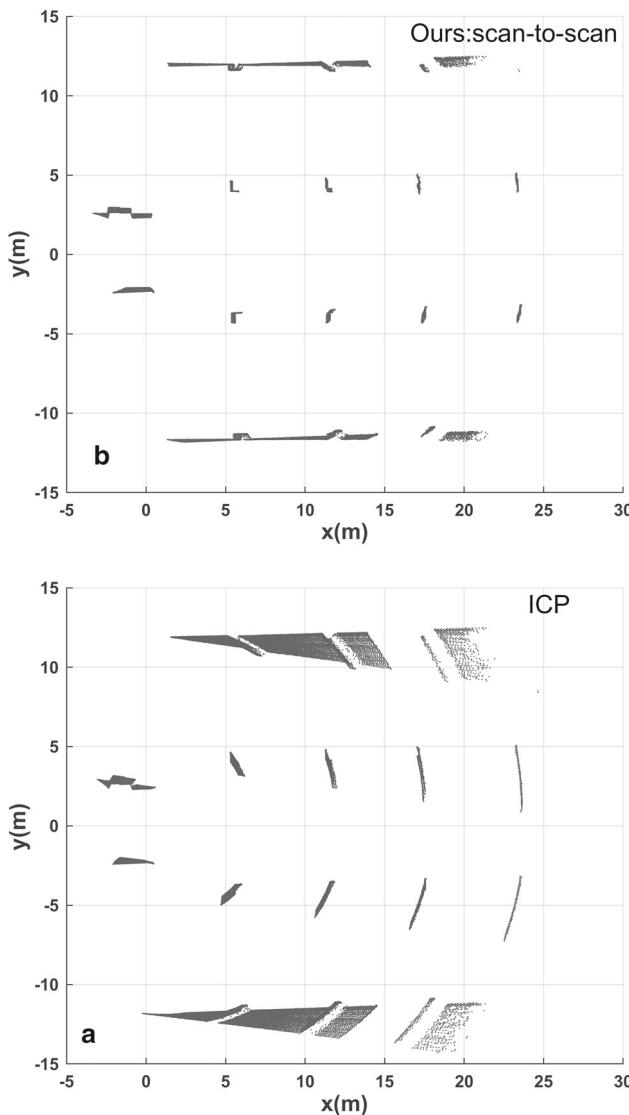


Fig. 14 Registration results using 100 consecutive scans. Gray dots indicate the raw laser data. **a** Scan-to-scan version of our method. **b** ICP

corded by Grisetti et al. (2007) at the Intel Research Lab in Seattle, USA (and was also used in their research). The Intel Research Lab data set, whose environment is $30\text{ m} \times 30\text{ m}$ in size, has a more complex structure that includes non-static elements, such as people in motion. In their work, they used 15 particles to generate a fine map, but the Gmapping method can finish the job with even one particle if the parameters are set appropriately. In this experiment, the threshold $\sigma_{threshold}$ in our method is set to 0.06 and the size of each square sub-region in the map is $0.8\text{ m} \times 0.8\text{ m}$ with 15 evenly arranged test points. The Gmapping parameters are nearly identical to those from Experiment 1. We use the first 2000 scans of the data set to complete a closed loop and use the odometry data provided in the data set to serve as the initial transformation for both methods. Our approach generates a fine map as good as that of Gmapping, as shown in Fig. 15. There are

several misaligned scans at the end of the loop. However, we still obtain an integral map, because the GP predictive points in that region have already converged by the time the laser has first passed through that region. When the misaligned points are reconstructed into the GP predictive points, they are rectified to the converged map points according to (21).

In Fig. 16, we show the MSE sequences of both methods in this experiment. These results show that, for this more complex environment, both methods can still accurately estimate the pose and the map. As in Experiment 1, our method performs more accurately, with an average MSE of 0.033 m compared to an average MSE of 0.044 m for Gmapping.

We also compare the computation time of both methods, as shown in Fig. 17. The average computation times of the Gmapping method and our approach are 0.18 s and 0.09 s, respectively. Our approach runs almost two times faster than Gmapping.

The map storage consumption of each method is compared in Fig. 18. The range of the occupancy grid map is set to the same value as that of Fig. 15b. Resolutions of both maps are 0.1 m. The maximum memory consumption of our method is 109.9 KB, while it is 957.0 KB for the Gmapping grid occupancy map.

4.3 Experiment 3: Bicocca

The experiments in this sub-section are based on the Bicocca_2009-02-27a data set (http://www.rawseeds.org/rs/capture_sessions/view/9) from the Rawseeds project. Specifically, we performed two sub-experiments. In the first sub-experiment, the basic version of our method was implemented in a local region (where ground truth data were collected) to further test its accuracy using the ground truth as the criterion. To further test the robustness of our method in large-scale scenarios, we implemented our method in a much larger loop in the second sub-experiment.

A. Local region (Experiment 3-A)

In this sub-experiment, we implemented the basic version of our method in the central region of the environment using the 3000th to 7000th scan frames. The threshold $\sigma_{threshold}$ and the size of the sub-region are set to 0.06 and 0.8 m, respectively, which are the same as those used in Experiment 2. Similar to Experiments 1 and 2, the maps and the trajectories estimated by both methods are shown in Fig. 19. As shown in this figure, the two methods still exhibit consistent results regarding both the map and the trajectory.

The errors of the trajectories are quantitatively analyzed using the ground truth data, as shown in Fig. 20. The mean error of the translations of our method is 0.042 m and is 0.070 m for Gmapping. Our method yields a better result with regard to accuracy of translation, which coincides with the results in Experiment 1 and Experiment 2 using MSE

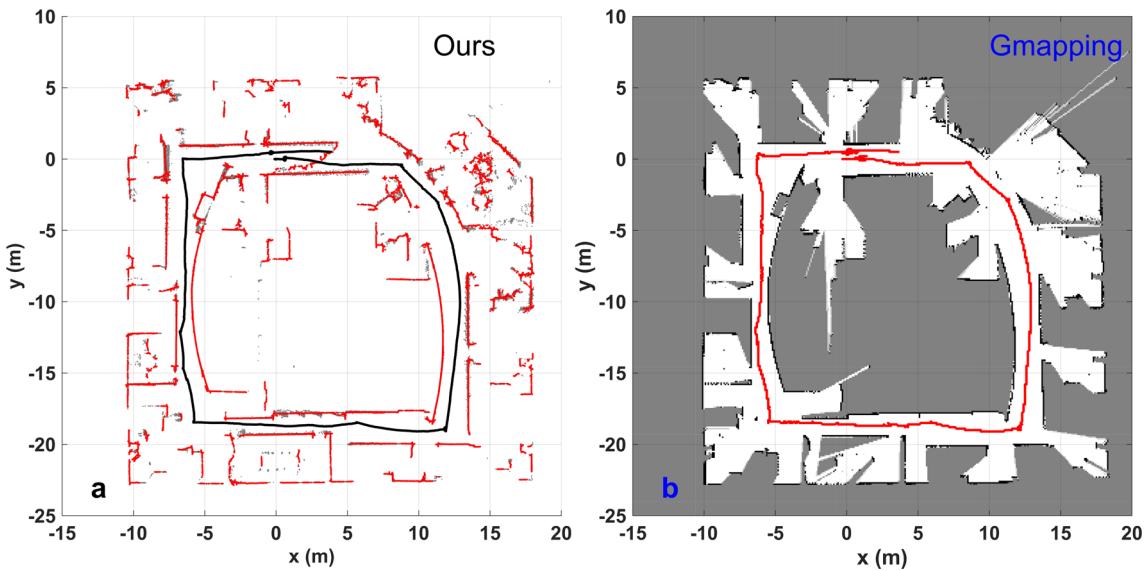


Fig. 15 **a** Map and trajectory results obtained using our approach for the Intel Research Lab experiment. Red dots and gray dots indicate the GP predictive points and the aligned laser measurements, respectively. The black line indicates the trajectory. **b** Map and trajectory results

obtained using Gmapping. Gray, white, and black squares indicate the occupancy value of each grid cell. The red line indicates the trajectory (Color figure online)

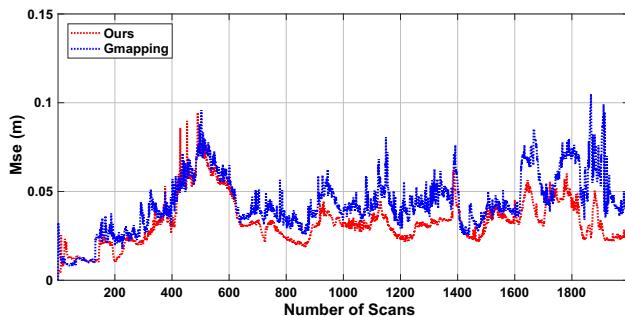


Fig. 16 MSE sequences of both methods for each SLAM step of the Intel Research Lab experiment

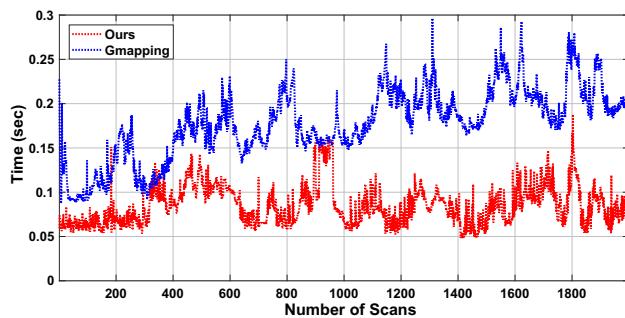


Fig. 17 Computation time sequences of both methods for each SLAM step of the Intel Research Lab experiment

as the criterion. Gmapping obtained a slightly better result for aligning rotation with a 0.018 rad mean orientation error, while ours was 0.024 rad.

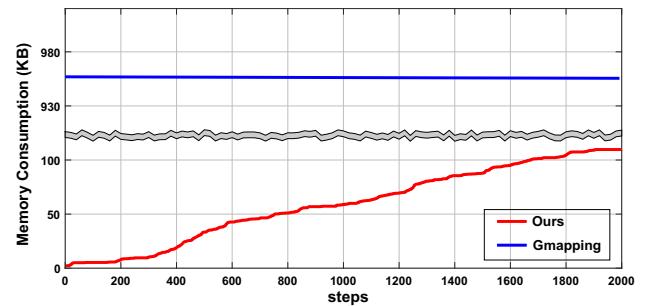


Fig. 18 Storage consumption of both methods for each SLAM step of the Intel Research Lab experiment

A comparison of the computation times of both methods is shown in Fig. 21. The average computation time of the Gmapping method and our approach are 0.18 s and 0.12 s, respectively. This demonstrates that our approach still ran faster than Gmapping.

The comparison of map storage consumption between each method is shown in Fig. 22. The range of the occupancy grid map is set to the same value as that of Fig. 19b. Resolutions of both maps are set to 0.1 m. The maximum memory consumption of our map is 67.2 KB, and is 957.0 KB for the Gmapping grid occupancy map.

B. Large loop (Experiment 3-B)

In this sub-experiment, we implement our method in a large loop of the environment. Here, the word “large” refers to the number of scans used during the SLAM process, as the error is accumulated scan by scan. There are over 170,000 scans in this data set, of which we used the first 50,000 scans

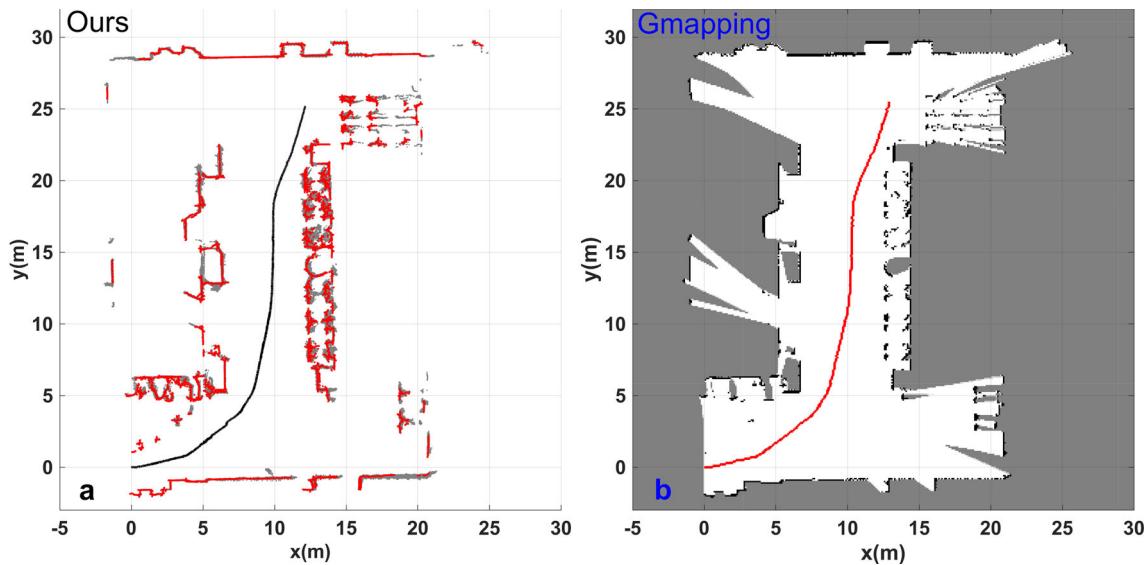


Fig. 19 **a** Map and trajectory results obtained using our approach in the Local region experiment. Red dots and gray dots indicate the GP predictive points and the aligned laser measurements, respectively. The black line indicates the trajectory. **b** Map and trajectory results obtained

using Gmapping. Gray, white, and black squares indicate the occupancy value of each grid cell. The red line indicates the trajectory (Color figure online)

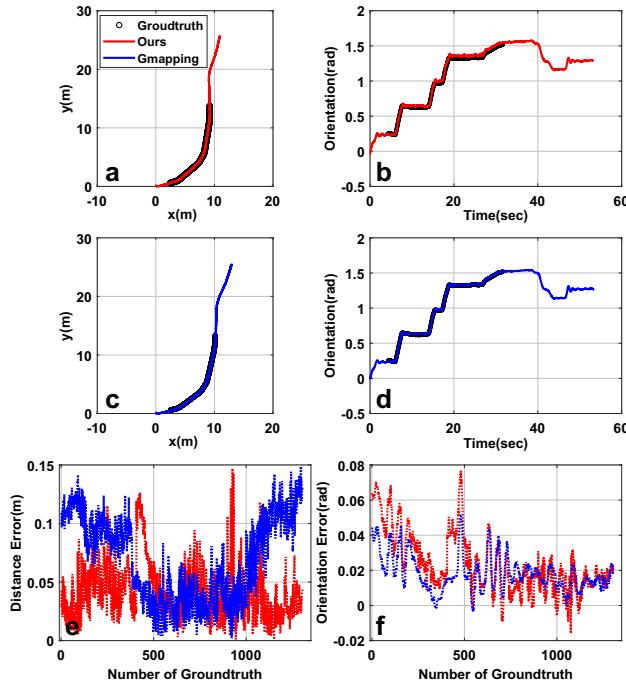


Fig. 20 **a** Estimated trajectory of our method compared to the ground truth data. **b** Estimated orientation of our method compared to the ground truth data. **c** Estimated trajectory of Gmapping compared to the ground truth data. **d** Estimated orientation of Gmapping compared to the ground truth data. **e** Sequences of position errors for both methods. **f** Sequences of orientation errors for both methods

to form a “∞-shaped” closed loop. In contrast, we used 2669, 2000, and 4000 frames of laser scan data in Experiment 1, Experiment 2, and Experiment 3-A, respectively, as previously mentioned. Facing such a long journey, both

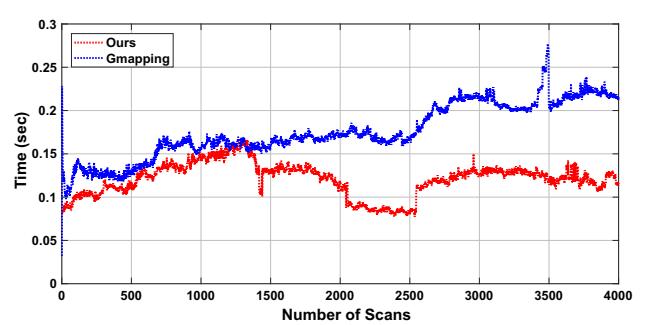


Fig. 21 Computation time sequences of both methods for each SLAM step of the local region experiment

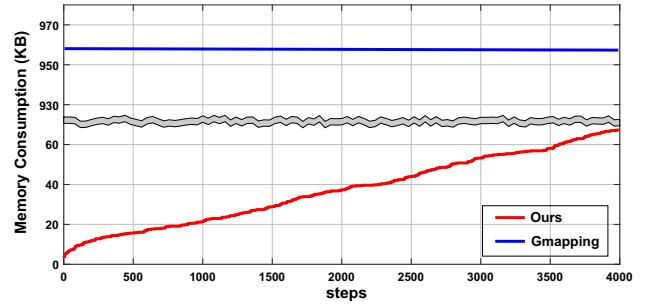


Fig. 22 Storage consumption of both methods for each SLAM step of the local region experiment

our method and the Gmapping method failed to close the loop. Therefore, we implemented the graph-based version of our method and used the aforementioned basic version as the frontend. Following the same concept as that utilized by Grisetti et al. (2010), we took the relative transformation between two nodes as the virtual measurement, the trans-

formation calculated between two key frame scans with an ICP-based method as the real measurement, and the function that computes the difference between them to determine edges. The real measurement, which is used for rectifying the virtual measurement, is supposed to be more accurate than the virtual measurement. Here, we chose our ICP-based method carefully, because our frontend, which is estimated by GP-SLAM, is so accurate that it may be better than the estimation of the ICP-based method. There is a wide spectrum of ICP-based methods available. After a series of tests, we chose the trimmed iterative closest point algorithm (Trim-ICP) presented by Chetverikov et al. (2002), as it performs more robustly in low overlap situations. Fig. 23 shows the map and the trajectory estimated from the frontend. Using only the frontend, a trajectory distance error of approximately 5 m accumulated and the map was distorted. With the help of the first graph optimization stage, the distance error was significantly reduced, although parts of the map remained inconsistent, as shown in Fig. 24. We then performed the second graph optimization stage based on the results of the first stage by incorporating a loop closure procedure. To accomplish the loop closing search discussed in Sect. 3.4, we cannot use the typical ICP methods or TrimICP for scan matching, because scans obtained even at very close distances may have different orientations. To solve this problem, there is no better option than the polar scan match algorithm (PSM) (Diosi and Keelman 2007), as it performs well with regard to aligning rotation. A loop closure detection is considered successful if the output of the error function in the PSM is below a certain threshold. After the second graph optimization stage, we finally obtained an accurate trajectory and a sufficiently fine map, as shown in Fig. 25. The rectangular region marked with gray lines in Fig. 25 is the region in which the ground truth data are measured. From Fig. 25, the trajectory estimated by our approach was so accurate that it almost completely overlapped the ground truth data.

The quantitative analysis is shown in Fig. 26. The position and orientation of the robot in the ground truth region are shown in Fig. 26a and b, respectively, along with the ground truth data. The robot passed this region twice, which split the trajectory into two sections. Because the timestamp of both the node and the ground truth data do not coincide exactly, we select the most recent ground truth data for each node to calculate the error. The position errors and the orientation errors are shown in Fig. 26c and d, respectively. The mean of the position errors is 0.078 m and the mean of the orientation errors is 0.0149 rad. The position error of the last node to appear in the ground truth area is 0.140 m, which occurs after the robot has traveled 229.815 m to reach that position. This means that our method accumulates a trajectory drift of only 0.06%.

During the graph-based SLAM process, the optimization is carried out only six times, including the second stage

of optimization. The speed of the frontend (averaged every 0.12 s) is almost the same as that of Experiment 3A. Thus, we do not perform the computation time analysis with data visualization. The average computation time of our backend was approximately 2.5 s.

4.4 Analysis

In Experiments 1, 2, and 3-A , our approach achieved better performance than Gmapping with regard to efficiency. The reason for this is as follows: Gmapping uses a brute force scan matcher “vasco” (Roy et al. 2003) for state estimation, and processes each beam of the laser measurement using the “beam model” (Thrun et al. 2005) for map construction. The time complexity of the scan match of Gmapping is $O(RMBH)$. Here, R , M , B and H represent the steps of refinement, the steps of movement, the number of beams, and the size of the hit-kernel, respectively, which are all important parameters in Gmapping. In the implementation of Gmapping in this research, R , M , and H were set to 12, 6, and 49, respectively, which were set based on the original C code of Gmapping (https://github.com/ros-perception/openslam_gmapping). B is decided by the type of sensor; it is 180 for SICK LMS and 1081 for Hokuyo UTM-30LX-EW. The time complexity of the map updating of Gmapping is $O(B(L+L))$. Here L represents the number of visited grid cells by one beam, ranging from 80 to 140. In contrast, our approach estimates the state of the robot analytically and directly utilizes the GP predictive points as the map. The time complexity of our registration method is $O(iBn^2)$. Here, i represents the number of iteration times and n represents the number of points in the sub-regions, and $n \ll B$. Because we update each predictive point separately, the time complexity of the map updating of our method is $O(M)$, where M represents the number of test locations. More importantly, the calculation of the inverse matrix in (3) and the SVD process can be accelerated considerably with numerical methods such as QR factorization.

The size of the sub-region and the $\sigma_{threshold}$ value are two important parameters among all parameters in our method, which have a significant effect on accuracy and efficiency. Although these two parameters are manually trained off-line, we find that the most influential factor relative to these two parameters is the type of sensor. When using the data collected by the UTM-30LX-EW laser range finder in Experiment 1, our method could obtain fine results with a square sub-region of approximately 1.5 m in size and a $\sigma_{threshold}$ value of approximately 0.1. In contrast, these parameters should be set to 0.8 m and 0.06, respectively, when using the data collected by a SICK LMS laser range finder. Furthermore, our method is more sensitive to the size of the sub-region in Experiment 2, the Intel Research Lab experiment. In consideration of the complex environment in

Fig. 23 Map and trajectory estimated from the frontend. Red dots and gray dots indicate the GP predictive points and the aligned laser measurements, respectively. The black line indicates the trajectory (Color figure online)

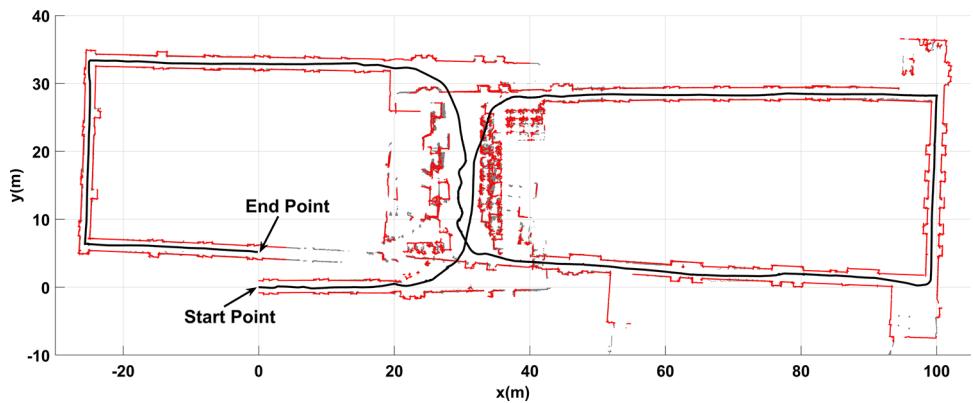


Fig. 24 Map and trajectory estimated after the first stage of optimization. Red dots and the black line indicate the GP predictive points and the trajectory, respectively (Color figure online)

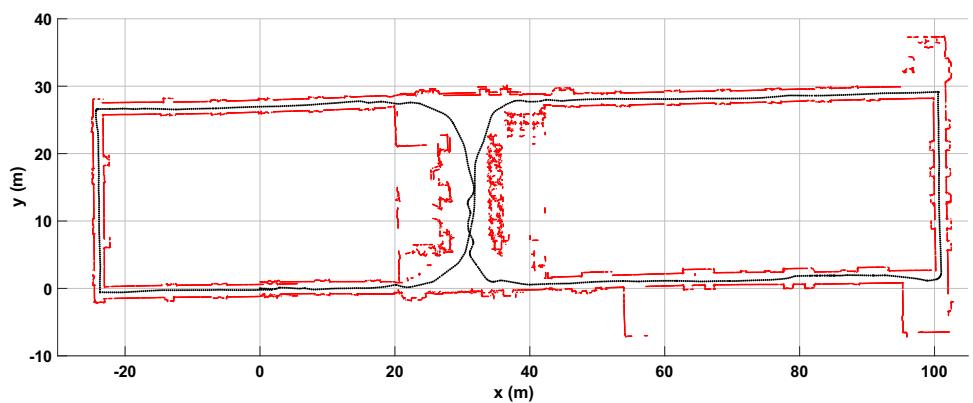
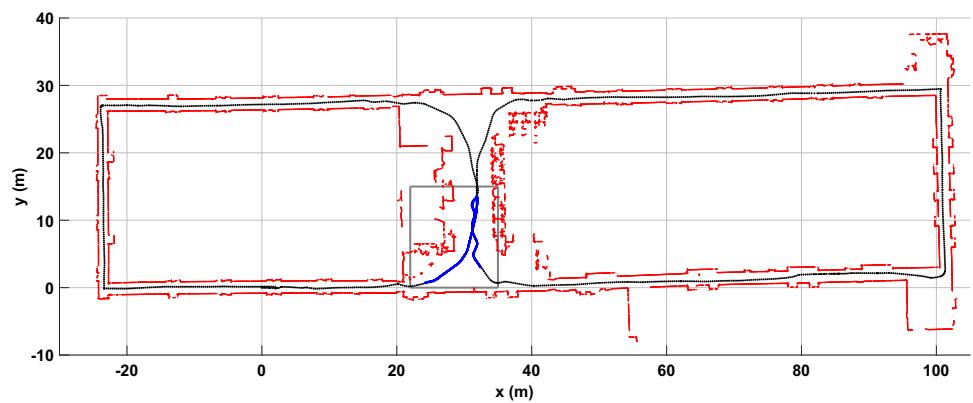


Fig. 25 Map and trajectory estimated after the second optimization stage. Red dots and the black line indicate the GP predictive points and the trajectory, respectively. Gray lines mark the region where the ground truth data are obtained. The blue lines indicate the ground truth trajectory (Color figure online)



Experiment 2, we presume that our method may not work well for some extremely unstructured environments.

The hyper-parameters σ and κ of the GPs are trained off-line and are fixed during the experiment. Their values are 0.01 and 1, respectively. We determined that our method is not sensitive to these two parameters.

Although slower than our approach, Gmapping performs more quickly in Experiment 2 than it does in Experiment 1. This occurs because of its scan match strategy mechanism and map-generating method. The Gmapping method takes the points in one scan as the “beam model.” With the Gmapping method, one must compute the pose score for

scan matching and generate the occupancy grid map beam by beam. The laser range finder used for the data set in Experiment 2 is a SICK LMS, which produces 180 range measurements in one scan, while the Hokuyo UTM-30LX-EW used in Experiment 1 produces 1081 range measurements per scan. Therefore, the lower number of range measurements in Experiment 2 leads to a faster performance.

The resolution of the map derived from Gmapping is set to 0.1 m in Experiment 1, 2, and 3-A, and can be further improved to 1 cm as demonstrated by Grisetti et al. (2007). However, generating such a high-resolution map using Gmapping is computationally intensive as mentioned

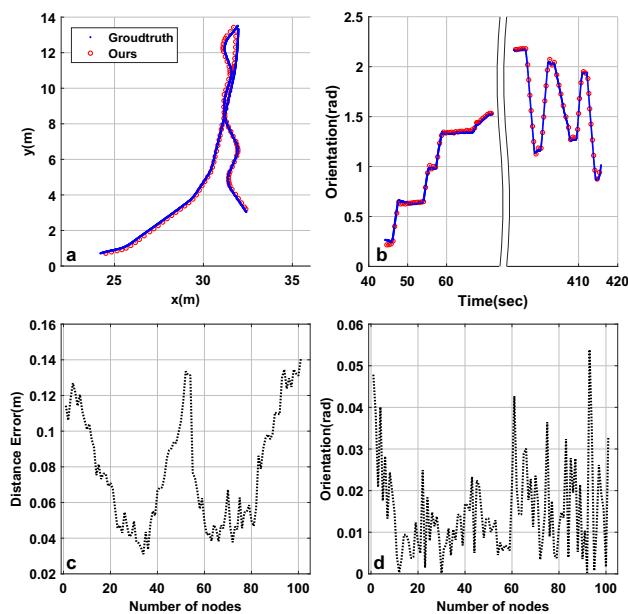


Fig. 26 **a** Estimated trajectory compared to the ground truth data. **b** Estimated orientation compared to the ground truth data. **c** Sequence of position errors. **d** Sequence of orientation errors

by Bachrach et al. (2011). In contrast, in our approach, generating a map with high resolution will only have a slight impact on the computation time. The reason is that the complexity of making prediction for M test locations is $O(M)$ using our method, while the complexity of building the GP model for N training points is $O(N^3)$ (Shen et al. 2006). Thus, the computational cost in our approach stems mainly from the calculation of the inverse matrix in (3). This is another advantage of our approach over Gmapping.

The MSE sequences in Experiments 1 and 2 demonstrate that our approach estimates the map and the trajectory more accurately than Gmapping. However, note that the MSE at this level is very low for either method, indicating that both methods yield an accurate result. Although the Gmapping method may produce more accurate results when it is extended to a multiple particle form, doing so would cause it to become inefficient.

Although the optimization calculation is carried out only a few times during the backend procedure in Experiment 3, the computational speed of the backend is much slower than that of the frontend. However, during this stage, most of the time is consumed when building the edges, which is a process that can be accelerated by multithreading.

5 Conclusion and future work

In this paper, we introduced a novel laser-based SLAM algorithm based on a new type of map representation using a regionalized GP map reconstruction algorithm. The new point-set-like map representation is compact and dense, and

can fully reflect the structural signature of the environment given from the raw laser sensor data. Based on this map representation, both the state estimation and the map updating problems can be accomplished efficiently. The proposed method has two versions. The basic version is composed of (1) the map construction, and (2) the state estimation. This version can be incorporated into available SLAM frameworks. Experimental results indicate that this version can cope with medium-scale scenarios and that it exhibits outstanding performance in terms of speed compared to the traditional widely used Gmapping method. For large-scale scenarios, we extend our approach to a graph-based version. The graph-based version of our approach accumulates an extremely small drift.

Although it uses a specific mathematical tool, namely the Gaussian process, the proposed method generalizes well because additional assumptions are not required, other than those associated with the sensor type. Furthermore, it can theoretically be generalized to a full 3D SLAM method by changing the assumption of the GP regression from an assumption of the form $y = f(x)$ to an assumption of the form $z = f(x, y)$ if 3D laser measurements are available. The functional relationship can be determined using a method similar to that of the 2D case by calculating the angles of the normal direction relative to the x , y and z axes, which will be investigated in future work.

Acknowledgements The paper is based upon work supported by the National Natural Science Foundation of China(Grant Nos. 61673341, 61573091, 2016FZA4023), the Fundamental Research Funds for the Central Universities(Grant No. 2017QN81006), National Key R&D Program of China (2016YFD0200701-3), the Project of State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT1913), and the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT1900312).

References

- Arun, K. S., Huang, T. S., & Blostein, S. D. (1987). Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5), 698–700.
- Bachrach, A., Prentice, S., He, R., & Roy, N. (2011). RANGE-robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics*, 28(5), 644–666.
- Bengtsson, O., & Baerveldt, A.J. (2001). Localization in changing environments-estimation of a covariance matrix for the IDC algorithm. In *IEEE international conference on intelligent robots and systems(CIRS)* (pp. 1931–1937).
- Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239–256.
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., et al. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6), 1309–1322.

- Censi, A. (2007). An accurate closed-form estimate of ICP's covariance. In *IEEE international conference on robotics and automation(ICRA)* (pp. 10–14).
- Chen, Y., & Medioni, G. (1991). Object modeling by registration of multiple range images. *IEEE international conference on robotics and automation(ICRA)* (pp. 9–11). CA, Sacramento.
- Chetverikov, D., Svirko, D., Stepanov, D. & Krsek, P. (2002). The trimmed iterative closest point algorithm. In *Proceedings of international conference on pattern recognition*, 3(16), 545–548, Quebec City, August 11–15.
- Choi, Y. H., Lee, T. K., & Oh, S. Y. (2008). A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot. *Autonomous Robots*, 24(13), 13–27.
- Diosi, A., & Keelman, L. (2007). Fast laser scan matching using polar coordinates. *International Journal of Robotics Research*, 26(10), 1125–1153.
- Doherty, K., Wang, J.K. & Englot, B. (2017). Bayesian generalized kernel inference for occupancy map prediction. In *IEEE international conference on robotics and automation (ICRA)*, Singapore, May 29–June 3.
- Gerardo-Castro, M. P., Peynot, T., & Ramos, F. (2015). Laser-radar data fusion with gaussian process implicit surfaces. In L. Mejias, P. Corke, & J. Roberts (Eds.), *Field and service robotics. Springer Tracts in advanced robotics* (Vol. 105, pp. 289–302). Cham: Springer.
- Grisetti, G., Kummerle, R., Stachniss, C., & Burgard, W. (2010). A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4), 31–43.
- Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid mapping with Rao–Blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1), 34–46.
- Guizilini, V., & Ramos, F. (2019). Variational Hilbert regression for terrain modeling and trajectory optimization. *International Journal of Robotics Research*. <https://doi.org/10.1177/0278364919844586>.
- Haykin, S. (2001). *Adaptive filter theory* (4th ed.). Englewood: Prentice-Hall.
- Hess, w., Kohler, D., Rapp, H. & Andor, D. (2016). Real-time loop closure in 2D LIDAR SLAM. In *IEEE international conference on robotics and automation (ICRA)*, Stockholm, Sweden, May 16–21.
- Holý, B. (2018). Registration of lines in 2D LIDAR scans via functions of angles. *Engineering Applications of Artificial Intelligence*, 67, 436–442.
- Huber, H. F. (2014). Recursive Gaussian process: On-line regression and learning. *Pattern Recognition Letters*, 45, 85–91.
- Jadidi, M. G., Miró, J. V., Valencia, R. & Andrade-Cetto, J. (2014). Exploration on continuous gaussian process frontier maps. In *IEEE international conference on robotics and automation(ICRA)*, Hongkong, China, May 31–June 7.
- Jadidi, M. G., Miró, J. V., & Dissanayake, G. (2018). Gaussian processes autonomous mapping and exploration for range-sensing mobile robots. *Autonomous Robots*, 42(2), 273–290.
- Kaess, M., Ranganathan, A., & Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6), 1365–1378.
- Kim, S., & Kim, J. (2013). Continuous occupancy maps using overlapping local Gaussian processes. *2013 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 3–7). Japan: Tokyo.
- Kohlbrecher, S., von Stryk, O., Meyer, J., & Klingauf, U. (2011). A flexible and scalable SLAM system with full 3D motion estimation. *IEEE international symposium on safety, security and rescue robots* (pp. 1–5). Kyoto, Japan: Security and Rescue Robots.
- Konolige, K., Grisetti, G., Kümmerle, R., Burgard, W., Limketkai, B. & Vincent R. (2010). Efficient sparse pose adjustment for 2D map-ping. In *2010 IEEE/RSJ international conference on intelligent robots and systems(IROS)*. Taiwan: Taipei.
- Kostavelis, I., & Gasteratos, A. (2015). Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 66, 86–103.
- Leung, C., Huang, S. & Dissanayake, G. (2008). Active SLAM in structured environments. In *IEEE international conference on robotics and automation(ICRA)*, Pasadena, CA, May 19–23.
- Maiseli, B., Gu, Y. F., & Gao, H. J. (2017). Recent developments and trends in point set registration methods. *Journal of Visual Communication and Image Representation*, 46, 95–106.
- Montemerlo, M., Thrun, S.,oller, D. & Wegbreit, B. (2002). Fast-SLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI national conference on artificial intelligence*, 593–598. Edmonton, Canada.
- Montemerlo, M., Thrun, S.,oller, D. & Wegbreit, B. (2003). Fast-SLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the international joint conference on artificial intelligence*, 1151–1156. Acapulco, Mexico, August 09–15.
- O'Callaghan, S. T., & Ramos, F. (2012). Gaussian process occupancy maps. *International Journal of Robotics Research*, 31(1), 42–62.
- Olson, E. B. & Arbor, A. (2009). Real-time correlative scan matching. In *IEEE international conference on robotics and automation(ICRA)*, Kobe, Japan, May 12–17.
- Park, C., Huang, J. Z., & Ding, Y. (2011). Domain decomposition approach for fast Gaussian process regression of large spatial data sets. *Journal of Machine Learning Research*, 12, 1697–1728.
- Plagemann, C., Kersting, K., & Burgard, W. (2008). Nonstationary Gaussian process regression using point estimates of local smoothness. In W. Daelemans, B. Goethals, & K. Morik (Eds.), *Machine learning and knowledge discovery in databases. Lecture notes in computer science* (Vol. 5212, pp. 204–219). Berlin: Springer.
- Rasmussen, C. E. & Ghahramani, Z. (2002). Infinite mixtures of Gaussian process experts. In *Advances in neural information processing systems* (Vol. 14, pp. 881–888). Cambridge: The MIT press.
- Rasmussen, C. E., & Williams, C. K. Y. (2006). *Gaussian Processes for machine learning*. Cambridge: The MIT press.
- Rusinkiewicz, S. & Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Third international conference on 3D digital imaging and modeling(3DIM)*, Quebec City, Canada.
- Roy, M., Montemerlo, M. & Thrum, S. (2003). Perspectives on standardization in mobile robot programming. In *2003 IEEE/RSJ international conference on intelligent robots and systems(IROS)*, Las Vegas, NV, October 27–31.
- Scaramuzza, D., & Fraundorfer, F. (2011). Visual Odometry: Part I, The first 30 years and fundamentals. *IEEE Robotics and Automation Magazine*, 18(4), 80–92.
- Shen, Y., Ng, A. Y., & Seeger, M. (2006). Fast Gaussian process regression using kd-trees. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems* (Vol. 18). Cambridge: The MIT press.
- Smith, R., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In I. J. Cox & G. T. Wilfong (Eds.), *Autonomous robot vehicles* (pp. 167–193). New York: Springer.
- Solvi, J., Fofi, D., & Forest, J. (2007). A review of recent range image registration method with accuracy evaluation. *Image and Vision Computing*, 25(5), 578–596.
- Teixeira, L., & Chli, M. (2007). Real-time local 3D reconstruction for aerial inspection using superpixel expansion. In *IEEE international conference on robotics and automation(ICRA)*, Singapore, May 29–June 3.
- Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. Cambridge: The MIT press.
- Tresp, V. (2000). A Bayesian committee machine. *Neural Computation*, 12(11), 2719–2741.

- Urtasun, R., & Darrell, T. (2008). Sparse probabilistic regression for activity independent human pose inference. In *IEEE conference on computer vision and pattern recognition (CVPR)*(pp. 1–8). Anchorage, AK, June 23–28.
- Vasudevan, S., Ramos, F., Nettleton, E. & Durrant-Whyte H. (2009). Gaussian process modeling of large scale terrain. In *IEEE international conference on robotics and automation (ICRA)*. Kobe, Japan, May 12–17.
- Wachinger, C., Golland, P., Reuter, M., & Wells, W. (2014). Gaussian process interpolation for uncertainty estimation in image registration. *International conference on medical image computing and computer-assisted intervention (MICCAI)*, 17(1), 267–74. Boston: MA, Sep. 14–18.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Bo Li received the B.S. degree in information and computation science from Zhejiang University, Hangzhou, China in 2008. He is now a Ph.D. candidate at school of aeronautics and astronautics, Zhejiang University, Hangzhou, China. His research interest is focus on autonomous robot navigation and environmental mapping.

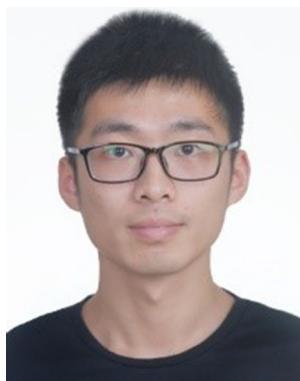
ests include navigation guidance and control of aerial robots, visual navigation and environmental mapping, intelligent control theory and application, artificial intelligence.



Wenjie Zhao received his B.E. degree and Ph.D. degree both in control science and engineering from Zhejiang University, Hangzhou, China in 2009 and 2014. He is now an associate professor at school of aeronautics and astronautics, Zhejiang University, Hangzhou, China. His research interests include unmanned aerial vehicles navigation and control, cooperative control of multiple aerial robots, and autonomous navigation.



Jianyuan Ruan received the B.E. degree in mechanical engineering from Zhejiang University, Hangzhou, China, in 2013. He is now a Master Candidate in school of aeronautics and astronautics of Zhejiang University. His research interests focus on autonomous robot navigation and environmental mapping.



Yinqiang Wang Received the B.Eng. degree in college of information engineering, Zhejiang University of Technology, Hangzhou, China, in 2017. He is now a master student in school of aeronautics and astronautics, Zhejiang University, Hangzhou, China. His research interests focus on autonomous robot navigation and environmental mapping.



Ping Li received his B.S. degree in Chemical Engineering Automation in 1982, and M.S. and Ph.D. degrees both in Industrial Automation in 1985 and 1988 respectively, all from Zhejiang University, Hangzhou, China. He is now a professor with the College of Control Science and Engineering, Zhejiang University. His research interests include unmanned aerial vehicles navigation and control, industrial process control and optimization, and intelligent transportation systems.



Yu Zhang received the B.S. degree in information engineering from Xi'an Jiaotong University, Xi'an, China in 2003, and the M.S. and Ph.D. degrees in computer science from Tsinghua University, Beijing, China in 2009. He was a post-doctor at Tsinghua University from 2009 to 2011 and a visiting scholar at Carnegie Mellon University from 2013 to 2014. He is now an associate professor in the College of Control Science and Engineering at Zhejiang University, China. His research inter-