Simultaneous Parameter Calibration, Localization, and Mapping for Robust Service Robotics

Rainer Kümmerle Giorgio Grisetti Cyrill Stachniss Wolfram Burgard

Abstract— **Modern service robots are designed to be deployed by end-users and not to be monitored by experts during operation. Most service robotics applications require reliable navigation capabilities of the robot. The calibration parameters of a mobile robot play a substantial role in navigation tasks. Often these parameters are subject to variations that depend either on environmental changes or on the wear of the devices. In this paper, we propose an approach to simultaneously estimate a map of the environment, the position of the on-board sensors of the robot, and its kinematic parameters. Our method requires no prior knowledge about the environment and relies only on a rough initial guess of the platform parameters. The proposed approach performs on-line estimation of the parameters and it is able to adapt to non-stationary changes of the configuration. Our approach has been implemented and is used on the EUROPA robot, a service robot operating in urban environments. In addition to that, we tested our approach in simulated environments and on a wide range of real world data using different types of robotic platforms.**

I. Introduction

A key motivation for building robots is to develop systems that can autonomously carry out different tasks and support humans in their environments. There are several research projects focusing on service robots that operate in domestic environments [1], hospitals [2], or urban environments [3], [4]. In all the these projects, robots are deployed in realistic scenarios, which can change over time and show different characteristics.

Whenever robots are deployed to operate without experts installing or supervising them, appropriate calibration and autonomous recalibration of the platform is important for robust operation. First, the robot must calibrate its sensor setup typically starting with the factory default settings. Second, the robot must adapt its calibration parameters over time for several reasons. For example, robots carrying loads are likely to experience a change in their navigation behavior. Health care robots guiding persons via a direct physical interface, e.g. [2], need to adapt to the changes in the environment at all time. The same holds for robots operating in urban environments, where changing underground will result in different navigation behaviors or odometry errors.

In most service robotics tasks, navigation and environment mapping for localization and planning plays an important role. Many navigation systems rely on the knowledge of the specific robot parameters. These parameters typically include the position of the sensor on the platform or the parameters of the kinematic model that translates encoder ticks into a relative movement of the mobile base. The influence of the parameters on the accuracy of state estimation processes can be substantial. For instance, an accurate calibration of the odometry can seriously improve the expected accuracy of the motion prediction by reducing the search space of the algorithms that provide the motion estimates.

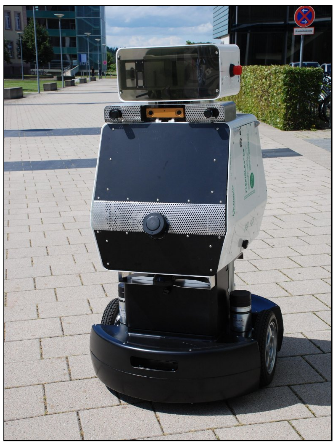

Fig. 1. The robot used within the EUROPA project offers service tasks in urban environments such as guided tours and delivery.

In this paper, we present an approach to estimate the calibration parameters of a robot equipped with a laser scanner and wheel encoders while it performs SLAM. The system is used within the EUROPA project [3] for robots offering service tasks in urban environments (see Figure 1). We model the problem as a hyper-graph, where each node represents either a robot position, the laser position on the robot, or the kinematic parameters of the odometry. Our approach allows to determine these state variables on the fly (e.g., sensor positions and odometry calibration). To deal with temporal changes, which occur for example if the EUROPA robot carries goods or a bag for a human, or moves on different surfaces, we re-estimate the parameters on the most recent data. This approach allows a mobile robot to estimate a different set of odometry parameters for different regions of the environment and to better model the motion of the robot in these areas. Our approach might additionally be beneficial in a variety of contexts including, for instance, terrain classification. We present evaluations of our approach in simulated and real world experiments using several robot platforms moving on different types of ground.

This work has partly been supported by the European Commission under FP7-231888-EUROPA and FP7-248873-RADHAR.

All authors are with the University of Freiburg. G. Grisetti is also with Sapienza, University of Rome.

II. Simultaneous Calibration, Localization, and Mapping

Our system relies on the graph-based formulation of the SLAM problem to estimate the maximum-likelihood configuration. In contrast to the traditional SLAM methods we explicitly model that the measurements obtained by the robot are given in different coordinate frames. For example, the odometry of the robot is given by the velocity measurements of its wheels. Applying the forward kinematics of the platform allows to transform the velocities measured during a time interval into a relative displacement of the platform expressed in the odometry frame. Additionally, the robot is usually equipped with a sensor that is able to observe the environment, e.g., a laser range finder. This sensor is mounted on the robot and obtains measurements in its own coordinate frame. Thus, a scan-matching algorithm which aligns two range scans in a common coordinate frame has to project the computed motion through the kinematic chain of the robot to estimate the motion of the robot's base. As it is not always easy to measure the transformation between the base of the robot and the sensor or to determine the parameters for the forward kinematics, we suggest to integrate those into the maximum likelihood estimation process.

A. Description of the Hyper-Graph

Whenever the robot obtains a measurement we add a node to the graph. This node represents the position of the robot at which the measurement was obtained. Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ be a vector of parameters, where $\mathbf{x}_i = (x_i, y_i, \theta_i)^\top$ describes the position of node i. Furthermore, let $\mathbf{l}$ be the 2D pose of the sensor relative to the coordinate frame of the robot and let $\mathbf{z}_{ij}$ and $\mathbf{\Omega}^{\mathbf{z}}_{ij}$ be respectively the mean and information matrix of an observation of node j seen from node i. Finally, let $\mathbf{k}$ be the parameters of the forward kinematics function and $\mathbf{u}_i$ and $\mathbf{\Omega}^{\mathbf{u}}_i$ be respectively the motion command and the information matrix which translates the robot from node i to $i+1$.

The error function $\mathbf{e}^{\mathbf{l}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{l}, \mathbf{z}_{ij})$ measures how well the parameter blocks $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{l}$ satisfy the constraint $\mathbf{z}_{ij}$. If the three parameters perfectly satisfy the error function, then its value is $\mathbf{0}$. Here, we assume that the laser is mounted without inclination which is the ideal condition. For simplicity of notation, we will encode the involved quantities in the indices of the error function:

$$\mathbf{e}^{\mathbf{l}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{l}, \mathbf{z}_{ij}) \overset{\text{def.}}{=} \mathbf{e}^{\mathbf{l}}(\mathbf{x}_i, \mathbf{x}_j) \overset{\text{def.}}{=} \mathbf{e}^{\mathbf{l}}_{ij}(\mathbf{x}). \tag{1}$$

The error function $\mathbf{e}^{\mathbf{l}}_{ij}(\mathbf{x})$ has the following form:

$$\mathbf{e}^{\mathbf{l}}_{ij}(\mathbf{x}) = ((\mathbf{x}_j \oplus \mathbf{l}) \ominus (\mathbf{x}_i \oplus \mathbf{l})) \ominus \mathbf{z}_{ij}, \tag{2}$$

where $\oplus$ is the usual motion composition operator [5] and $\ominus$ its inverse.

Additionally, the error function $\mathbf{e}^{\mathbf{u}}_i(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{k}, \mathbf{u}_i)$ measures how well the parameter blocks $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{k}$ satisfy the constraint $\mathbf{u}_i$. Again, a value of $\mathbf{0}$ means that the constraint is perfectly satisfied by the parameters. The error function $\mathbf{e}^{\mathbf{u}}_i(\mathbf{x})$ is defined as

$$\mathbf{e}^{\mathbf{u}}_i(\mathbf{x}) = (\mathbf{x}_{i+1} \ominus \mathbf{x}_i) \ominus K(\mathbf{u}_i, \mathbf{k}), \tag{3}$$

where $K(\cdot)$ is the forward kinematics function converting from wheel velocities to a relative displacement of the vehicle. In Eq. (3) we applied the same simplifying notation as in Eq. (1).

For a robot with a differential drive, which is one of the most common types of robots, the odometry $\mathbf{u} = (v_l, v_r)^\top$ consists of the velocities of the left and the right wheel. The wheel velocities are computed by counting the encoder ticks of the motors during the time step which are multiplied by the respective radii r_l and r_r of the wheels. Furthermore, the distance b between the two wheels has to be known to compute the circular arc on which the robot moves. The relative motion during the time interval Δt is given by

$$K(\mathbf{u}, \mathbf{k}) = \begin{pmatrix} R(\Delta t\omega) & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} -ICC \\ 0 \end{pmatrix} + \begin{pmatrix} ICC \\ \Delta t\omega \end{pmatrix}, \tag{4}$$

where $R(\cdot)$ is the 2D rotation matrix of its argument, $ICC = (0, \frac{b}{2}\frac{r_l v_l + r_r v_r}{r_l v_l - r_r v_r})^\top$, and $\omega = \frac{r_l v_l - r_r v_r}{b}$. Thus, the calibration parameter $\mathbf{k} = (r_r, r_l, b)^\top$ for the odometry is a three-dimensional vector.

The goal of our maximum likelihood approach is to find the configuration of $[\mathbf{x}^*, \mathbf{l}^*, \mathbf{k}^*]$ which minimizes the negative log-likelihood $\mathbf{F}(\mathbf{x}, \mathbf{l}, \mathbf{k})$ given all the observations

$$\begin{aligned}&\mathbf{F}(\mathbf{x}, \mathbf{l}, \mathbf{k}) \\ &\quad = \sum_{\langle i,j \rangle} \mathbf{e}^{\mathbf{l}}_{ij}(\mathbf{x})^\top \mathbf{\Omega}^{\mathbf{z}}_{ij} \mathbf{e}^{\mathbf{l}}_{ij}(\mathbf{x}) + \sum_i \mathbf{e}^{\mathbf{u}}_i(\mathbf{x})^\top \tilde{\mathbf{\Omega}}^{\mathbf{u}}_i \mathbf{e}^{\mathbf{u}}_i(\mathbf{x}),\end{aligned} \tag{5}$$

where $\tilde{\mathbf{\Omega}}^{\mathbf{u}}_i$ is the projection of $\mathbf{\Omega}^{\mathbf{u}}_i$ through the forward kinematics function $K(\cdot)$ via the unscented transformation [6]. Since the projection depends on the estimate of $\mathbf{k}$, we update the projection if $\mathbf{k}$ changes substantially.

Given this formulation we may easily integrate prior knowledge, for example, the manually — thus non-precisely — measured transformation of the laser. This is possible as long as the prior information can be represented by a Gaussian distribution. Furthermore, actuated sensors can be easily incorporated by extending the the error function to take into account the measured positions of the joints. This is for instance the case of a laser mounted on a pan-tilt unit.

If a good initial guess of the parameters is known, a numerical solution of Eq. (5) can be obtained by using the popular Gauss-Newton or Levenberg-Marquardt algorithms [7, §15.5]. The idea is to approximate the error function by its first order Taylor expansion around the current estimate. The Taylor expansion leads to a linear system of equations whose solution is used to update the current guess resulting in a new estimate for the parameters. Iterating the linearization, solving, and updating steps yields the solution to Eq. (5).

We employ the g^{2}o toolkit [8] which allows us to solve one iteration of a calibration problem having 3,000 nodes in less than 0.01 s using one core of an Intel i7@2.8 GHz.

III. Experiments

The approach described above has been implemented and evaluated on both simulated and real-world data acquired

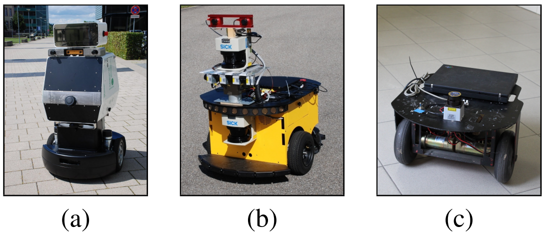

Fig. 2. The robots used to acquire the real-world data sets: (a) EUROPA platform (b) MobileRobots PowerBot (c) Pioneer.

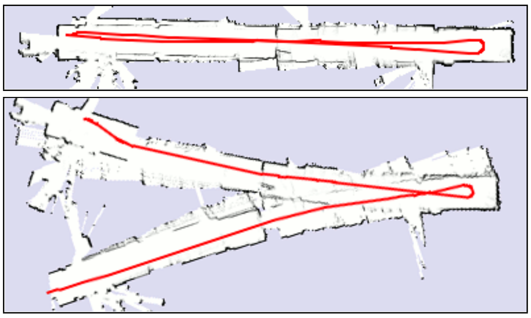

Fig. 3. Robot driving up and down a corridor. Top: Applying the calibration corresponding to the current configuration of the robot leads to a good odometry estimate. Bottom: If the robot is carrying a load, the same calibration parameters results in a severe drift in the odometry.

with a heterogeneous set of robots equipped with laser range finders. Figure 2 visualizes the robots we used to collect the real-world data used in this paper.

The SLAM front-end for processing the data is an own implementation of the framework described by Olson [9] which employs a correlative scan-matcher to estimate the transformation of the laser along with the 3×3 covariance matrix representing the uncertainty of the estimated transformation. The correlative scan-matcher performs an exhaustive search to determine the best fitting alignment for two laser scans within a given search radius.

A. Online odometry calibration

In real world scenarios the odometry is affected by different factors. For example, if a service robot assists a person with carrying bags, the additional weight compresses inflated tires and results in reduced wheel radii. To this end, we used the PowerBot platform (see Figure 2a) which has a maximum payload of 100 kg to carry a load of approximately 40 kg. The wheels of the PowerBot are inflated tires whose radii are affected by both the air-pressure of the tires and the distribution of the load over the platform. In this set of experiments, the load was intentionally placed on the left hand side of the robot. In a first experiment we recorded datasets in which the robot was either carrying the load or it was operating in its normal configuration. We used one data set for estimating the parameters and a different one for evaluating the odometry calibration parameters. Our approach estimated wheel radii of $r_r = 0.1251\,\text{m}$, $r_l = 0.1226\,\text{m}$ for the normal configuration of the robot and $r_r = 0.1231\,\text{m}$, $r_l = 0.1223\,\text{m}$ while carrying the load. The difference seems to be small, however it has a substantial effect. Figure 3 shows the outcome of applying the estimate of the normal configuration to the robot carrying the load. Applying the wrong calibration parameter has a crucial effect on the trajectory as it is estimated by the odometry. Since the weight of the load is mutable and can be placed in an arbitrary position on the robot, the best performance can be obtained by calibrating the odometry parameters while the robot is operating.

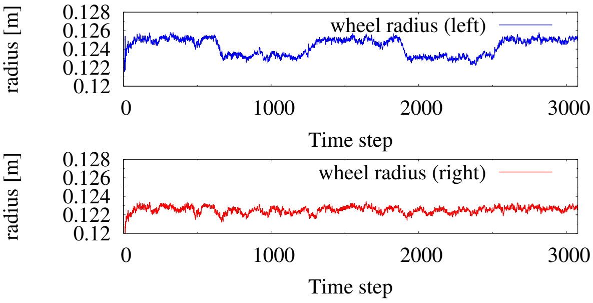

Fig. 4. Results of the online estimation of the wheel radii. The robot had to carry a load twice which was placed on the left hand side of the platform leading to a compression of the left wheel.

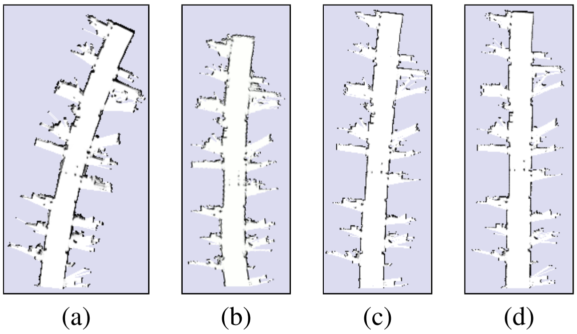

Fig. 5. (a) Map obtained by the raw uncalibrated odometry of a robot traveling along a corridor. The result of applying a scan-matching algorithm with a large search space to account for the uncalibrated odometry leads to the shortened map shown in (b). A restriction of the search space is not able to fully correct the errors as visualized in (c). However, applying the accurate calibration together with a small search space leads to an accurate estimate depicted in (d).

By considering the 50 most recent measurements within a sliding window around the current node we are able to estimate the wheel radii online also when they are subject to change due to external factors. Figure 4 visualizes the estimated wheel radii during an experiment in which the robot had to carry a load placed on the left hand side of the platform. The robot was carrying the load during the intervals $[600, 1250]$ and $[1865, 2530]$. Using our approach we are able to correctly estimate the wheel radii independent of the load carried by the robot along with the maximum likelihood map of the environment.

To further illustrate the importance of an accurate calibration on the navigation performance of a service robot, we recorded a dataset in which the EUROPA robot drives along a straight corridor. As depicted in Figure 5, an accurate calibration of the odometry can seriously improve the expected accuracy of the motion prediction. Here, a scan-matching algorithm based on the odometry prediction is

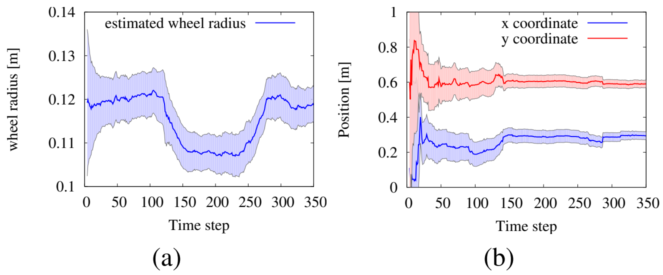

Fig. 6. (a) Estimating the wheel radii online based on the most recent observations. Here, the robot was carrying a load during the time interval $[120, 240]$. (b) The evolution of the x and y coordinate of the laser transformation as it is estimated by our approach. The true value of the x and y coordinate is 0.3 m and 0.6 m respectively.

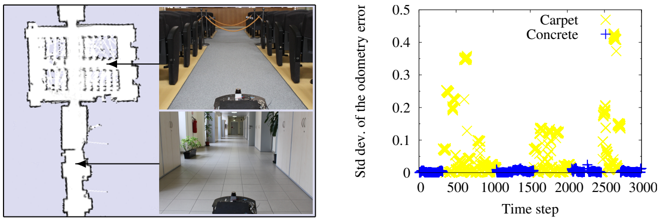

Fig. 7. Left: In indoor environments a robot may encounter different floor types. Right: The standard deviation of the error of the odometry edges for the sliding window at each time step.

mapping the corridor. A scan-matching algorithm aligns two point sets such that the sum of the squared distance between corresponding points is minimized. Since the corridor is not rich in features, the scan matcher yields solutions that are highly ambiguous along the corridor's axis. As a result, scan-matching approaches tend to make corridors "shorter". To limit this effect one can restrict the search space of the scan-matcher to a small region around the position predicted by the odometry. This reduces the computational requirements but requires a highly accurate calibration of the odometry to achieve the correct alignment. As shown in Figure 5c and 5d a precise calibration of the odometry parameters provides an acceptable initial guess which allows the scan-matching algorithm to accurately map the corridor.

B. *Simulation Experiments*

As it is hard to obtain ground truth data for real-world data-sets, we simulated a robot which allows us to directly judge the quality of the calibration results. Within a simulation experiment we modeled a robot carrying a weight which we simulated having the effect of a reduction of the wheel radius from 0.12 m to 0.11 m. The robot carries the load during the time interval $[120, 240]$. Figure 6a depicts the results of the online calibration based on the most recent measurements. As we can see, the estimate is able to represent the compressed wheels and corresponds well to the ground truth given by the simulator. Furthermore, Figure 6b shows that our approach is able to accurately estimate the position of the laser mounted on the robot.

C. *Influence of the ground surface*

Within real world environments a robot may encounter different floor types. For example, the negotiable ground surfaces in outdoor environments are typically concrete streets or paved paths. However, driving on cobble stone pavements differs from driving on the smooth surface of a street. Additionally, in indoor environments the service robot may encounter an even larger variety of floor types, e.g., tiling, PVC flooring, wooden floor, or carpets are possible. To test the influence of the floor type, we recorded data sets in which the robot drives on a soft carpet and on concrete tiling floor, see left image in Figure 7. In this experiment we estimated the odometry parameters online. On both floors the estimated wheel radii were the same. However, by analyzing the standard deviation (see right part of Figure 7) in the error of the odometry measurements $\mathbf{e}_i^{\mathbf{u}}$ for the sliding window around the current node, we observe a higher noise in the odometry due to slippage on the carpet. This information can be stored in the map so that the robot can consider it to adjust the motion model noise during a localization task. Exploiting this information may improve the robustness of a service robot while performing its navigation tasks.

IV. Conclusions

In this paper, we presented an approach to estimate the calibration parameters of a service robot while performing SLAM. Our approach extends the graph-based formulation of the SLAM problem to handle the calibration parameters. The overall approach is accurate and applying it online allows the robot to handle changes in the parameters. For example, a robot assisting a person with carrying bags is affected by the influence of the additional load on its odometry parameters. Additionally, in long-term operations the effects of the wear of the robot may be substantial and our approach allows the robot to adapt itself to correct these effects.

Compared to ad-hoc calibration methods our approach solely relies on the on-board sensors of the robot and does not require external information. Additionally, our approach has the potential to provide useful information about the ground surface which affects the uncertainty of the odometry measurements. This information may in the future be be considered by localization algorithms which adjust the motion model based on the stored information.

References

[1] Technical University of Vienna, "RobotsHome—an open platform for home robotics," http://robots-at-home.acin.tuwien.ac.at.

[2] Georgia Tech Healthcare Robotics, "Robotic nurse assistant," http://www.hsi.gatech.edu/hrl/project_nurse.shtml.

[3] University of Freiburg, "EUROPA—european robotic pedestrian assistant," http://europa.informatik.uni-freiburg.de.

[4] Universitat Politécnica de Catalunya, "URUS—ubiquitous networking robotics in urban settings," http://urus.upc.es.

[5] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial realtionships in robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds. Springer Verlag, 1990, pp. 167–193.

[6] S. Julier, "The scaled unscented transformation," in *Proc. of the IEEE Amer. Control Conf*, 2002.

[7] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes, 2nd Edition*. Cambridge Univ. Press, 1992.

[8] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g^{2}o: A general framework for graph optimization," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[9] E. Olson, "Robust and efficient robotic mapping," Ph.D. dissertation, MIT, Cambridge, MA, USA, June 2008.