

Parameter Optimization for Loop Closure Detection in Closed Environments

Nils Rottmann¹, Ralf Bruder¹, Honghu Xue¹, Achim Schweikard¹, Elmar Rueckert¹

Abstract—Tuning parameters is crucial for the performance of localization and mapping algorithms. In general, the tuning of the parameters requires expert knowledge and is sensitive to information about the structure of the environment. In order to design truly autonomous systems the robot has to learn the parameters automatically. Therefore, we propose a parameter optimization approach for loop closure detection in closed environments which requires neither any prior information, e.g. robot model parameters, nor expert knowledge. It relies on several path traversals along the boundary line of the closed environment. We demonstrate the performance of our method in challenging real world scenarios with limited sensing capabilities. These scenarios are exemplary for a wide range of practical applications including lawn mowers and household robots.

I. INTRODUCTION

Algorithms for simultaneous localization and mapping (SLAM) [9], [2] such as FastSLAM [24], GMapping [13], [12] or RTabMap [17], [18] require the tuning of a large number of parameters. A correct setting of these parameters is crucial for the performance of these algorithms [1]. In general, finding convenient parameters for a certain mapping task requires prior knowledge on the structure of the environment and the robot itself. However, truly autonomous systems are expected to be able to adapt themselves to any environment and thus, being able to learn the required parameters autonomously. A well-known method for such meta-parameter learning problems is classical Reinforcement Learning (RL) [31], more specifically Bayesian Optimization (BO) [29], [30]. BO is a black box optimizer that only requires a definition of a cost function. A proper definition of the cost function is critical for the success of the parameter learning procedure. For mapping algorithms, a natural choice would be to define the cost as the difference between the estimated map and the respective ground truth. However, the ground truth is not known a priori such that other cost measures have to be developed for the meta-parameter learning.

An area of increasing importance in the last decade is the field of low-cost robotics [7], [15]. Robots such as lawn mowers or vacuum cleaners are used ubiquitously in households and work exclusively in closed environments, e.g. on a lawn or in an apartment. In general, these robots have only limited sensing capabilities due to the low-cost design. Algorithms dealing with the mapping problem for

this type of robots are proposed in [26] and [6], where sonars or infrared sensors are used and linear features required. An indoor mapping approach using a wall following scheme has been presented in [34], where map rectification has been used under the assumption of straight wall segments.

Where there is an active research for SLAM approaches for autonomous vacuum cleaner, e.g. vision SLAM [16], [19], [20], autonomous lawn mowers still move randomly within the area of operation. Thereby, they use a boundary wire enclosing the working area which emits an electromagnetic signal that can be detected by the robot. Towards efficient localization and planning, a first step can be taken by mapping the enclosure. In [10], a map generation approach based the loop closure detected by returning to the home station has been introduced. Thereby, the lawn mower was driving along the boundary wire while measuring movements with the wheel odometry. However, using only a single loop closure requires to distribute the error along all estimated positions equally. Hence, detecting additional loop closures is favorable for a robust mapping approach. In [28], the authors proposed a loop closure detection approach for low-cost robots based on odometry data only. The data is collected when the robot is following the boundary of the closed environment. The performance of this approach depends highly on the correct meta-parameter setting which requires a priori knowledge about the closed environment. Hence, to enable truly autonomous behavior the robot has to learn the parameter by itself such that it can adapt to any arbitrary closed environment. Therefore, we developed a RL approach for learning meta-parameters under the assumption that the average distance traveled by the robot along a closed environment is equal to its circumference. We demonstrate the performance and robustness of our approach in different challenging simulation and real world scenarios.

The contributions of the paper are three-fold. First, we adapt and improve the method introduced in [28] by introducing relative error measurements for each loop closure using the *Iterative Closest Point* (ICP) approach [4]. Second, we insert a feasibility check in order to cope with recurrent symmetric structures and third, we introduce a RL scheme for learning the meta-parameters to enable true autonomous behavior. For our approach, we require that the robot is able to travel several times along the boundary line of the closed environment, e.g. by using a perimeter wire.

We start by summarizing and adapting the mapping method

¹Institute for Robotics and Cognitive Systems, University of Luebeck, Ratzeburger Allee 160, 23562 Luebeck, Germany
{rothmann, bruder, schweikard, rueckert}@rob.uni-luebeck.de

from [28], Section II. In Section III, we derive the RL procedure for meta-parameter learning. The procedure is divided into two stages, parameter learning for loop closure detection and pose graph optimization. We evaluate our approach in simulations and on real data in Section IV and in Section V we conclude.

II. MAPPING PROCEDURE

As the robot follows the boundary line of the closed environment, e.g. by means of the electromagnetic wire signal, a path based on the robot odometry data can be recorded. This path can then be transferred into the well-known pose graph representation [11]. Loop closures can be identified by comparing the neighborhoods of the pose graph vertices with each other, e.g. due to shape comparison. Based on the identified loop closing constraints the pose graph can be optimized by reducing the sum of weighted residual errors. Both, finding good loop closing constraints and optimizing the pose graph are strongly dependent on the correct parameter tuning. In the following, we shortly recapture pose graph representation as well as the general idea for detecting loop closing constraints. In Table I, we listed our notations for the different variables used throughout this paper.

A. Pose Graph Representation

Let $\mathbf{p} = \{\mathbf{p}_0, \dots, \mathbf{p}_N\}$ be a set of $N+1$ poses representing the position and orientation of a mobile robot in a two dimensional space, hence $\mathbf{p}_i = [\mathbf{x}_i^\top, \varphi_i]^\top$. Here, $\mathbf{x}_i \in \mathbb{R}^2$ is the cartesian position of the robot and $\varphi_i \in [-\pi, \pi]$ the corresponding orientation as an euler angle with the integer $i = 0:N$. The relative measurement between two poses i and j is then given as

$$\xi_{ij} = \begin{bmatrix} \mathbf{R}_i^\top (\mathbf{x}_j - \mathbf{x}_i) \\ \varphi_j - \varphi_i \end{bmatrix} = \mathbf{p}_j \ominus \mathbf{p}_i, \quad (1)$$

where $\mathbf{R}_i = \mathbf{R}_i(\varphi_i)$ is a planar rotation matrix and \ominus the pose compounding operator introduced by [21]. The pose graph is then a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $N+1$ vertices, representing the poses, and $N+M$ edges, representing the relative pose measurements. In our case, these pose measurements are composed of N odometric constraints and M loop closing constraints. In Figure 1, an example of a pose graph with four odometric and one loop closing

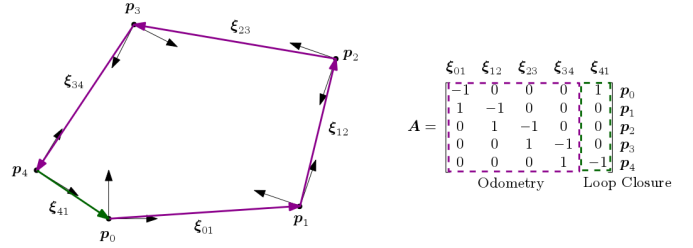


Fig. 1: Pose graph with five vertices connected with five edges. Four of the edges are odometric constraints and one is a loop closing constraint. On the right, the incidence matrix is shown divided into the parts containing the odometric constraints and the loop closing constraints.

constraint is shown. The connection between the vertices by the edges can be compactly written using an incident matrix \mathbf{A} , which is exemplarily shown on the right in Figure 1.

To account for noise in the relative pose measurements, we include zero mean Gaussian noise $\epsilon_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{ij})$, where

$$\hat{\xi}_{ij} = \xi_{ij} + \epsilon_{ij}, \quad (2)$$

denotes the with noise corrupted relative pose measurements. The overall optimization problem is then to minimize the sum of weighted residual errors $\mathbf{r}_{ij}(\mathbf{p})$ with respect to the pose estimates \mathbf{p} ,

$$\min_{\mathbf{p}} \sum_{(i,j) \in \mathcal{E}} \|\mathbf{r}_{ij}(\mathbf{p})\|_{\mathbf{P}_{ij}}^2, \quad (3)$$

where

$$\|\mathbf{r}_{ij}(\mathbf{p})\|_{\mathbf{P}_{ij}}^2 = [(\mathbf{p}_j \ominus \mathbf{p}_i) - \hat{\xi}_{ij}]^\top \mathbf{P}_{ij}^{-1} [(\mathbf{p}_j \ominus \mathbf{p}_i) - \hat{\xi}_{ij}]. \quad (4)$$

Here, \mathbf{P}_{ij} is the covariance matrix corresponding to the noise of the relative measurements $\hat{\xi}_{ij}$.

B. Loop Closure Detection

Based on the pose graph, loop closing constraints are detected by comparing the shape of the neighborhood regions of each vertex with another. Therefore, a piecewise linear function

$$\theta(x) = \phi_i \quad \text{for } l_{i-1} \leq x < l_i, \quad i = 0, 1, \dots, N. \quad (5)$$

representing the shape of the pose graph is constructed by accumulating the orientation and distance differences between the poses

$$\begin{aligned} \phi_i &= \phi_{i-1} + \Delta\phi_i \\ l_i &= l_{i-1} + \|\mathbf{v}_i\|. \end{aligned} \quad (6)$$

Here, $\mathbf{v}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$ and $\Delta\phi_i = \varphi_i - \varphi_{i-1}$ starting by $\phi_0 = \varphi_0$ and $l_0 = 0$. Figure 2 shows such a constructed piecewise orientation function. By defining the neighborhood of a vertex i as $[l_i - L_{\text{NH}}, l_i + L_{\text{NH}}]$, a comparison error between two vertices i and j is given as

$$\mathbf{C}_{ij} = \int_{-L_{\text{NH}}}^{+L_{\text{NH}}} [\theta(l_i + x) - \phi_i] - [\theta(l_j + x) - \phi_j] dx. \quad (7)$$

TABLE I: Variable definitions used throughout this paper.

\mathbf{p}	\mathbb{R}^3	poses
\mathbf{x}	\mathbb{R}^2	positions in meters
φ	\mathbb{R}	orientations in rad
\mathbf{R}	$\mathbb{R}^{2 \times 2}$	two-dimensional rotation matrix
ξ	\mathbb{R}^3	relative measurements
\mathbf{P}	$\mathbb{R}^{3 \times 3}$	cov. matrix to the noise of the rel. measurements
N	\mathbb{N}	number of odometric constraints
M	\mathbb{N}	number of loop closing constraints
L_{NH}	\mathbb{R}	neighborhood length in meters
c_{min}	\mathbb{R}	minimum comparison error
γ_1, γ_2	\mathbb{R}	pose graph optimization parameters
U	\mathbb{R}	circumference of the closed environment in meters
u	\mathbb{R}	path distance between loop closing pairs in meters
$\Delta\varphi$	\mathbb{R}	difference in orientation in rad
φ_{cycle}	\mathbb{R}	feasibility check parameter

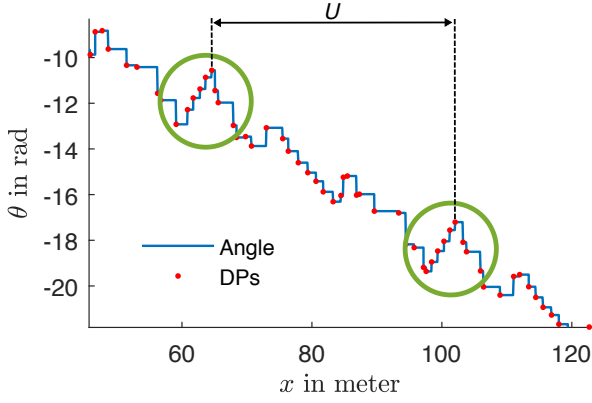


Fig. 2: Example for the piecewise linear orientation function $\theta(x)$. The green circled regions show similar path segments. The vertices or dominant points (DPs) of the pose graph are pictured as red dots. The estimated circumference U for the closed environment is exemplarily depicted for a possible loop closing pair.

We rewrite Equation (7) as a sum over m linearly distributed evaluation points

$$C_{ij} = \frac{1}{m} \sum_{k=1}^m [\theta(l_i + x_k) - \phi_i] - [\theta(l_j + x_k) - \phi_j] \quad (8)$$

with $x_1 = -L_{\text{NH}}$, $x_m = +L_{\text{NH}}$. In Figure 3, a resulting error matrix between all vertices is graphically illustrated. A loop closing pair $SP_k = \{\mathbf{p}_i, \mathbf{p}_j\}$ for $i \neq j$ is defined as a local minimum of C_{ij} for which holds $C_{ij, \min} < c_{\min}$. A local minimum represents thereby the best possible loop closure in a certain region of the error matrix and the threshold c_{\min} ensures that not every local minimum is selected as loop closing pair, but only sufficient accurate ones. Thus, the parameters L_{NH} and c_{\min} are crucial for efficiently finding convenient loop closing pairs and will be learned through Bayesian Optimization. This process is discussed in Section III-A.

After detecting a loop closure between the vertices i and j of the pose graph, the loop closing constraint as a relative measurement $\hat{\xi}_{ij}$ has to be added. Therefore, the neighborhood regions of both poses i and j are discretized as distinct points, represented by the sets $X_i = \{\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,K}\}$ and $X_j = \{\mathbf{x}_{j,1}, \dots, \mathbf{x}_{j,K}\}$, and transformed such that both poses i and j are equal with $\hat{\mathbf{p}}_i = \hat{\mathbf{p}}_j = [0, 0, 0]^\top$. By using an adapted ICP approach [3], which minimizes the distance error

$$\min_{\mathbf{R}_\beta, \mathbf{t}} E_{\text{dist}}(\mathbf{R}_\beta, \mathbf{t}) = \min_{\mathbf{R}_\beta, \mathbf{t}} \sum_{k=1}^K \|\mathbf{R}_\beta \mathbf{x}_{i,k} + \mathbf{t} - \mathbf{x}_{i,k}^*\| \quad (9)$$

with $\mathbf{x}_{i,k}^*$ being the point of X_j closest to $\mathbf{x}_{i,k}$, a two dimensional rotation \mathbf{R}_β with β being the rotation angle and a translation vector $\mathbf{t} = [t_x, t_y]^\top$ can be calculated. The loop closing constraint can then be derived using Equation (1) by transforming $\hat{\mathbf{p}}_j$ given the rotation and translation which leads to

$$\hat{\xi}_{ij} = [t_x \quad t_y \quad \beta]^\top. \quad (10)$$

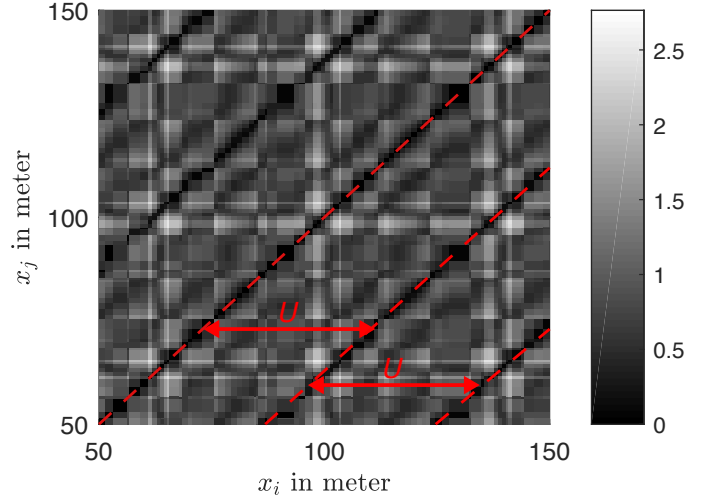


Fig. 3: Comparison error of the shapes of the neighborhood between the vertices of the pose graph. For better reading we plotted the error in the form $\log(1 - C_{ij})$ and only a section of the matrix. The variables x_i and x_j are representing the position l of the vertices i and j in meter along the pose graph. The estimated circumference U for the closed environment can be read directly from the graphic.

The corresponding covariance matrix can be calculated using the correlation error C_{ij} and tuneable parameters γ_1 and γ_2

$$\mathbf{P}_{\text{ic},ij} = \text{diag}([\gamma_1 \quad \gamma_1 \quad \gamma_2]) C_{ij}. \quad (11)$$

The parameters γ_1 and γ_2 are constant for all loop closing constraints and will be learned through Bayesian Optimization. This process is discussed in Section III-B. For the odometric constraints we generate the covariance matrices as

$$\mathbf{P}_{\text{odometric},ij} = \text{diag} \left(\begin{bmatrix} \cos(\varphi_i)(\alpha_3 \delta_T + \alpha_4 \delta_R) \\ \sin(\varphi_i)(\alpha_3 \delta_T + \alpha_4 \delta_R) \\ \alpha_1 \delta_R + \alpha_2 \delta_T \end{bmatrix} \right) \quad (12)$$

on the basis of the odometry model presented in [32] and under the assumption that only one translation δ_T and one rotation δ_R occur. The parameters $\alpha_1, \dots, \alpha_4$ can be learned. Here, we assume these parameters are given due to a known odometry model of the underlying differential drive system.

C. Recurrent Symmetric Structures

A problem for the approach introduced above are recurrent symmetric structures. Such structures are present in many real world scenarios, and hence an autonomous robot needs to be able to cope with them. Therefore, we introduce a feasibility check

$$|\pi - \text{mod}(\Delta\varphi_{ij}, 2\pi)| > \varphi_{\text{cycle}} \quad (13)$$

for every loop closing pair $\{\mathbf{p}_i, \mathbf{p}_j\}$ with the respective difference in orientation $\Delta\varphi_{ij} = \varphi_j - \varphi_i$. Here, $\text{mod}(a, b)$ is the modulo function which gives back the remainder of the Euclidean division of a by b . Only loop closing pairs which pass the check of Equation (13) are considered for pose graph optimization. The feasibility check is based on the assumption, that, on average, the orientation error of the

odometric measurements will sum up to zero. However, the orientation difference can largely differ and thus the meta-parameter $\varphi_{\text{cycle}} \in [0, \pi]$ has to be selected accordingly.

III. META-PARAMETER LEARNING

To learn the unknown meta-parameters for the above mapping algorithm, we define an optimization problem with the objective

$$\min_{\theta} c(\theta) \quad (14)$$

as a general cost function. This cost is then minimized through episodic BO [29] with expected improvement [23]. To optimize both terms, the loop closing parameters L_{NH} , c_{min} , φ_{cycle} and the pose graph optimization parameters γ_1 , γ_2 we define a two-stage optimization process. First, we optimize the loop closing parameters L_{NH} , c_{min} , φ_{cycle} which gives us as a by-product an estimate of the circumference of the closed environment U . Based on the estimated circumference U we can define a cost function for optimizing the pose graph parameters γ_1 , γ_2 . Hence, a joint optimization of all parameters is not suitable. In the following, we derive the two cost functions required for the optimization process.

A. Stage 1 – Optimization of Loop Closing Parameters

We assume that the odometric error between two poses i and j is on average zero. This is a quite strong assumption, however, a non-zero mean value will be inherent in the generated map and thus compensated when navigating with the same robot odometry. To model this error, we use a Gaussian Distribution $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$ with the covariance matrix \mathbf{P} . Let u then denote the distance along the pose graph between a loop closing pair i, j

$$u = \sum_{k=i}^{j-1} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|. \quad (15)$$

Given the assumption from above, the path distances for all loop closing pairs $\mathbf{u} = [u_1, u_2, \dots, u_M]$, identified by cycling around a closed environment, are, on average, multiples of the circumference nU . Here, $n \in \mathbb{N}^+$ is a positive integer, representing the number of cycles before the loop closure detection. Hence, if all loop closures are detected properly, a histogram of the path distances \mathbf{u} has only equally distributed peaks at positions nU . The right panels of Figure 4 show such histograms for ill-detected loop closures (top) and well-detected loop closures (bottom). To transform this idea into a cost function, we can learn a Gaussian Mixture Model (GMM) [27] with the probability distribution

$$p(\mathbf{u}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{u} | \mu_k, \Sigma_k) \quad (16)$$

from observed path distances \mathbf{u} . Here, K is the number of mixture components and π_k , μ_k , Σ_k the mixture weight, the mean and the variance of the k -th component respectively. As part of the cost function we use the negative log likelihood of

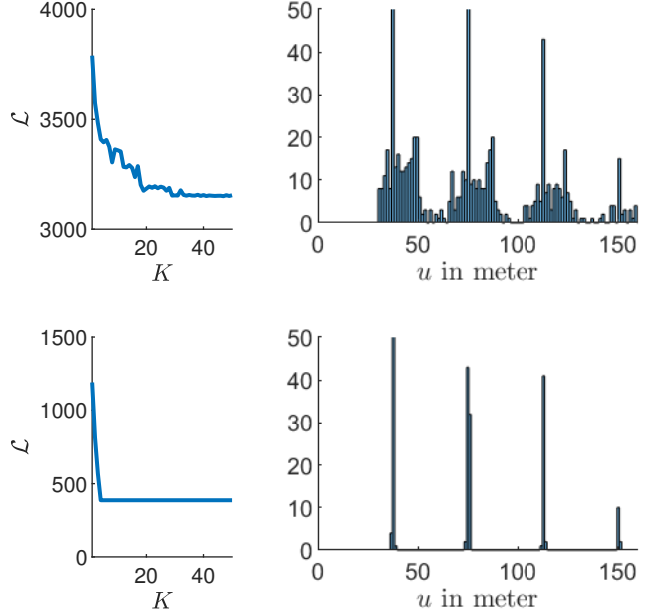


Fig. 4: For the top panels, the mapping parameters have been ill-chosen. The upper left panel shows the negative log likelihood history and the upper right panel the histogram for the path distances of the loop closing pairs. In the bottom panels the mapping parameters have been well-chosen. Again, in the left panel the negative log likelihood history is shown and in the right the histogram.

the GMM

$$-\mathcal{L} = -\ln p(\mathbf{u} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\sum_{i=1}^M \ln \left[\sum_{k=1}^K \pi_k \mathcal{N}(u_i | \mu_k, \Sigma_k) \right] \quad (17)$$

over the data set $\mathbf{u} = [u_1, u_2, \dots, u_M]$. The log likelihood decreases if the dataset \mathbf{u} meets the above assumption of evenly distributed peaks at positions nU . A common strategy for training GMMs is to iteratively increasing K until the log likelihood does not improve further. In the left column in Figure 4, the evolution of the negative log likelihood with respect to the number of components of the GMM is shown. For fitting the GMM the iterative Expectation-Maximization (EM) algorithm is used [8], [22]. The EM algorithm starts with a randomly selected model and then alternately optimizes the allocation of the data \mathbf{u} , i.e. the weighting π_k , to the individual parts of the model and the parameters of the model μ_k and Σ_k . If there is no significant improvement, the procedure is terminated.

We define the cost function for the loop closure detection as

$$\min_{\theta} c(\theta) = \min_{\theta} (-\mathcal{L} - \log(M)), \quad (18)$$

with the unknown parameters $\theta = [L_{\text{NH}}, c_{\text{min}}]$, the length of the neighborhood and the minimum comparison error, and M being the number of loop closures found. The cost function represents a trade-off between the number of loop closures,

where more reliable loop closures result in a better pose graph optimization, and a restrictive choice of loop closures to avoid false detection. Based on the best GMM fit, the circumference of the closed environment U can be estimated.

To also learn the meta-parameter φ_{cycle} for the feasibility check for recurrent symmetric structures, we can calculate the negative log likelihood for orientation differences of the loop closing pairs $\Delta\varphi_{ij}$, similar to Equation (17), under the assumption that all $\Delta\varphi_{ij}$ for accurate loop closures are close to $n2\pi$ with $n \in \mathbb{N}^+$. Equation (18) then turns into

$$\min_{\theta} c(\theta) = \min_{\theta} (-\mathcal{L} - \mathcal{L}_{\varphi} - \log(M)), \quad (19)$$

with θ being now $\theta = [L_{\text{NH}}, c_{\text{min}}, \varphi_{\text{cycle}}]$. The additional cost term $-\mathcal{L}_{\varphi}$ represents the negative log likelihood of the GMM from Equation (17) over the data set $\Delta\varphi = \{\Delta\varphi_1, \Delta\varphi_2, \dots, \Delta\varphi_M\}$, thus

$$-\mathcal{L}_{\varphi} = -\ln p(\Delta\varphi | \pi, \mu, \Sigma). \quad (20)$$

B. Stage 2 – Optimization of Pose Graph Parameters

Based on our assumption of a zero mean odometric error we can assume the estimated circumference U from the first stage of our optimization process to be the true circumference of the closed environment. Hence, we can define a cost function for learning the pose graph optimization parameters $\gamma = [\gamma_1, \gamma_2]$ as

$$\min_{\gamma} c(\gamma) = \min_{\gamma} |U - \hat{U}| \quad (21)$$

where \hat{U} represents the estimated circumference after pose graph optimization. Thus, we punish deviations between the estimated circumference based on the original pose graph and the optimized one. In order to estimate the circumference after pose graph optimization, a fit onto GMMs is performed as proposed in Section III-A.

IV. RESULTS

We evaluated the accuracy of the pose graph optimization (performance) and the generality of our approach in different environments (robustness). As a measure for the performance, we used an error metric based on the relative displacement between poses

$$E_{\text{rel}}(\xi) = \frac{1}{N} \sum_{i,j} \text{trans}(\xi_{i,j} \ominus \xi_{i,j}^*)^2 + \text{rot}(\xi_{i,j} \ominus \xi_{i,j}^*)^2 \quad (22)$$

as introduced in [5]. Here, $\xi_{i,j}$ are the relative transformations after pose graph optimization, $\xi_{i,j}^*$ ideally the true relative transformations and *trans* and *rot* separate the translational and rotational components. Additionally, we used a second error metric for comparing results obtained on real lawns where the true poses of the robot are unknown but a groundtruth of the environment is available. Therefore, we constructed a polygon defined by the points X out of the optimized pose graph data and compare this polygon with a polygon representing the groundtruth, X_{true} . We then transform

$$X \leftarrow \mathbf{R} \cdot X + \mathbf{t}, \quad (23)$$

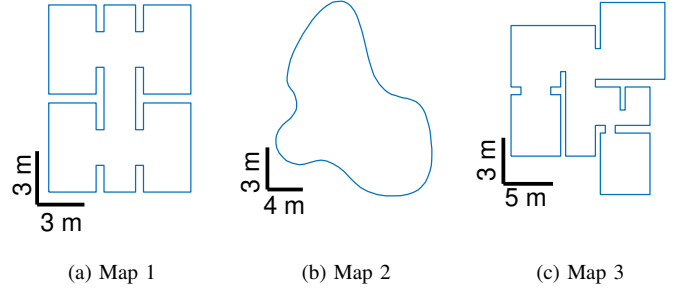


Fig. 5: Simulation environments used for evaluating the proposed learning procedure for mapping in closed environments. From left to right: A symmetric environment ($U = 77\text{ m}$), a curved environment ($U = 52\text{ m}$) and an apartment environment ($U = 100\text{ m}$).

such that the deviation between the enclosed areas A , A_{true} of the polygons

$$\Delta A = 1 - \frac{A_{\text{true}} \cap A_{\text{estimate}}}{A_{\text{true}} \cup A_{\text{estimate}}} \quad (24)$$

is minimized. Here, \mathbf{R} is a rotation matrix and \mathbf{t} a translational vector. The minimized difference then serves as secondary error metric.

We compared to the original approach from [28] using handcrafted and learned parameters. The handcrafted parameters have been selected according to the following rules:

The neighborhood L_{NH} should be chosen such that $2L_{\text{NH}}$ is slightly larger than half of the true circumference. Thus, we like to use slightly more than 50% of U for shape comparison. The comparison error threshold c_{min} should be chosen according to the complexity of the given map. A more complex map requires a larger comparison error threshold to account for more complicated comparisons. The meta-parameter φ_{cycle} has to be chosen with regard to the recurrent symmetric structures of the given map. Here, only Map 1 has such structures with which we can cope by setting $\varphi_{\text{cycle}} = \pi/2$. The pose graph optimization parameters γ_1, γ_2 are kept constant with $\gamma_1 = 1$ and $\gamma_2 = 1$.

A. Simulation

We show the robustness of the approach applying our mapping procedure in different simulated closed environments with hard features, such as recurrent structures, large dimensions or curvatures. For the simulation environment, we used the odometry motion model presented in [32]. We calibrated the odometry model by tracking lawn mower movements using a visual tracking system (OptiTrack) and computed the parameters using maximum likelihood estimation [25]. The calibrated parameters for the Viking MI 422P robot are presented in Table II and are used for the simulation. To generate movement data, we used a wall-following algorithm cycling for $T = 2000\text{ s}$ along the boundary of the closed environment. We statistically evaluated our approach

TABLE II: Measured Parameters for the odometry motion model [32].

α_1	α_2	α_3	α_4
0.0849	0.0412	0.0316	0.0173

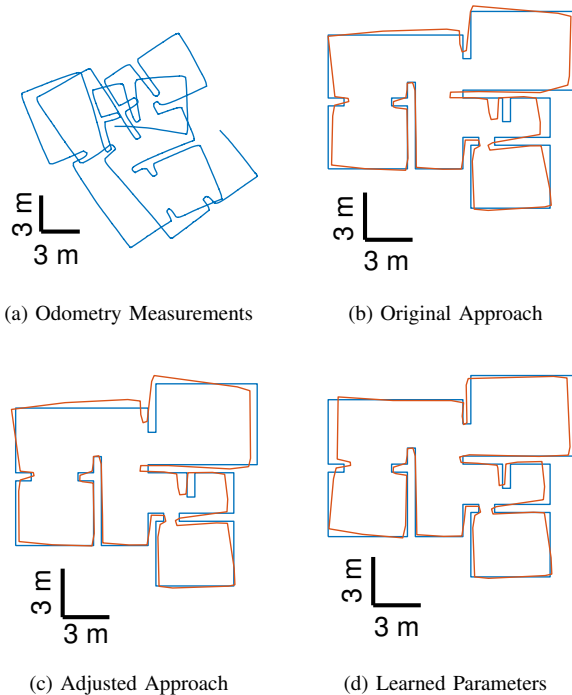


Fig. 6: Exemplary mapping results with the simulation environment "Map 3" and an odometry error of $\alpha_i = 0.2$. In (b)-(d) the blue line shows the true shape of the environment and the red line the map estimate.

simulating 20 runs with a maximum of 30 iterations for the Bayes Optimizer. This optimization is designed for global optimization of black-box functions and does not require any derivatives.

In Table III, the simulation results for the different maps from Figure 5 with different combinations of hand-crafted parameters are presented. Our adjusted approach, using the ICP method, clearly outperforms the original method. In all 20 runs it leads to better map estimates after pose graph optimization. Moreover, learning the parameters enables the algorithm to generalize to different environments without prior knowledge about the odometry error, the shape or the circumference of the environment. This prior knowledge is essential for choosing suitable hand-crafted parameters. Without such knowledge, the parameters have to be manually tuned which might lead to disastrous mapping results. For example, changing the neighborhood parameter L_{NH} for Map 3 to $L_{NH} = 15$ results in a large increase of the mapping errors. In addition, a change in odometry error accuracy can be compensated by learning the mapping meta-parameters, as demonstrated in simulations with odometry model parameters $\alpha_i = 0.1, 0.2$ for $i = 1, \dots, 4$.

B. Real Data

For generating real data, we drove the lawn mower along the boundary line of two different lawn areas. The velocity of the lawn mower driving along the boundary has been set to 0.3 m s^{-1} . The odometry data has been sampled with a frequency of approximately 20 Hz.

In Figure 7, the university courtyard, the measured odometry data and the generated map estimate are shown. The ground truth is available as CAD data, such that we can compare our map estimations using Equation (24). Based on the circumference $U = 106.8 \text{ m}$ and the complexity of the environment, the hand crafted parameters have been set to $L_{NH} = 30$, $c_{\min} = 0.3$. The resulting mapping error for the original approach is $\Delta A = 11.49\%$, for the adjusted approach $\Delta A = 9.77\%$ and the mapping error with learned parameters $c_{\min} = 0.1967$, $L_{NH} = 32.32$, $\varphi_{\text{cycle}} = 1.57$, $\gamma_1 = 0.0104$, $\gamma_2 = 0.0122$ is $\Delta A = 9.24\%$. Again, the adjusted approach outperforms the original approach and learning the parameters with the proposed cost functions leads to sufficiently accurate results.

In addition, we evaluated the mapping approach in a second real environment, a representative of a typical private lawn. In Figure 8 from left to right, we show a part of the private lawn, the measured odometry data and the map estimate. Since we do not have ground truth data for this lawn, we compared the map results qualitatively with the image of the real garden. As demonstrated, the approach is capable of mapping large closed environments with narrow corridors based on severely distorted odometry data.

V. CONCLUSION

Towards efficient localization and planning for low-cost robots, a first step is the generation of an accurate map estimate of the enclosed environment. Thereby, the robot has to learn required meta-parameters automatically to be able to adapt to different environments. Here, we have made improvements to the mapping algorithms for closed environment introduced in [28], which significantly enhance the performance by allowing the algorithm to cope with recurrent symmetric structures as well as reducing the relative displacement error. Moreover, we proposed a cost function for meta-parameter learning for mapping algorithms in closed environments. This cost function does neither require any a-priori information about the environment nor domain expert knowledge and thus enables the robot to act truly autonomously. We demonstrated the feasibility, robustness and performance of our approach in both simulated and real closed environments. Thereby, we showed that based on the proposed mapping procedure, accurate map estimates of underlying closed environments can be produced. These map estimates are the first step towards intelligent behavior for low-cost robots, such as autonomous lawn mowers.

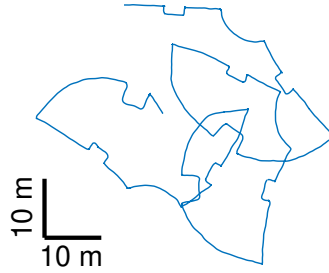
TABLE III: Simulation results for different maps and hand-crafted parameters for the original approach from [28], the adapted approach and with learned parameters. The table shows the mean and the standard deviations for the relative displacement errors.

*the measured odometry parameters from Table II are used

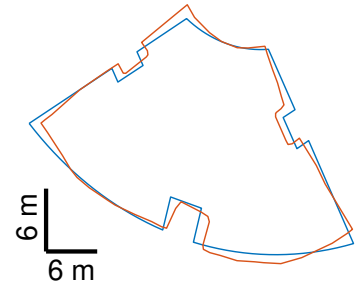
Map	L_{NH}	c_{min}	α_i	Original Approach		Adjusted Approach		Learned Parameters	
				E_{trans}	E_{rot}	E_{trans}	E_{rot}	E_{trans}	E_{rot}
1	20	1.0	*	0.0036 ± 0.0028	0.0076 ± 0.0053	0.0021 ± 0.0034	0.0052 ± 0.0055	0.0006 ± 0.0015	0.0093 ± 0.0166
2	15	0.5	*	0.0406 ± 0.0943	0.0569 ± 0.0399	0.0183 ± 0.0068	0.0564 ± 0.0404	0.0002 ± 0.0005	0.0003 ± 0.0003
3	30	0.3	*	0.1290 ± 0.5532	0.0049 ± 0.0098	0.0020 ± 0.0050	0.0021 ± 0.0021	0.0024 ± 0.0036	0.0262 ± 0.0780
3	30	1.5	*	2.787 ± 12.16	0.0095 ± 0.0093	0.5442 ± 2.004	0.0063 ± 0.0136	0.0017 ± 0.0026	0.0335 ± 0.0607
3	15	0.3	*	35.57 ± 158.0	0.0191 ± 0.0214	31.03 ± 137.6	0.0139 ± 0.0220	–	–
3	30	0.3	0.1	0.0151 ± 0.0418	0.0060 ± 0.0086	0.0085 ± 0.0296	0.0025 ± 0.0020	0.0070 ± 0.0079	0.0665 ± 0.1932
3	30	0.3	0.2	44.54 ± 188.8	0.0304 ± 0.0580	1.65 ± 6.88	0.0158 ± 0.0285	0.0205 ± 0.0387	0.0352 ± 0.0642



(a) The courtyard of our Institute. We used the inner lawn area for testing the proposed mapping method.



(b) The estimated path of the robot generated from its wheel odometry.

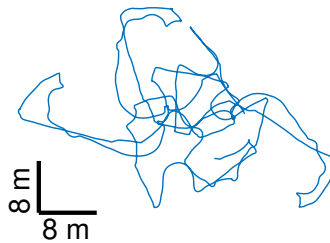


(c) The estimated map (red) and the true shape of the test environment (blue).

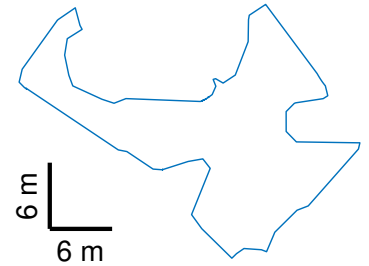
Fig. 7: The real courtyard depicted (a), the collected odometry data (b) and the map estimate with learned parameters (c).



(a) The top view onto a part of a lawn of a typical private household.



(b) The estimated path of the robot generated from its wheel odometry.



(c) The estimated map.

Fig. 8: A typical lawn (a), the collected odometry data (b) and the map estimate with learned parameters (c).

A. Discussion

The underlying assumption of a zero mean odometry error is quite strong and might not hold true under many circumstances, for example if one of the wheels is slightly smaller (e.g. due to air pressure). However, fusing the wheel odometry with IMU measurements, we are able to compensate for such inaccuracies. Moreover, we can detect wheel slippage. Otherwise, a non-zero odometric mean error will be inherited in the final map estimate and thus compensated by navigating with the same robot odometry.

“mown lawn” probabilities similar as in [14] can be used in combination with an adjusted intelligent complete coverage path planning algorithm, e.g. neural network approach [33]. Thereby, the “mowing probabilities” of the grid map are actualized based on a particle filter estimation.

In future work, we will investigate the possibilities of probabilistic approaches for efficiently mowing the lawn with high-confidence. Therefore, coverage grid maps with “already

REFERENCES

- [1] Yassin Abdelrasoul, Abu Bakar Sayuti HM Saman, and Patrick Sebastian. A quantitative study of tuning ros gmapping parameters and their effect on performing indoor 2d slam. In *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pages 1–6. IEEE, 2016.
- [2] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117, 2006.
- [3] Per Bergström and Ove Edlund. Robust registration of point sets using iteratively reweighted least squares. *Computational optimization and applications*, 58(3):543–561, 2014.
- [4] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.
- [5] Wolfram Burgard, Cyrill Stachniss, Giorgio Grisetti, Bastian Steder, Rainer Kümmerle, Christian Dornhege, Michael Ruhnke, Alexander Kleiner, and Juan D Tardós. A comparison of slam algorithms based on a graph of relations. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2089–2095. IEEE, 2009.
- [6] Young-Ho Choi, Tae-Kyeong Lee, and Se-Young Oh. A line feature based slam with low grade range sensors using geometric constraints and active exploration for mobile robot. *Autonomous Robots*, 24(1):13–27, 2008.
- [7] V Ciupe and I Maniu. New trends in service robotics. In *New trends in medical and service robots*, pages 57–74. Springer, 2014.
- [8] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.
- [9] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.
- [10] Nils Einecke, Jörg Deigmöller, Keiji Muro, and Mathias Franzius. Boundary wire mapping on autonomous lawn mowers. In *Field and Service Robotics*, pages 351–365. Springer, 2018.
- [11] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.
- [12] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 2432–2437. IEEE, 2005.
- [13] Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard, et al. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1):34, 2007.
- [14] Jürgen Hess, Maximilian Beinhofer, and Wolfram Burgard. A probabilistic approach to high-confidence cleaning guarantees for low-cost cleaning robots. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 5600–5605. IEEE, 2014.
- [15] Martin Hägele. Robots conquer the world [turning point]. *IEEE Robotics & Automation Magazine*, 23(1):120–118, 2016.
- [16] Myung-Jin Jung, Hyun Myung, Sun-Gi Hong, Dong-Ryeol Park, Hyoung-Ki Lee, and SeokWon Bang. Structured light 2d range finder for simultaneous localization and map-building (slam) in home environments. In *Micro-Nanomechatronics and Human Science, 2004 and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society, 2004.*, pages 371–376. IEEE, 2004.
- [17] Mathieu Labbe and François Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666. IEEE, 2014.
- [18] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019.
- [19] Hyoung-Ki Lee, Kiwan Choi, Jiyoung Park, and Hyun Myung. Self-calibration of gyro using monocular slam for an indoor mobile robot. *International Journal of Control, Automation and Systems*, 10(3):558–566, 2012.
- [20] Seongsu Lee, Sukhan Lee, and Seungmin Baek. Vision-based kidnap recovery with slam for home cleaning robots. *Journal of Intelligent & Robotic Systems*, 67(1):7–24, 2012.
- [21] Feng Lu and Evangelos Miliotis. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.
- [22] Geoffrey J McLachlan and Kaye E Basford. *Mixture models: Inference and applications to clustering*, volume 84. M. Dekker New York, 1988.
- [23] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129):2, 1978.
- [24] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598, 2002.
- [25] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.
- [26] Ozan Ozisik and Sirma Yavuz. Simultaneous localization and mapping with limited sensing using extended kalman filter and hough transform. *Tehnicki vjesnik/Technical Gazette*, 23(6), 2016.
- [27] Douglas Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, pages 827–832, 2015.
- [28] Nils Rottmann, Ralf Bruder, Achim Schweikard, and Elmar Rueckert. Loop closure detection in closed environments. In *European Conference on Mobile Robots*, pages 1–8, 2019. <https://arxiv.org/abs/1908.04558>.
- [29] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.
- [30] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.
- [31] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 2. MIT press Cambridge, 1998.
- [32] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.
- [33] Simon X Yang and Chaomin Luo. A neural network approach to complete coverage path planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):718–724, 2004.
- [34] Ying Zhang, Juan Liu, Gabriel Hoffmann, Mark Quilling, Kenneth Payne, Prasanta Bose, and Andrew Zimdars. Real-time indoor mapping for mobile robots with limited sensing. In *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, pages 636–641. IEEE, 2010.